UQAC
Université du Québec
à Chicoutimi

PEDESTRIAN SOFT BIOMETRICS RECOGNITION USING DEEP LEARNING ON THERMAL IMAGES IN SMART CITIES

PAR RANI BAGHEZZA

THESIS PRESENTED TO L'UNIVERSITÉ DU QUÉBEC À CHICOUTIMI IN PARTIAL FULFILLMENT OF THE REQUIERMENTS FOR THE DEGREE OF PHILOSOPHIÆ DOCTOR (PH.D.) IN THE SUBJECT OF INFORMATIQUE

QUÉBEC, CANADA

ABSTRACT

With technological advancement and the rise of the Internet of Things, our society is becoming more interconnected than ever before. Our computers and devices are getting smaller, and their computing power and memory has been increasing. These advances coupled with the leaps in artificial intelligence caused by the deep learning revolution in recent years have led to an increasingly rising interest in the field of pervasive intelligence.

Intelligence in the environment has been used in smart homes in order to bring assistance to semi-autonomous people by performing activity recognition based on sensor data. As technology keeps improving, we may start to investigate the extension of assistive technologies beyond the boundaries of smart homes and into our smart cities. In order to bring assistance to semi-autonomous people, the first step is to be able to recognize profiles of vulnerable people. In order to leverage technology and artificial intelligence to make our cities smarter, safer and more accessible, this thesis investigates the use of environmental sensors such as thermal cameras to perform pedestrian soft biometrics recognition (age, gender and mobility) in the city.

In this thesis, the process of building prototypes from scratch in order to collect thermal gait data in the city is explored, and the use and optimization of deep learning algorithms to perform soft biometrics recognition, as well as the feasibility of implementing these algorithms on limited resource boards are explored. The use of unprocessed thermal images allows a higher degree of privacy for the citizens, and it is novel in the field of human profile recognition. This thesis aims to set the foundation of future work, both in the field of thermal images-based soft biometrics recognition and pervasive intelligence in our cities in order to make them smarter, and move towards an interconnected society.

RÉSUMÉ

Les progrès technologiques et le développement de l'Internet des Objets nous mènent vers une société de plus en plus interconnectée. Nos ordinateurs et nos appareils deviennent de plus en plus petits et leur puissance de calcul et leur mémoire ne cesse de s'améliorer. Ces avancées combinées aux récents progrès dans le domaine de l'intelligence artificielle avec la révolution de l'apprentissage profond ont mené à un intérêt grandissant dans le domaine de l'intelligence ambiante.

L'intelligence ambiante a été utilisée dans le domaine des maisons intelligentes sous forme de reconnaissance d'activités, permettant d'assister les personnes semi-autonomes en utilisant des données collectées par des capteurs. Alors que le progrès technologique continue, nous arrivons à un point où l'hypothèse d'étendre ces stratégies d'assistance des maisons aux villes intelligentes devient de plus en plus réaliste. Afin d'étendre cette assistance aux villes, la première étape est d'identifier les personnes vulnérables, qui sont celles qui pourraient bénéficier de cette assistance. Dans le but d'utiliser la technologie pour rendre nos villes plus intelligentes, plus sûres et plus accessibles, cette thèse explore l'utilisation de capteurs environementaux tels que des caméras thermiques pour effectuer de la reconnaissance de profils dans la ville (age, genre et mobilité).

Dans cette thèse, le processus de construction de prototypes pour récolter des données thermales dans la ville est présenté, et l'utilisation ainsi que l'optimisation d'algorithmes d'apprentissage profond pour la reconnaissance de profils est explorée. L'implémentation des algorithmes sur un système embarqué est également abordée. L'utilisation d'images thermiques garantit un plus grand degré d'anonymat pour les citoyens que l'utilisation de caméras RGB, et cette thèse représente les premiers travaux de reconnaissance de profils multiples en utilisant uniquement des images thermiques sans pré-traitement. Cette thèse a pour objectif de poser les bases pour des travaux futurs dans le domaine de la reconnaissance de profils en utilisant des images thermiques, ainsi que dans le domaine de l'intelligence ambiante dans nos villes, afin de les rendre plus intelligentes et de se diriger vers une société interconnectée.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

ANT	Ambient Systems, Networks and Technologies
BGRU	Bidirectional Gated Recurrent Units
BLE	Bluetooth Low Energy
BPTT	Backpropagation Through Time
CASIA	Chinese Academy of Sciences (Dataset)
CCR	Correct Classification Rate
CCTV	Closed-Circuit Television
CGI	Chrono-Gait Image
CNN	Convolutional Neural Network
CPU	Central Processing Unit
DAE	Deep Auto-Encoders
DAGSVM	Direct Acyclic-Graph Support Vector Machine
DBN	Deep Belief Networks
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DT	Decision Tree
DTW	Dynamic Time Warping
EMR	Electromagnetic Radiation
FDEI	Frame-Difference Energy Images
FIR	Far Infrared Radiation
FN	False Negative
FP	False Positive
GAN	Generative Adversarial Networks
GCN	Graph Convolutional Networks

GCEM	Gait Convolutional Energy Map
GEI	Gait Energy Image
GEM	Gait Energy Motion
GEnI	Gait Entropy Image
GPR	Gaussian Process Regression
GPS	Global Positioning System
GPU	Graphics Processing Unit
GRU	Gated Recurrent Unit
GSV	Gait Silhouette Volume
HMM	Hidden Markov Model
HOG	Histograms of Oriented Gradients
HTI	Head Torso Image
INTER	Ingénierie de Technologies Interactives en Réadaptation
IoIT	Internet of Intelligent Things
IoMT	Internet of Medical Things
IoT	Internet of Things
IR	Infrared
IRCNN	Image Restoration using Convolutional Neural Networks
k-NN	k-Nearest Neighbors
LBP	Local Binary Patterns
LED	Light-Emitting Diode
LDA	Linear Discriminant Analysis
LiDAR	Light Detection and Ranging
LR	Logistic Regression
LSTM	Long Short Term Memory

LWIR	Long-Wavelength Infrared Radiation
MAE	Mean Absolute Error
ML-KNN	Multi-Label k-Nearest Neighbors
MLBP	Multi-Level Local Binary Pattern
MLP	Multi-Layer Perceptron
MSE	Mean Squared Error
MWIR	Mid-Wavelength Infrared Radiation
NB	Naive Bayes
NIR	Near Infrared
NN	Neural Network
NSERC	Natural Sciences and Engineering Research Council of Canada
ODR-GLCNN	Ordinal Distribution Regression with a Global and Local Convolutional Neural Network
OU-ISIR	Osaka University Institute of Scientific and Industrial Research
OU-LP	Osaka University Large Population
OU-MVLP	Osaka University Multi-View Large Population
PCA	Principal Component Analysis
PEI	Period Energy Image
RBM	Restricted Botzmann Machines
ReLU	Rectified Linear Unit
RF	Random Forest
RGB	Right, Green and Blue
RGB-D	Right, Green, Blue and Depth
RMS	Root Mean Squared
RNN	Recurrent Neural Network
SGD	Stochastic Gradient Descent

SLP	Silhouette Longitudinal Projection
SOTON	Southampton (Gait Dataset)
SSD	Single Shot Detector
STP	Silhouette Transverse Projection
SVM	Support Vector Machine
SVR	Support Vector Regression
SWIR	Short-Wavelength Infrared Radiation
SYSU	Sun Yat-sen University (Gait Dataset)
TN	True Negative
TP	True Positive
TUM-GAID	Technical University of Munich Gait from Audio, Image and Depth
USF	University of South Florida (Gait Dataset)
UQAC	Universite du Quebec à Chicoutimi
VDSR	Very Deep Super Resolution
VGG	Visual Geometry Group
WSN	Wireless Sensor Networks

DEDICATION

To ∞

ACKNOWLEDGEMENTS

This thesis has been carried out as a part of the activities of the Laboratoire d'Intelligence Ambiante pour la Reconnaissance d'Activités (LIARA) at the Université du Québec à Chicoutimi (UQAC) and has been partially funded by the Regroupement INTER (Ingénierie de Technologies Interactives en Réadapatation) and the Natural Sciences and Engineering Research Council of Canada (NSERC), so my first thank goes to these entities for making this thesis possible. I would also like to thank UQAC's Ethics Committee for making the experiments of this thesis possible under Project 2019-227.

I would like to thank my three directors as well. To Kévin Bouchard for his presence, his availability, for sharing his expertise of the world of research with me and for understanding the way I work, which allowed him to give me sufficient freedom while still keeping me on track. To Charles Gouin-Vallerand for his technical expertise, his complementary perspective and his valuable insights at different stages of the project, as well as the opportunities he has provided me with. To Abdenour Bouzouane for his early lectures, for initially putting me on the right track, and for his overall big picture approach of the project, allowing me to stay grounded throughout the thesis despite his many obligations.

Part I

Introduction

CHAPTER I

INTRODUCTION

1.1 A SMART AND CONNECTED SOCIETY

"When wireless is perfectly applied, the whole earth will be converted into a huge brain, which in fact it is, [...] and the instrument through which we shall be able to do this will be amazingly simple compared to our present telephone"— Nikola Tesla

Technological advancement and miniaturization in the field of computer science have allowed the emergence and widespread of mobile and pervasive computing in the past decades. Moore's law, predicting the exponential growth of computer speed thanks to the miniaturization of transistors, has proved to be true to a certain extent, leading to a tremendous increase in performance in commercial computers, supercomputers, as well as mobile, wearable and embedded devices [1]. The number of devices connected to the Internet [2] as well as the overall traffic [3] have been following a similar growth. These two factors combined have led to the rapid development of the Internet of Things (IoT), which is generally defined as physical objects equipped with sensors and software having the capability of communicating with each other and the Internet. The Internet of Things paradigm creates an intelligent, invisible network fabric that can be sensed, controlled and programmed [4] to handle specific tasks. The root of the idea leading to the IoT has sometimes been attributed to Nikola Tesla, based on the opening quote of this chapter [5].

The IoT paradigm has been used in many different fields such as transportation [6], healthcare [7] and industrial communication [8]. By filling the environment with smart objects able to collect data and send it over to distant servers for analysis, we are getting closer to Weiser's vision of technology being weaved into the fabric of reality [9]. In the same way

smartphones and social media connect people, the IoT connects objects and the data they collect, which allows us to explore opportunities in the field of pervasive computing and ambient intelligence. The IoT has also given rise to smart wearables for health, sports and daily activity, tracking and localization as well as safety applications [10]. Everyday objects and accessories, such as watches, bracelets, belts or even clothing are equipped with various sensors, such as gyrometers, accelerometers or heart rate sensors and use Wi-Fi or Bluetooth Low Energy (BLE) to communicate with a smartphone or a distant server in order to store the collected data and provide the user with various analytics.

Regardless of the field of application, tools are needed in order to analyze the data collected by various types of sensors. We also need intelligent programs capable of using this data to recognize patterns, predict future behaviours and trends, and solve various societal problems. One of the most investigated avenue to extract value from data has been the use of artificial intelligence related methods. Artificial intelligence in broad terms has been defined as *"The science and engineering of making intelligent machines, especially intelligent computer programs"* [11] and it has been used in many different fields such as finance [12], healthcare [13], cyber security [14], education [15], the military [16] or advertising [17]. Techniques such as data mining and machine learning can be used to extract knowledge from data [18], or learn from experience without being explicitly programmed to perform a task, respectively [19].

The end vision of combining artificial intelligence and the Internet of Things would be the widespread of self-aware, self-managing and self-configuring pervasive intelligent systems [20]. The term of Internet of Intelligent Things (IoIT) has been coined by Serra et al. [21], where the emphasis is put on connectivity between objects and energy management, which are some of the main obstacles for the development of efficient ambient intelligence. We have highlighted 6 axes of optimization in order to move from offline to real-time distributed activity

recognition in Wireless Sensor Networks (WSN) in a recent survey: processing, memory, communication, energy, time and accuracy [22]. Being able to embed the machine learning algorithm directly on the IoT nodes would allow for quick decision making closer to the data source and a reduction in the communication burden between the nodes and a central server, leading to more energy efficient systems. Such a system would also remove the need for a very high centralized processing power in the cloud that could be the source of a bottleneck in the system. The distribution of computation and the reduction of the distance between data collection point and processing unit have started with the use of Fog Computing [23] and Edge Computing [24], where devices on the edge of the network (between sensor nodes and distant servers) carry out some of the computation, as opposed to relying solely on distant servers.

1.2 INTERNET OF THINGS FOR HEALTHCARE

The field of healthcare as a whole has benefited a lot from the emergence of the IoT. Smart watches are used by recreational or competitive athletes to monitor their running speed and heart rate. Smart bracelets can be used by the elderly to allow their healthcare professional to have access to, and remotely monitor their health status. Some example of wearable devices can be found in Figure 1.1.The miniaturization and the reduction of the cost of these devices has allowed to bring healthcare directly into the home in the form of ambient assisted living systems inside smart homes.

1.2.1 INSIDE SMART HOMES

Alam et al. [26] have defined a smart home as being: *an application of ubiquitous computing in which the home environment is monitored by ambient intelligence to provide context-aware services and facilitate remote home control*. Smart homes provide a very good

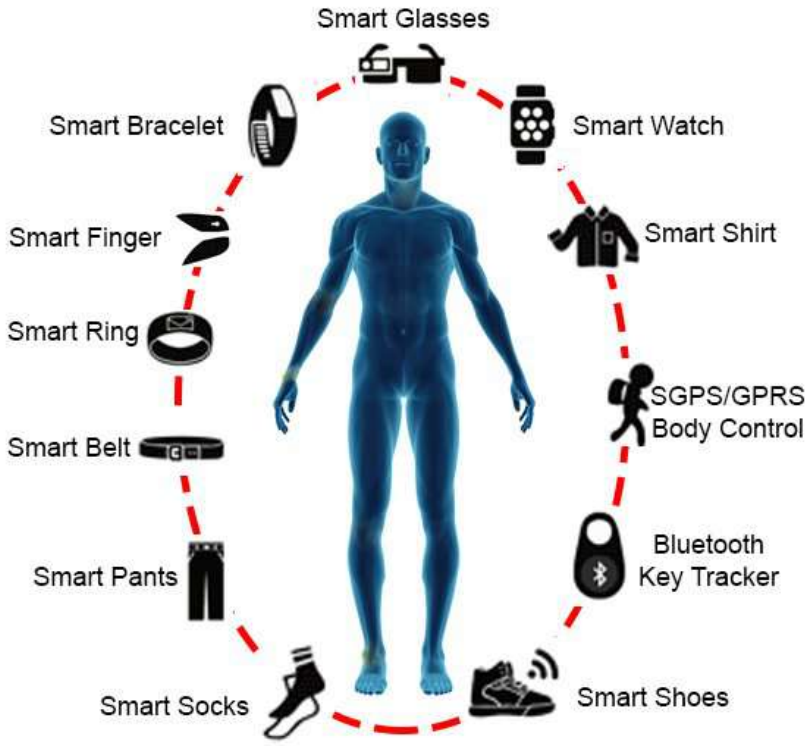

Figure 1.1 : Example of wearable sensor [25]. © 2018 IEEE.

opportunity for healthcare. Indeed, it is projected that the number of people over the age of 65 will reach 1.5 billion in 2050 [27]. Population ageing comes with a set of problems such as a lack of autonomy, Alzheimer's Disease (AD) or dementia. Patients suffering from these ailments require daily assistance that can either be provided by professional caregivers or by family members. The total costs generated by these conditions have been valued at $234 billion dollars in 2019 [28]. The use of technology and ambient assisted living inside smart homes can help reduce these costs by relying on various sensors to monitor the user's Activities of Daily Living (ADL) [29], and performing anomaly detection to anticipate dangerous situations and recognize when the user significantly strays from their habits. Elders seem to welcome this kind of technology into their home, as long as it is user-friendly [30].

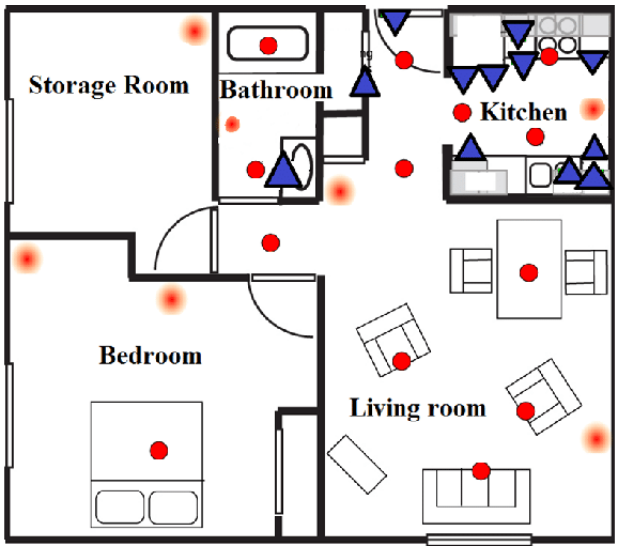

Figure 1.2 : Illustration of a typical smart home where the red circles represent motion sensors and the triangles represent door and cabinet sensors [31]. © 2011 Springer Nature.

Various sensors are used inside smart homes such as environmental and wearable sensors (Figure 1.2). The sequences of activation and the data they collect can be matched with the observed activities of the resident in order to train a machine learning model to learn the pattern of these activities. Initiatives such as Cook's *Smart Home in a Box* [32] have helped making it easier to set up sensors in a smart home, in order to facilitate the collection of data. The limited perimeter of smart homes make it an ideal environment to experiment with ambient assisted living. As long as the set of activities remains limited, and as long as sensors are positioned in strategic locations, activity recognition can be performed with good results. The task becomes more complex when moving to multi-resident homes, as resident identification becomes an issue [33]. One of the additional challenges of activity recognition inside smart homes is the discovery of new activities, that has been tackled by Fortino et al. [34] using the apriori algorithm to find emerging patterns in the data.

However, these ambient assisted living approaches are limited to smart homes and do not allow the user to be monitored when they are outside of their home and in the city, thus

limiting their autonomy. With the current technological advancement of the IoT, it is possible to extend these systems to smart cities to a certain extent.

1.2.2 EXTENSION TO SMART CITIES

Extensive research has been carried out on the use of the IoT in smart cities for applications outside of healthcare [35]. It has been used for various tasks in the city such as controlling street lights [36], monitoring traffic [37], optimal sensor placement for smart parking [38] or smart waste collection systems [39]. The diversity of devices, communication protocols, applications and services makes it difficult to build a general architecture for the systematic development of IoT systems in the city [40]. An overview of the main elements making up a smart city can be found in Figure 1.3.

In an effort to extend ambient assistance to smart cities, we have to take incremental steps. On the way towards pervasive ambient assistance outside of smart homes, a lot of benefits and added value can be found for vulnerable or semi-autonomous populations in the city, such as children, elders and persons with reduced mobility. Added value can be provided for all of the citizens as well. Pervasive technology can be used for accessibility, inclusivity as well as a better social participation for people with disabilities [41].

Directly transferring activity recognition methods from smart homes to smart cities is not a realistic endeavour. Most activities in smart homes are location-dependent, and the concept of *activity* in the city can be very vague, as it requires a proper context. GPS data has been used by Boukhechba et al. [42] to perform outdoor activity recognition using smartphones. In this context, we are considering macro-activities, centered around points of interest in the city. The activity and the location of the user are directly tied: if the user stops for an extended period of time at a bank, it can be inferred that the user is performing some kind of

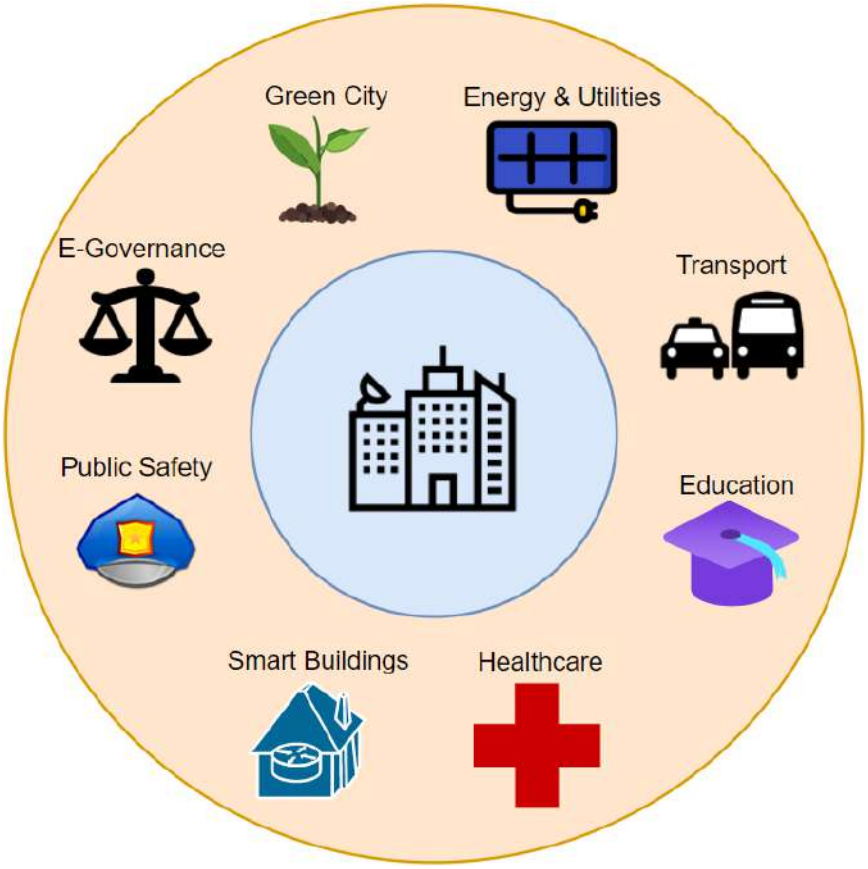

Figure 1.3 : Overview of the main elements of a smart city. © Rani Baghezza, 2022.

financial activity, such as withdrawing money, or meeting with a counselor. The use of GPS data coupled with inertial sensors, such as gyroscopes, can help us narrow down the activity being performed, but it still carries a certain amount of uncertainty, especially considering the wide pool of activities that could be performed anywhere.

As a first avenue to extend activity recognition to smart cities, the use of a smartphone, or wearable sensors directly attached to the user's body (wrist, ankle, chest) could be considered to be quite invasive. Each user would have to individually agree, and remember to equip a smart device or install an application on their smartphone, and possibly provide information about the activities they have performed throughout the day in order to train machine learning

algorithm with the correct ground truth. Such a system would be invasive for the user as well as time consuming, and any user not wearing the device would simply be excluded from the city-wide experiment.

Another more promising approach would be to use environmental sensors directly in the city. This method is less invasive than the use of wearables or a smartphone application, and it allows the collection of data from any user walking in range of the sensors without them having to perform any additional action. The main drawback of this approach is the impossibility to perform user-specific activity recognition. If sensors could be deployed across the entire city, one could argue that the location of the sensors would be a good indicator of the activity being performed, effectively translating the smart home model to smart cities: *if the oven is on, the resident might be cooking* would be translated to *if the bank sensors are firing, users might be performing finance-related activities*. However, it might not be possible to performed detailed activity recognition, and such a system would only allow to perform macro-activity recognition.

As a city-wide deployment is not a realistic first approximation, we have to consider restricting the area of deployment. Regardless of the area, another issue is raised by the use of environmental sensors: user identification. In a smart home with a single resident, there is no issue in knowing who is performing the activity. In multi-resident homes, the issue becomes clearly apparent [33]. In a city portion, even limited to a neighborhood or a street, dozens or hundreds of users could walk by the sensors in a short period of time, and might not ever be seen again by the system. These considerations lead us to the conclusion that simply translating activity recognition from smart homes to smart cities is not the right way to approach this issue. The research problem can therefore be expressed in different terms: how can we take advantage of the IoT and pervasive computing to extend ambient assisted living

from smart homes to smart cities, and how can we add value to vulnerable populations' lives, and facilitate their inclusivity in the city?

The first step towards bringing assistance to vulnerable users is to be able to identify said users in real-time in the city. The term *vulnerable* could encompass many different categories of users depending on the context: children could be considered vulnerable users as their risk for severe injuries following a car accident when crossing the road is higher than the risk for an adult [43]. The elderly could be considered vulnerable because of the higher death rate following fall-related injuries, compared to the rest of the population [44]. Persons with reduced mobility also fall into the category of vulnerable users because of their special needs. Being able to recognize these users could also provide the municipality with useful data about the accessibility of different points of interest in the city, and anticipate the need for adaptations such as ramps, wider parking spaces or more accessible doors and entrances.

1.3 HUMAN PROFILE RECOGNITION

Identifying vulnerable users in the city requires us to perform human profile recognition using environmental sensors. The most direct way of performing profile recognition would be to use cameras, as they allow us to capture detailed visual information about the users in the city. However, a dense deployment of RGB cameras in the city would have two downsides: it could get expensive depending on the number of cameras, as well as the quality and resolution of the captured footage, and it would raise serious privacy concerns from the citizens, as well as ethical issues. CCTV cameras are used all across the world, especially in big cities such as London [45], where an estimated number of 500.000 CCTV cameras are spread across the city. However, these cameras are meant for security purposes, and their positioning is not optimized for profile recognition.

A more interesting avenue to explore is the use of low resolution thermal cameras. The use of thermal imaging devices in smart cities allows to preserve user privacy [46], as distinctive features that could allow the identification of a specific user are not as recognizable. The shape of the silhouette is preserved, and the mode of locomotion of the user can easily be recognized when it comes to identifying people with reduced mobility. By analyzing the way a user walks, as well as the shape of their silhouette and their posture, it is possible to use gait recognition techniques and perform privacy-preserving profile recognition. The low resolution of the cameras add to the challenge from a computer vision standpoint, but the reduced costs make the system viable as a first approach. This is preferable to CCTV approaches that might present a privacy issue for the citizens [47].

1.3.1 GAIT RECOGNITION

Gait recognition is defined as the use of videos of human gait, processed using computer vision methods to recognize or identify persons based on their body shape or their way of walking [48]. With the use of thermal cameras to collect visual data about the users in the city, gait recognition appears to be the most promising avenue for human profile recognition. Most gait recognition approaches fall under one of these three categories: using wearable sensors on the user's body, using video footage and markers located on the user's joints, or using a marker-less, video-based approach, which is the approach of interest in the case of this thesis. This approach is illustrated in Figure 1.4.

Most gait recognition methods aim to identify a single user among several users, based on one or several entries per person. The frames making up the footage are generally captured in an indoors, controlled environment with a fixed background, and the images are captured in high resolution and stored in datasets such as the OU-ISIR dataset [50]. These ideal conditions

Figure 1.4 : Example of gait data from the CASIA dataset [49]. © 2006 IEEE.

allow for very good results, but they are not directly translatable to the real-life case of gait recognition in the city because of a few major differences. The use of low resolution thermal cameras makes it more challenging to achieve good results: a lower resolution comes with a more discrete, less detailed representation of each individual. In the city, the background might be subject to a lot of changes, such as temperature variations causing the thermal sensors to adjust and change the color of the image, or cars, buses and trucks driving by in the background as well as pedestrians walking on the opposite pavement. Performing gait analysis on images with changing background has been known to cause a drop in classification performance [51].

1.3.2 GAIT-BASED PROFILE RECOGNITION

In the context of this thesis, we have no interest in using gait recognition to recognize a specific user: we are interested in categories of users. Recognizing a single user would also be unrealistic considering the high number of different class labels (one per user) that would be generated, and the very low amount of instances per class, namely, a few low resolution frames per user.

Instead, we would like to perform human profile recognition in the city, where each user is associated with a specific profile based on an external observation. The end goal of such a

system is to classify users into different categories corresponding to profiles of interest using thermal imaging sensors in real-time in the city. The additional challenges that comes with the abstraction from individual user to profile recognition are the high intra-class variation and the fuzzy inter-class boundary. A clear example of the latter would be the problem of performing binary classification between adults and elders, considering elders as the more vulnerable population that would benefit from ambient assistance in the city. Indicators such as the pace of the user, as well as their posture could give us information about which class they belong to. However, the line between adult and elder is not always so clear, as some elders can appear to be very healthy, and some adults could move slower and exhibit a hunched over posture, even though they would still qualify as adults if the only criterion was age.

Similarly, classifying individuals between mobile users and users with reduced mobility could be a challenge: a user showing signs of a limp would be considered having reduced mobility, and identically, any person in a wheelchair would fall into that category. However, the challenges these two users face would be different: the first user would probably be able to walk upstairs as long as a rail is present, whereas the user in a wheelchair would require a ramp to access the same building. As far as computer vision methods go, the visual signature of a wheelchair and a person with a limp are drastically different, and the user with a limp would appear to match a mobile user more than the user in a wheelchair, simply because of the standing position. It is therefore necessary to make smart choices and clearly define classes in order to optimize the profile recognition performance of the system.

The problem of classifying broader categories of people has been addressed in the literature in the form of age and gender recognition [52]. However, the vast majority of these approaches use RGB cameras positioned indoors in a controlled environment. A few recent papers have tackled the challenge of gender recognition using a combination of RGB and

thermal cameras, as described in the related work part of this thesis. The objectives of this thesis are therefore the following:

- Build prototypes and design an experimental setup to collect thermal data in the city.
- Perform profile recognition using unprocessed thermal images and conclude on the performance of the approach.
- Optimize deep neural networks and explore methods to improve classification performance.
- Explore the on-board implementation of the classification algorithms and conclude on the limitations and future avenues.

The main research questions this thesis aims to answer are:

- Is it possible to recognize profiles of pedestrians in the city using thermal images?
- Can profile recognition be performed in real-time on an embedded system?
- Which deep learning architecture seems to give the best results, what are their limitations, and how can they be improved?

1.4 CONTRIBUTIONS OF THIS THESIS

This thesis aims to extend the use of ambient intelligence from smart homes to smart cities to promote the accessibility and inclusivity of vulnerable populations.

The first contribution of this thesis is the development of two generations of battery-operated devices capable of collecting thermal and sound data in the city for profile recognition.

The choice of sensors and devices, and the improvement of the devices from the first to the second version of the prototypes are described. The main goal is the optimization of the quality of the data collected by the sensors while keeping the energy consumption manageable and making sure that the system can run in real-time. The first prototype and experiments have been described in a paper accepted in the 11th International Conference on Ambient Systems, Networks and Technologies (ANT) [53].

The second contribution is the collection of an experimental low resolution thermal image-based dataset in the city using the handmade devices and an Android application over the course of several experiments. Indeed, datasets in the field of gait recognition are mainly composed of RGB images, or a mix of RGB images and a few Near Infrared (NIR) images in more recent papers. Moreover, these datasets generally use higher resolution cameras than the ones used in this thesis, making the collected dataset novel, and appropriate for low resolution computer vision tasks. The additional data collected during the experiment, such as traffic data and various movement in the background could also be used for additional work besides gait recognition. The design of the second prototype as well as the associated experiments and the first results have been presented in a paper accepted in the Special Issue on Sustainable Solutions for the Internet of Things of the IEEE Internet of Things Journal [54].

The third and main contribution is the use of deep learning algorithms to perform profile recognition on low resolution thermal data. Various architectures, configurations and datasets were explored to find the best performing parameters for binary profile recognition on the experimental dataset. The path from standard supervised machine learning algorithms on low resolution data collected with the first version of the prototypes, to the use of deep learning models on data collected with the second version of the prototypes is described. Avenues for future work to generalize the task of binary profile recognition to multi-task classification on thermal images in the city are explored. The results of this contribution are currently under

review for publication in the Expert Systems With Applications journal, and a shorter version of the paper is under review for the Good IT 2022 conference.

The last contribution of this thesis is the implementation of deep learning algorithms directly on a Raspberry Pi 3 to explore the feasibility of real-time profile recognition in an embedded system. The necessary adaptations of the algorithms from a desktop to an IoT board in order to ensure the real-time execution of the classification are presented as well.

1.5 RESEARCH METHODOLOGY

The research methodology followed for this thesis was organized in two macro cycles based around the experiments.

The first part of each cycle has been the definition of the research problem. Indeed, the leap from activity recognition in smart homes to profile recognition in the city has required some analysis and re-framing of the problem to solve. It was also necessary to re-frame the research question after the lessons learned from the first experiment, going into the second one.

The second part was the exploration of the literature and the acquisition of the necessary theoretical background. The first cycle was still oriented towards activity recognition and the literature review has lead to the publication of a survey paper in Sensors [22], discussing the evolution from offline activity recognition to distributed, real-time approaches. Some knowledge in electrical engineering had to be explored in order to build the first Arduino Due-based prototype. For the second cycle, the focus was mainly shifted towards gait recognition, image analysis and deep learning.

The third part was the fabrication of the prototypes, which involved a hardware and a software component in each case. That step includes the choice of the boards and sensors, which are guided by the two previous steps and a careful review of the available sensors on the market. The hardware component proved to be more challenging for the first prototype, whereas the software aspect of the second one posed more issues because of the more advanced sensors used. The creation of the associated Android application used during the experiments is included in this step for each cycle,

The fourth part was centered around the experiments, including the design of the experimental setup, the online and offline scouting of different locations, and the evaluation of their suitability for the experiment. The ethics authorizations have been obtained ahead of time for the first experiment and carried over to the second one. The experiments are then carried out, and after each experiment, the raw data and the associated labels were stored, the batteries were charged, and any apparent misbehavior was corrected for the subsequent experiment. A single experiment was carried out with the first prototype, followed by 10 experiments for the second one.

The fifth part was the formatting and sorting of the datasets, as well as the training and testing of the machine learning algorithms. This step was followed by the subsequent analysis and publication of the results in the ANT 2020 Conference for the first prototype [53] and the Special Issue on Sustainable Solutions for the Internet of Things in the Internet of Things Journal for the second prototype [54]. During this part, the algorithms are tuned and experimented with until the best results are achieved, and the next step of the research is planned out depending on the new insights learned throughout this analysis.

1.6 THESIS ORGANIZATION

This thesis is split into four parts. The first part lays down the theoretical foundation in machine learning and deep learning in order for the reader to have the tools to apprehend the rest of the thesis. The second part reviews related work in the literature, starting with the introduction to individual gait recognition and the different methods used over the years in that field, followed by the exploration of gait-based profile recognition, with a focus on age and gender recognition and various applications involving thermal images. The third part presents the experiments, datasets, methodology, results and analysis that constitute the added value of this thesis. In this part, the first results from early experiments are presented before diving into more details in the improved prototypes, experiments, and higher quality data used to perform profile recognition in the city using thermal images. Different deep learning models are explored and compared, and the results are put into the context of this work, taking into account the noisy environment and the challenging dataset. The embedded implementation of the deep learning models for on-node classification is also explored, as well as the limitations of this implementation. Finally, the fourth part concludes the thesis with a discussion of the overall research problem and the results obtained, as well as the future research avenues, questions to be answered, and challenges to overcome in order to achieve real-time profile recognition in an urban environment using thermal images.

Part II

Machine Learning Background

CHAPTER II
MACHINE LEARNING

This second chapter covers the basics of machine learning with an emphasis on deep learning. Since the following chapters present methods using deep learning for gait and profile recognition, it is important for the reader to have a good grasp on these approaches. This chapter is not as crucial for readers who are already advanced or experts in the field of machine learning and deep learning, however, it can provide a few reminders or at least give a consistent set of terms that will be used throughout the rest of this thesis.

In its most widely accepted definition, machine learning refers to *algorithms that learn from experience without being explicitly programmed to perform a task* [19]. Machine learning is a subset of artificial intelligence and machine learning tools can also be used for data mining, which is the process of extracting useful knowledge from data. Since the end goal of this thesis is to build a system that can recognize different human profiles, we explore machine learning and deep learning.

In this chapter, we begin by giving a reminder of some general definitions in the field for supervised, unsupervised and semi-supervised machine learning. We then explore some of the different metrics that can be used to evaluate a model's performance, as well as different training and testing procedures. We then dive into deep learning algorithms, beginning with standard Neural Networks (NN) before moving on to more relevant algorithms for this thesis, such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN)

2.1 GENERAL DEFINITIONS

Generally speaking, machine learning algorithms can be split into three main categories: supervised, unsupervised and semi-supervised learning. The choice for a specific category of algorithms depends on the problem to solve as well as the availability of ground truth for the collected data. Depending on the task to accomplish, machine learning can either be used to perform classification or regression. In the first case, we are trying to classify an instance as belonging either to a class A or B (binary classification), or several different classes (multi-class classification). In the case of regression, we want to guess a numerical, continuous, value with the lowest possible error between the real value and the predicted value. In this thesis, we generally focus more on classification, even though some regression approaches are described when discussing age estimation methods.

2.1.1 SUPERVISED MACHINE LEARNING

Supervised machine learning is the most intuitive approach and it has been used across many different fields, such as finance [55], weather forecast [56] or social media [57]. In supervised machine learning, a dataset called the *training set* is used to train the model. Each instance of the training set is made of a feature vector, which is simply a set of values that can be numerical (discrete or continuous), or textual. Each instance is associated with a label (or a numerical value for regression problems) called the ground truth. Using the training dataset, the model learns correlations between feature values and class labels. Once the model has been trained, it is tested on the *test set*. A *validation set* can be used in order to fine tune parameters before the model is ready for testing. This set is usually made up of a subset of the training set, and it is not used for training.

In the general case, the more data we have in the training set, the more accurate the model will be, as it will have a better chance at capturing the underlying probability distribution of the event that is being analyzed. However, it is possible to overtrain a model, which leads to a lower error on the training set, but a higher error on the validation and test set.

2.1.1.1 DISCRIMINATIVE MODELS

There exists a wide variety of models that can be used in machine learning. Models such as Logistic Regression (LR), Support Vector Machine (SVM) or Decision Trees (DT) fall into the category of discriminative models. They are opposed to generative models, such as Hidden Markov Models (HMM), or Naive Bayes (NB).

Discriminative models rely on observed data without making any prior assumption about its underlying probability distribution. At classification time, a discriminative model tries to find the conditional probability $P(Y|X = x)$ where, x is a specific instance in the dataset, and Y is a class label. At training time, the decision boundaries are built between classes based on the observed data only. At classification time, an unknown instance is fed to the model, and classified to the right class based on the learnt boundaries.

2.1.1.2 GENERATIVE MODELS

Generative models use the training data to estimate the prior probability $P(Y)$ and the likelihood $P(X|Y)$. The prior probability is generally the number of instances belonging to a certain class divided by the total number of instances. The idea behind this probability is to have a starting point by estimating the most naive probability of an instance belonging to a class. The likelihood carries the idea of the probability of the instances in the dataset

belonging to a specific class, assuming the class is known in advance. Generative models aim to find the posterior probability $P(Y|X)$, which can be expressed using Bayes' Rule:

$$P(Y|X) = \frac{P(Y)(X|Y)}{P(X)} \tag{2.1}$$

A conditional independence assumption between the instances can be added in order to simplify the computation from a chain of probabilities to a product of individual probabilities, such as in Naive Bayes. Generative models are interesting as they try to model each class of the problem by trying to find out the underlying density estimation of the data associated with each class. Discriminative models simply try to find boundaries between classes without trying to understand what each class represents. Depending on the problem to solve, either type of model could be used.

Supervised machine learning techniques require previous knowledge about the class label of each instance in the training dataset. In some applications, we have no previous knowledge of class distribution and we need to have a way of finding patterns in the data. This is when unsupervised machine learning techniques come into play.

2.1.2 UNSUPERVISED MACHINE LEARNING

Unsupervised learning methods aim to find patterns in a dataset where no explicit class labels are known. Clustering algorithms are the most common way of performing unsupervised learning. The goal of a clustering algorithm is to split a dataset into n different clusters based on different metrics.

2.1.2.1 K-MEANS CLUSTERING

Some clustering algorithms such as K-Means [58], rely on a similarity measure to form the clusters. In the specific case of K-Means, the number of clusters, k, has to be specified as an input. k initial centroids are chosen (usually at random), and a measure of similarity is used to assign each point in the dataset to a specific centroid. When dealing with numerical values, it is common to use the Euclidean distance as a similarity measure, where the distance between two data points is the difference of the square of the value of each of their features, one by one. Once all points have been assigned to a cluster, the new cluster centroids are computed by taking the average of all the points in each cluster. This process is repeated for several iterations.

The overall goal of K-Means is to maximize inter-cluster similarity which also leads to minimizing intra-cluster similarity. Once the similarity does not change much at the end of a cycle, or when a specific number of iterations has been reached, the algorithm stops. One of the main downsides of K-Means is that the number of clusters k has to be specified in advance. With no previous knowledge of the data, and when dealing with very abstract datasets, it is difficult to pick an optimal value of k. The overall process is illustrated in Figure 2.1.

2.1.2.2 DBSCAN AND HIERARCHICAL CLUSTERING

Different categories of clustering allow the formation of clusters without the need to define a number of clusters in advance, such as Density-based clustering such as DBSCAN [59]. DBSCAN uses the concept of density expressed as the number of points in a close neighbourhood of a point in order to form clusters. One of the main advantages of this type of clustering is that more intricate cluster shapes can be extracted because of the transition from

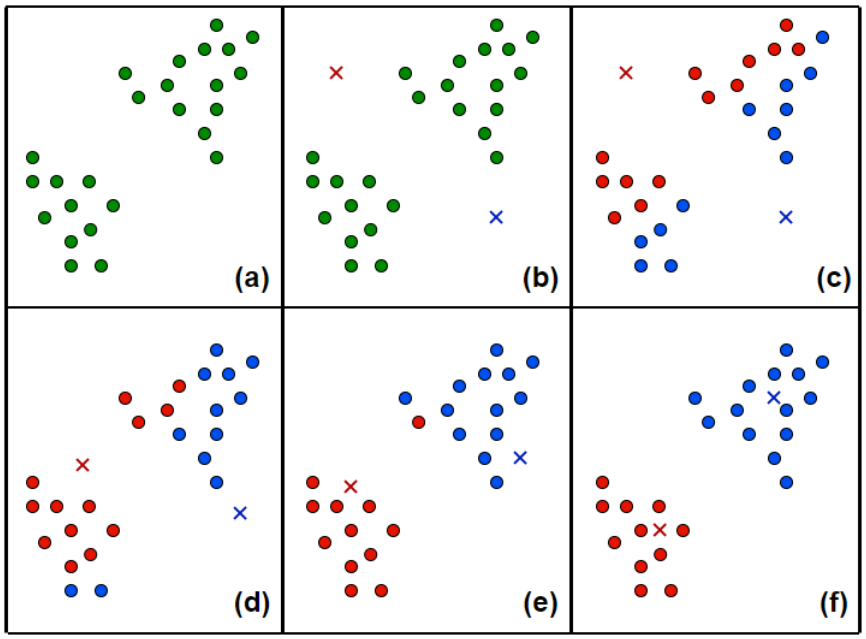

Figure 2.1 : Illustration of the K-Means clustering process. The centroids are random at first and slowly move in a way that minimizes inter-cluster similarity. © Rani Baghezza, 2022.

an absolute distance to a center to a local distance to a neighbour. Hierarchical clustering is another popular clustering method that begins with each point in the dataset being a cluster, and successively merging clusters into bigger clusters containing more instances [60].

2.1.2.3 NEURAL NETWORKS APPROACHES FOR CLUSTERING

We have briefly mentioned Neural Networks in the previous subsection as they are generally used for supervised learning. However, some NNs can also be used to tackle unsupervised learning tasks, such as autoencoders [61] or self-organizing maps[62]. We explore deep learning architectures relevant to this thesis later in this chapter.

Whereas supervised neural networks aim to find the optimal weights between neurons to predict the class of an instance as their output, autoencoders aim to find an approximation

of the identity function where the output is equal to the input. However, because of the use of various numbers of hidden layers, and a different number of neurons in the output layers, autoencoders often learn a lower dimensional, more compact representation of the input, which can be very useful for image recognition and compression.

While supervised algorithms work by having a full knowledge of the classes of the instances of the training dataset, and unsupervised algorithms operate on data without any knowledge of its class distribution, semi-supervised algorithms aim to bring the best of both world, and can be really efficient when only a small initial training dataset is available.

2.1.3 SEMI-SUPERVISED MACHINE LEARNING

The appeal of semi-supervised learning comes from the fact that acquiring labeled data usually requires the intervention of a human expert. Data labeling is often a long and tedious process, and it might not always be possible to label enough data to properly train a model. To train a semi-supervised learning model, a small batch of labeled data and a larger batch of unlabeled data are used. The actual training process depends on the algorithm, but the overall goal remains the same: the semi-supervised model should yield better performance than a supervised model trained only on the labeled dataset [63]. The semi-supervised learning process is illustrated in Figure 2.2.

Some of the methods used for semi-supervised learning are self-training and co-training, where self-training can be seen as a particular case of co-training with a single model [65]. In the case of self-training, the model is first trained on the labeled instances, just like a supervised model. The trained model is then used on the set of unlabeled instances, and a subset of these instances is used to enrich the labeled dataset. Typically, the unlabeled instances with the highest prediction confidence are chosen to be a part of the labeled dataset. Any supervised

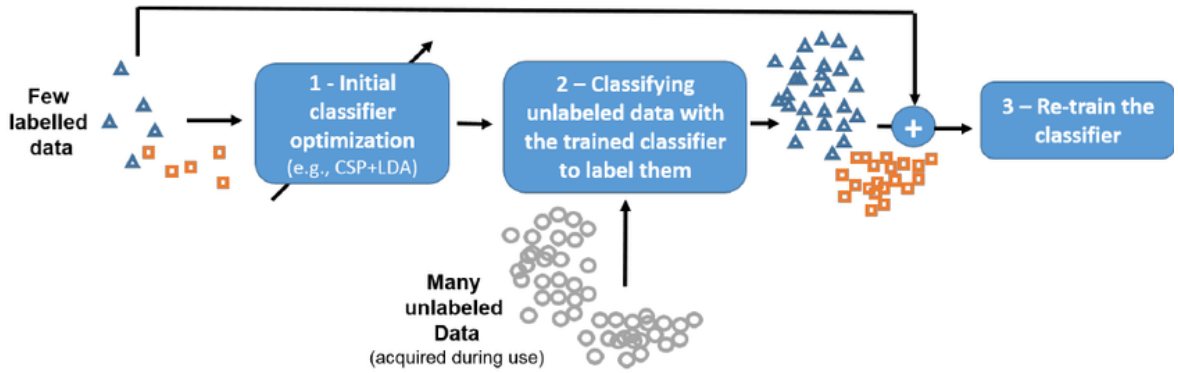

Figure 2.2 : Illustration of the semi-supervised learning process [64]. © 2015 IEEE.

classifier can be used for this process. However, if the classifier is flawed on the first part of the training, there is a high probability that subsequent learning steps using unlabeled data will reinforce these flaws. A high quality initial dataset is therefore needed, with a clear separation between classes. The instances making up that initial dataset should accurately represent the distribution of each class in order to reduce the possibility of compound errors later in the training process.

Co-training works in a different way. Two models are trained on the same initial labeled dataset, but each model uses a different subset of features. Each model then tries to predict the instances of the unlabeled dataset using the associated feature subset and the classifiers teach each other. The most confident predictions from one classifier are added to the labeled dataset to train the other classifier (and they are removed from the unlabeled dataset) and vice-versa. Each feature subset is called a view of the data. Co-training relies on two assumptions: each view alone should be sufficient to make good classifications if enough labeled data is available, and each view should be conditionally independent given the class label [63]. The first assumption guarantees that each model can be accurate enough on its own. If it is not met, self-training might be a better alternative as there simply might not be a way to split the data

in two views that would lead to a better overall learning performance. The second assumption guarantees that knowing the true label of an instance and one view does not affect what we will observe with the other view, and it is typically met if the data can be split in independent views.

Since the prediction of an instance's class label is based on confidence only, there is no way to be sure that the predicted class is equal to the ground truth. In order to progressively label the data as the system runs, active learning can be used [66]. Active learning assumes the presence of an end user who provides the label for a previously unknown instance. This label is then considered as being the ground truth, and the associated instance can be used to train the model in a semi-supervised context. Active learning can also be used for supervised learning, in order to check that the prediction of the classifier was actually correct. In a real-time system, active learning is a great method as it allows to continually add more data to the training set, and to delegate the labeling task to the end user, assuming we trust the users to make correct predictions.

2.2 EVALUATION METRICS

After a machine learning model has been trained, it is important to use good evaluation metrics in order to assess its performance, and eventually compare it with other models in a similar field [67].

2.2.1 TRAINING AND TESTING PROCEDURES

It is possible to evaluate the error rate or the success rate of a model on both the training and testing set. The error measured on the training set is called the resubstitution error, and is generally very optimistic as the model is evaluate with instances it already knows. Generally

speaking, a training set is used for training, a validation set is used for fine tuning, and a test set is used to compute the success rate of the model. Some methods use a combination of the error on the training and test set, such as the bootstrap method.

The most commonly used method for success rate estimation is the n-fold cross validation. The general idea behind that method is to split the full dataset into n equal parts. The model is then trained on *n-1* subsets of the data and tested on the last subset. Each class should be represented in the same proportions in the testing and training set in order to eliminate possible bias based on a class's frequency of apparition. We can used stratification to avoid class imbalance in the dataset, which is a sampling technique used to make sure that each dataset contains a distribution similar to the entire dataset. The n-fold cross validation process is usually repeated several times with random data segmentation and the average of all the runs is used as the success rate of the model.

Pushing n-fold cross validation to the extreme, we get to the leave-one-out method, where n is simply the number of instances in the dataset. The model is trained on all the instances except one, and tested on the last one. The success rate is then averaged over the n tests. This method is useful when only small datasets are available, and allows to exploit these datasets to the fullest, but it implies a very high computational cost for big datasets.

The bootstrap method can also be used for small datasets. For a dataset containing n instances, the idea is to sample the dataset n times with replacement to create a training dataset of size n that will inevitably contain repeated instances. The instances that have not been picked are used as the test set. The probability of an instance not being picked in the training set is:

$$(1-\frac{1}{n})^n \approx e^{-1} = 0.368 \tag{2.2}$$

Since the training set only contains about 63% of unique instances, the error computed on the test set will tend to be pessimistic. The error on the training set however, will be more optimistic. The final error is computed using a combination of the errors on the training and test set using the probability ratio (and its inverse) above, such that:

$$e = 0.632 \times e_{test} + 0.368 \times e_{training} \tag{2.3}$$

The process is repeated several times with different samples and the results are averaged to find a better estimation of the error.

As machine learning often comes down to evaluating the true probability distribution of the classes of interest, the same goes for performance prediction. The success rate of an algorithm is evaluated on a finite set of instances (the test set). Therefore, this success rate can only be an estimation of the true success rate of the algorithm when the number of instances tends to infinity (theoretically).

When dealing with NNs, it is possible to fall into overfitting if the model is too deep and complex. This concept is linked with a model's capacity, which is defined by Goodfellow et al. [19] as the model's ability to fit a wide variety of functions. Neural networks can be made as deep and as complex as the hardware allows, which allows them to approximate any function (assuming no limitations). However, if a model has a very high capacity, and the problem to solve is simple (classes are easily separable), or if the dataset is very small, it could lead the model to learn the training data, leading to outstanding results on the training dataset, but very poor generalization to different datasets.

A typical example is to use different polynomials of higher or lower degree to try and fit the data. If the degree is too low, the model will underfit the data, and the generalization will simply be inaccurate. If the degree is too high, every single point of data will lie on the output function of the model, and no globally encompassing pattern will be discovered, as the model is just *connecting the dots*. It is therefore important to choose the right model for the right problem. This process is illustrated in Figure 2.3.

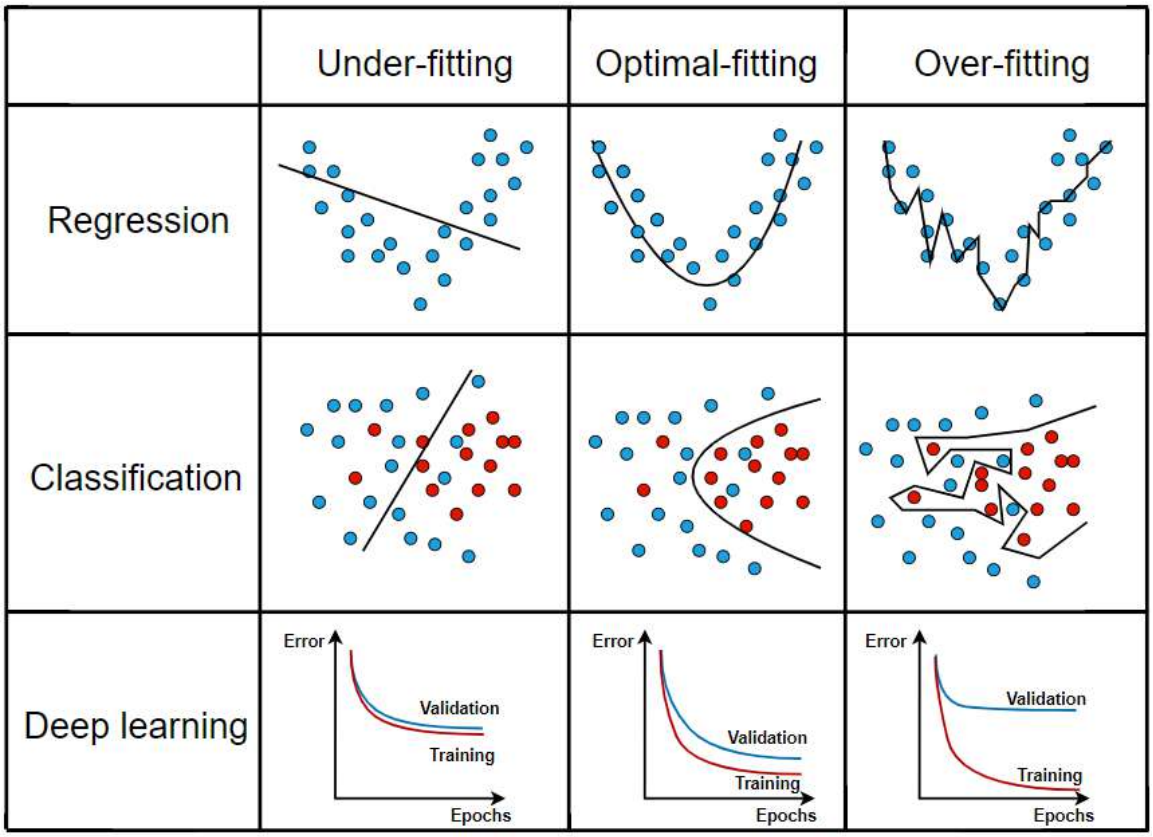

Figure 2.3 : Example of overfitting in different scenarios. Each point represents an instance, and in each case, the curve represents the model's boundary between classes. When the model is underfitting (left), it has not learned enough to separate classes or predict a correct value. When it is overfitting (right), the model is unable to generalize its knowledge to the validation dataset. © Rani Baghezza, 2022.

2.2.2 PERFORMANCE INDICATORS

In the case of a binary classifier, there are 4 possible relationships between the predicted label and the actual label: true positive (TP), true negative (TN), false positive (FP) and false negative (FN). True positives are instances correctly classified as being part of the *positive* class (the email is a spam, an intrusion is detected...), true negative are instances correctly classified as being part of the *negative* class, false positive are instances misclassified as being *positive* (when they are really negative), and false negative, instances misclassified as being *negative*. The following 2D matrix illustrates these outcomes (Table 2.1).

Table 2.1 : Summary of the relationship between predicted and real class label for binary classification.

		Predicted class	
		Yes	No
Actual class	Yes	True positive	False negative
	No	False positive	True negative

Using these outcomes, the overall success rate of the binary classifier can be expressed as the sum of the correctly classified instances (TP+TN) divided by the sum of all instances (TP+TN+FP+FN). Other performance indicators such as precision and recall can be used to get a better understanding of the proportion of true positive among all instances labeled as positive (precision), or among all actually positive instances, whether they have been misclassified or not (recall). The diagram below gives a good illustration for these indicators (Figure 2.4).

The F1-Score is sometimes used, especially in the field of information retrieval, as the harmonic mean of the precision and recall [68]. It is expressed as:

$$F_1 = \frac{2 \times recall \times precision}{recall + precision} \tag{2.4}$$

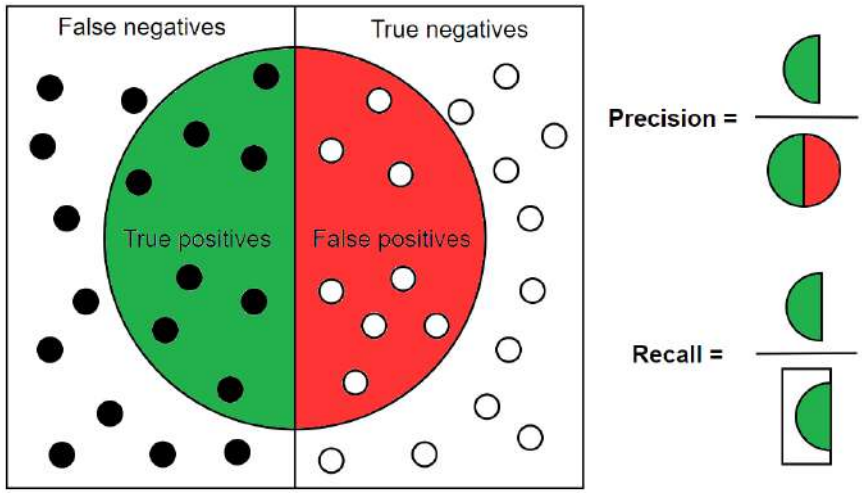

Figure 2.4 : Precision and Recall for binary classification. The circle represents the instances classified as positive by the model. An visual illustration of the precision and recall is given as well. © Rani Baghezza, 2022.

The most basic test for a machine learning algorithm is to evaluate it against a random classifier. If the model doesn't perform better than a random classifier (an algorithm that would predict the class of an instance purely based on the observed distribution of the classes in the training set, without the use of any features), it does not have any added value. Cohen's Kappa is an interesting indicator as it allows us to evaluate where the model lies in the spectrum from *random model* to *perfect model* (which would be a model with 100% success rate) [69]. It is expressed as:

$$k = \frac{p_o - p_e}{1 - p_e} \tag{2.5}$$

Where p_o is the precision of the model, p_e is the precision of a random estimator, and 1 represents the 100% precision of a perfect classifier. The values of Kappa are comprised between 0 and 1. A model scoring really low is very close to a random classifier, whereas

a model scoring closer to 1 will be closer to a perfect classifier. This indicator is useful to understand if the results are mostly obtained by chance, or if the estimator has actually learned how to discriminate between classes during training.

2.3 DEEP LEARNING

Deep learning is a general term that encompasses all neural networks based approaches, which are a bio-inspired family of machine learning models based on the way the human brain works. Each atomic unit of a neural network is a neuron, and neurons are arranged in successive layers. Once trained, the model takes a vector of features as an input, just like most supervised machine learning models, and based on these values, a specific sequence of neurons fires up throughout the layers until the final layer where a prediction is made. Because of the presence of a non-linear *activation function* at each neuron, neural networks are a non-linear machine learning model. The diagram below illustrates the difference between a problem where classes are linearly separable and when they are not, in the simplified case of instances with 2 numerical features represented as a point in 2D space (Figure 2.5).

Some models are linear by nature, but can be extended to the classification of non-linearly separable functions, such as Support Vector Machines (SVM), with the use of the kernel trick [70], where a kernel function is applied to map the features into a higher dimensional space where the instances can be linearly separated. Neural Networks are, by design, adapted to deal with non linearly separable classes. The atomic element of a Neural Network is the neuron, which is represented on the diagram below (Figure 2.6).

Each neuron a_i maintains a set of weights w_{ij} where each weight is associated with a neuron $a_{j=0,n}$ of the previous layer. A linear combination between all of the outputs of

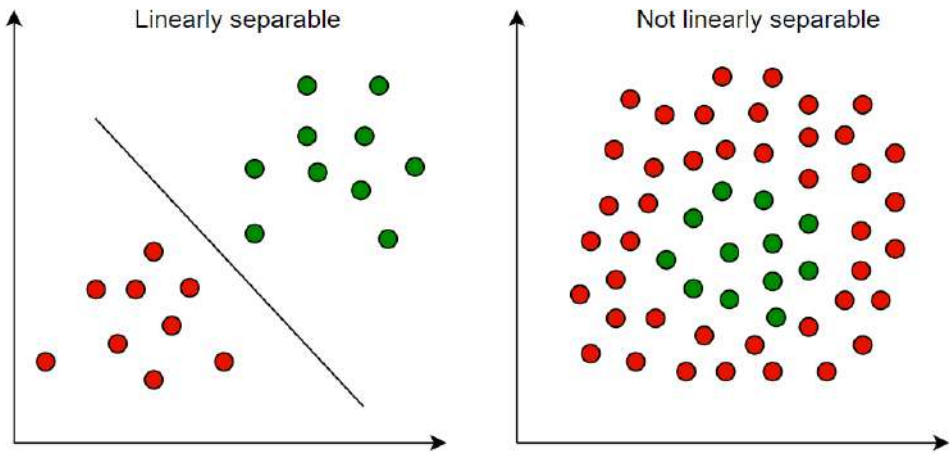

Figure 2.5 : Linearly separable and not linearly separable classes. © Rani Baghezza, 2022.

the previous neurons and their associated weights is fed to the neuron a_i. Each neuron also uses an additional weight, the *bias*, which allows to shift the activation function threshold. That activation function is then applied to the result of the linear combination. The activation function represents the non-linear part of neural networks: it maps any number to a value between 0 and 1. Different types of activation functions can be used in neural networks, such as the step function, the sigmoid function, the tanh function, or the ReLU (Rectified Linear Unit) function [71]. Linear functions can also be used, but they reduce the capacity of the neural network and limit it to solving more simple machine learning problems. The last layer of the neural network gives the output prediction.

Now, in order to train a neural network, we want to find the optimal set of weights at each layer, that allows the network to output a correct prediction for each new instance. In order to find the correct weights, we have to be able to estimate *how wrong* the current weights are, which is the role of the *loss function* (or cost function). A few different loss functions are generally used in machine learning models, such as the Mean Squared Error (MSE), or the Mean Absolute Error (MAE). The quadratic loss function is another valid

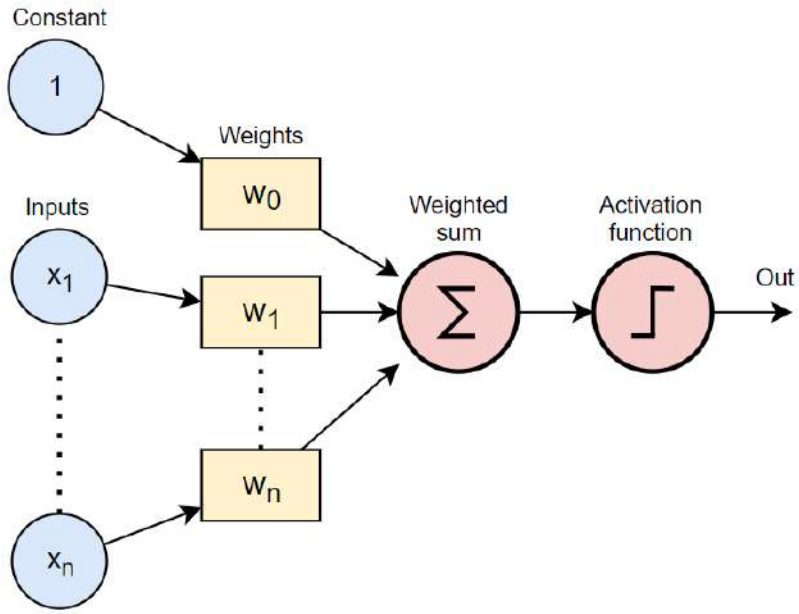

Figure 2.6 : Diagram representing a single neuron, using the step function as the activation function. © Rani Baghezza, 2022.

example; it computes the sum of the square of the differences between the predicted output and the expected output of the network. The objective is to minimize the value of that loss function during training through *gradient descent*.

Let us assume that we use the sigmoid function σ as the activation function of the network, and the quadratic function as the cost function. We use a_j^L to represent the activation value of the jth neuron of the Lth layer, assuming that the layer has n_L neurons. Similarly, we use a_k^{L-1} to represent the activation value of the kth neuron of the $L-1^{th}$ layer, assuming that the layer has n_{L-1} neuron. The ground truth vector containing all the correct activation values for the neurons of the output layer is called y, and the cost function is simply the square of the difference between each a_j^L and each y_j, assuming here that the Lth layer is the output layer [72]

$$C = \sum_{j=1}^{n_L-1} (a_j^L - y_j)^2 \tag{2.6}$$

Still reasoning on two consecutive layers, we express the weight between a_k^{L-1} and a_j^L as w_{jk}^L. The letter a both identifies the neuron and its activation value here. Each neuron has an associated bias, expressed as b_j^L for the neuron a_j^L. The diagram below illustrates this notation (Figure 2.7).

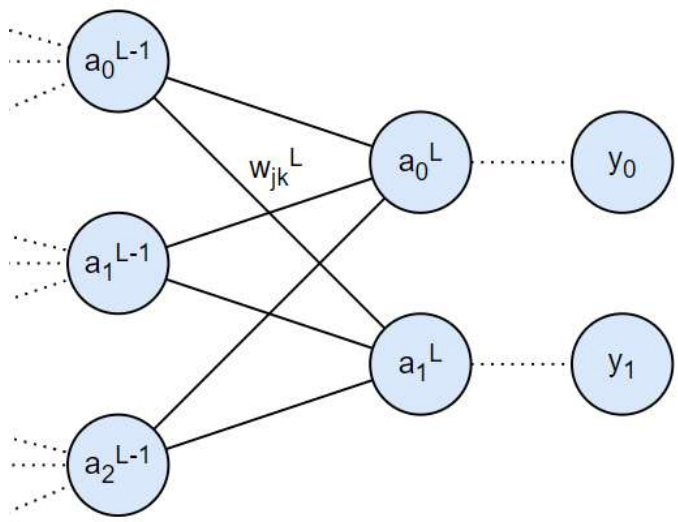

Figure 2.7 : Diagram of the last two layers of a neural network. y_0 and y_1 represent the true output values (ground truth). © Rani Baghezza, 2022.

We express the linear combination of weights and activation values fed as an input to the neuron a_j^L as:

$$z_j^L = \sum_{i=1}^{n_L-1} w_{ji}^L a_i^{L-1} + b_j \tag{2.7}$$

The activation function is then applied, giving the activation value of a_j^L.

$$a_j^L = \sigma(z_j^L) \tag{2.8}$$

The most basic approach to train a NN is to randomly initialize the weights, run all the instances through the network, compute the cost function over all the instances, giving us an idea of the global error of the network, perform the gradient descent process and correct the weights through backpropagation. This process can be repeated n times until the cost stabilizes around a local minima. The main drawback is how computationally intensive this task gets as the training set size increases. Other alternatives methods are to use mini-batches of randomly selected training instances (stochastic gradient descent [73]), or to perform gradient descent for each training instance.

Once the cost function has been computed, its gradient is computed with respect to the weights of the neural network in order to know how each weight affects the error. In order to make the model more accurate, we need the predictions to get closer to the ground truth which means that we have to minimize the cost function. By considering the negative of the gradient, we can find the direction to follow in order to get closer to a local minima of the cost function. Each component of the gradient is associated with a specific weight in the network, and at each learning step, each weight is modified by the product of its associated gradient component and a *learning step* α comprised between 0 and 1. This step allows to control the rate of learning, and reduce the probability of missing a local minima by taking too big a step in the gradient descent. Using all the notation above, the derivative of the cost function C with respect to a weight w_{jk}^l is expressed as:

$$\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1} \sigma'(z_j^l) \frac{\partial C}{\partial a_j^l} \tag{2.9}$$

Where a_k^{l-1} is the activation value of the neuron of the previous layer, connected to a_j^l by the weight w_{jk}^l. $\sigma'(z_j^l)$ is simply the derivative of the sigmoid function (or any other activation function that might be used instead), and (z_j^l) is the linear combination from equation 2.12. The third term of the derivative is itself calculated based on the parameters from the next layer, such that:

$$\frac{\partial C}{\partial a_j^l} = \sum_{j=0}^{n_{l+1}-1} w_{jk}^l \sigma'(z_j^{l+1}) \frac{\partial C}{\partial a_j^{l+1}} \tag{2.10}$$

In the specific case of the last layer, still assuming we are using the quadratic loss function, the derivative of the cost function with respect to a_j^l is simply expressed as:

$$\frac{\partial C}{\partial a_j^l} = 2(a_j^l - y_j) \tag{2.11}$$

In some cases, the factor 2 is simplified by using half of the quadratic loss function as the cost function instead. The inputs, activation values and output are known from the previous step of the learning process. If a single training example is used, the values are straightforward, otherwise, the average over the training batch is used. Once the error is considered acceptable, or when further backpropagation iterations do not show significant error reduction, the training phase is over.

Neural networks are a great tool to perform complex machine learning tasks because of their high capacity. The activation values of the hidden layers of a trained neural network can be considered as new, hidden features that the NN has empirically found through the training process. They are used in speech recognition, as well as image analysis in the form of Convolutional Neural Networks (CNN) which we discuss in the next section. Since the crux of this thesis is to analyze image data to perform profile recognition, neural networks

are a very good candidate algorithm. However, training a neural network in real-time, in an embedded system is a challenge due to the number of individual computations that have to be carried out. Training the network offline with similar data and using it in real-time to perform classification could be a good first approximation, as classification is just a series of linear combinations and application of the activation function.

2.3.1 CONVOLUTIONAL NEURAL NETWORKS

Convolutional Neural Networks (CNN) are particularly suited for image analysis and recognition tasks. They have been used with great success for face recognition [74], sentence modelling [75] or 3D object recognition [76]. CNNs typically take an 2-dimensional input, such as an image, and use several successive layers to extract feature maps from that input. They are designed in a way that allows the *feature learning* part of to be independent from the *classification* part, which allows for a lot of flexibility based on the task at hand. A CNN architecture example can be found in Figure 2.8.

The *convolutional* part of CNNs comes from the use of the convolution operation between an image as an input and a kernel (or filter) [19]. The result of that convolution operation is then subjected to a non-linear activation function, such as the ReLU function, and to a pooling operation that aims to reduce the dimension of the output whilst extracting significant features. These steps constitute a full layer of the network, and can be repeated as necessary until a set of feature maps is extracted and used in a subsequent classification process. The classification stage is usually a fully connected layer, analogous to Neural Networks, but as long as features are extracted, a number of other machine learning algorithm can be used for this step. The convolution operation is expressed as:

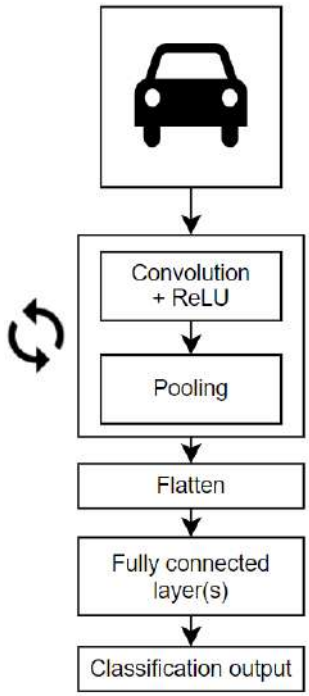

Figure 2.8 : Diagram illustrating the different layers of a Convolutional Neural Network. © Rani Baghezza, 2022.

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n) \tag{2.12}$$

Where S is the output, called the *feature map*. Since the input is of a higher dimension than the kernel, the commutative property of convolution is used to express it as:

$$S(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i-m, j-n)K(m,n) \tag{2.13}$$

Using this expression allows to slide the kernel over the input, perform the convolution operation, and repeat the process for each entry (i,j) of the feature map. Goodfellow et

al. [19] specify that the actual implementation in most CNN implementations actually use cross-correlation, rather than convolution, where the kernel is not flipped. It is expressed as:

$$S(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i+m, j+n)K(m,n) \tag{2.14}$$

For each cell of the feature map, the values of the input image and the kernel are multiplied one by one and added together. The kernel then slides over the image, in 1 or more pixel increments. When the step (or *stride*) is higher than one, 0-padding can be used in order to reduce the drop in dimensionality between successive layers. Indeed, if the kernel skips several pixels on each pass, the output feature map will have a smaller dimension. After the pooling operation, this dimension is further reduced, which reduces the number of successive layers that can be used in the CNN depending on the dimension of the input. We talk about *same padding* when the padding allows the output feature map to be of identical dimension to the input image, and *valid padding* when the output dimension is reduced compared to the input. The number of rows and column for padding depend on the image and kernel sizes. The illustration below gives an example of convolution without padding (Figure 2.9).

In the general case of image analysis, each cell will in fact be represented by 3 values (or more), that represent the RGB intensity of the pixel. A different kernel can be applied to each of these channels and the individual convolution results are added up and combined into a single channel output feature map. Once the feature map is extracted, a non-linear activation function is applied to each cell. When the ReLU function is used, all negative values are brought to 0, and any other value remains unchanged.

The following step is the pooling stage. Max-pooling is often used as it allows to highlight the dominant features by only keeping the maximum value in a small region of

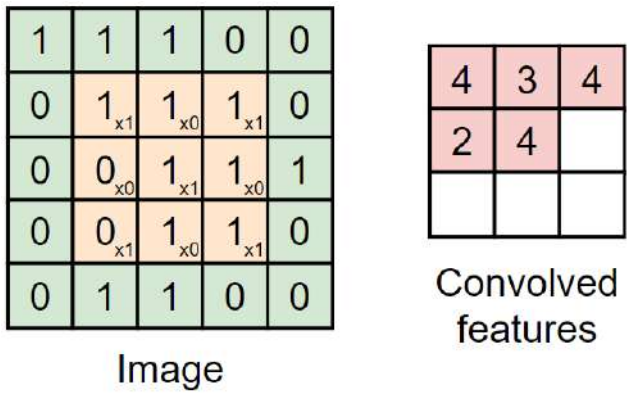

Figure 2.9 : Illustration of the convolution operation used between a 5x5 input image and a 3x3 kernel. The values of the kernel are displayed in red, in the bottom right corner of the highlighted cells of the image. © Rani Baghezza, 2022.

the feature map [77]. Pooling allows to reduce the dimension further, therefore reducing the computation complexity in deeper layers, but it also helps with the extraction of positionally invariant features, which is crucial for object recognition in images. The same object could be located in a different part of the image, or appear in a different orientation, which sparks the need for the extraction of translation and rotation invariant features.

The final element of the CNN is usually a full connected layer that takes as an input a flattened representation (1-dimensional) of the feature maps that have been extracted by the CNN. This layer behaves like a classical neural network and it is trained as such. The softmax activation function is generally used to determine the activation values of the final layer.

2.3.2 RECURRENT NEURAL NETWORKS

CNNs are suited to computer vision tasks, however, they are designed to operate on a single image at a time. Recurrent Neural Networks (RNNs) are a category of deep learning

algorithms designed to handle sequential data. They have been used in various fields such as language translation [78], text classification [79] and gait recognition [80].

At each time step t, the RNN takes a signal x_t as an input, as well as an internal state (or cell state) h_t as shown on Figure.2.11. This internal state stores the temporal information that is passed to the next step, updated, and so on until the last step of the network. The following equation summarizes how the cell state is updated at each step (2.15).

$$h_t = f_W(h_{t-1}, x_t) \tag{2.15}$$

Where h_{t-1} is the hidden state of the cell at the previous time step and x_t the input at the current time step. f_W is a function parametrized by the weights W which are learned throughout the training process using Backpropagation Through Time (BPTT) [81]. That function is the sum of two matrix multiplications and the application of a non-linear activation function, such as tanh, as shown in equation (2.16).

$$h_t = tanh(W_{hh}^T h_{t-1} + W_{xh}^T x_t) \tag{2.16}$$

Finally, another matrix is used to derive the output y_t:

$$y_t = W_{hy}^T h_t \tag{2.17}$$

RNNs are very versatile and can be used in a myriad of ways. In the case of language translation, they allow to use the entire context of the sentence in order to perform efficient translations rather than translating each word one by one, which would lead to inaccurate, out

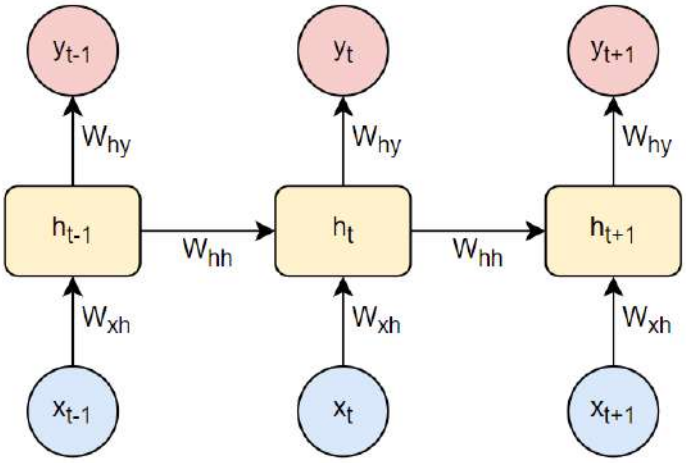

Figure 2.10 : Example of a RNN architecture with 3 time steps. x represents the input signal, h the cell state and o the output at each time step. © Rani Baghezza, 2022.

of context translations. In the case of gait recognition, we would like each input, at each time step, to be one of the frames of the gait sequence of interest. This is when combining CNNs and RNNs can become very efficient. A CNN can be used on each image in order to apply a chain of convolution and pooling operations to extract a final set of feature maps, that can be flattened into a 1D vector and fed at each time step of a RNN. The inputs are therefore a compact representation of the most relevant features in the image extracted by the CNN. From this point, the RNN can learn correlations between these vectors at each time step and extract temporal information that can lead to better gait recognition results.

2.3.2.1 LONG SHORT TERM MEMORY NETWORKS AND GATED RECURRENT UNITS

One of the main challenges RNNs have faced is the vanishing (or exploding) gradient problem. The core idea of neural networks also applies to recurrent neural networks: gradient descent is used to minimize a loss function in order to train the model. At each step, the

weights are updated using the gradient. However, the longer a sequence is, the more the gradient shrinks through backpropagation, leading the early layers to be left out of the training process and to have a marginal contribution at test time. This is also a challenge in very deep neural networks that has been addressed by the use of skip connections, which are briefly covered in the fourth chapter. In RNNs, this also leads the network to simply forget long-term dependencies: early time steps in the network are not taken into account, and the emphasis is put on the last time steps.

To counteract this, Long Short Term Memory Networks (LSTM) have first been introduced back in 1997 by Hochreiter et al. [82], even though more modern implementations are often used [83]. The main difference between LSTMs and vanilla RNNs is the use of a cell state and gates inside the cells that allow to control the flow of information. A LSTM cell contains three gates: a *forget gate*, an *input gate*, and an *output gate*. Gated Recurrent Units (GRU) are a more recent, alternative architecture that relies on two gates, a *reset gate* and an *update gate* [84]. Because of the inferior number of tensor operations at each step, GRU is generally faster to train than LSTM. Both models can be used for the same applications, however, the general consensus seems to be that GRU is more suited to smaller datasets, whereas LSTM is more suited to bigger datasets. An overview of the cell architecture for both models is shown in Figure 2.11.

Starting with the LSTM and working our way from the end of the cell to the beginning, the hidden state h_t^j for the j-th LSTM unit depends on the cell state c_t^j such that:

$$h_t^j = o_t^j tanh(c_t^j) \tag{2.18}$$

With o_t^j given by the output gate:

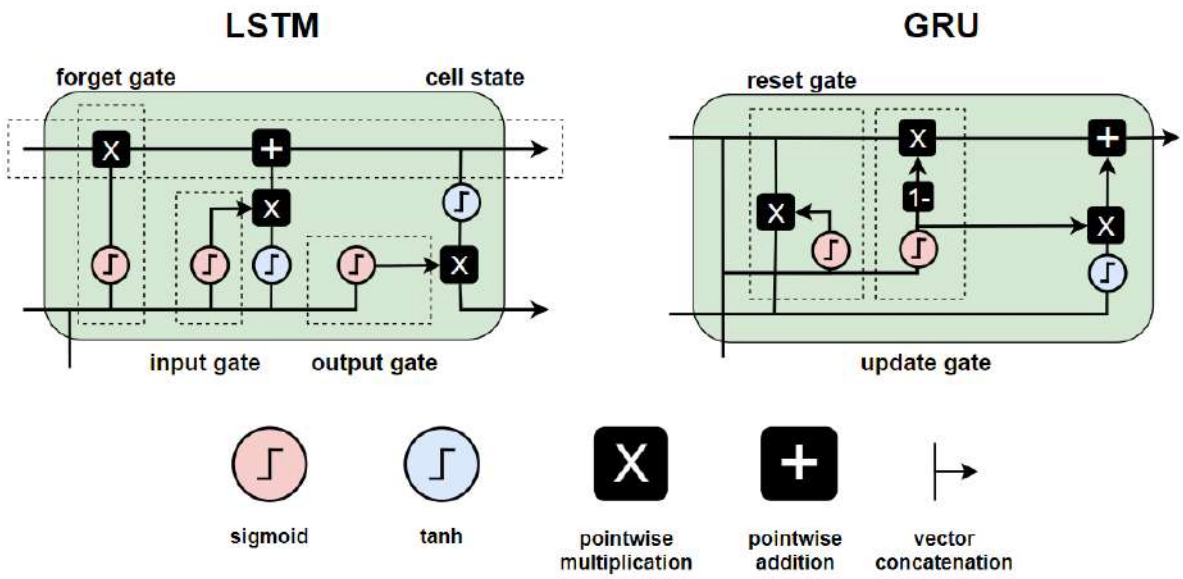

Figure 2.11 : Detailed view of LSTM and GRU cells, and of the flow of information within the cells. © Rani Baghezza, 2022.

$$o_t^j = \sigma(W_o x_t + U_o h_{t-1} + V_o c_t)^j \tag{2.19}$$

Where W_o, U_o and V_o are learned weight matrices, and σ is the sigmoid function, as illustrated in the diagram above. The memory cell c_t^j is itself affected by the influence of the forget (f) and input (i) gates, which modulate how much of the information from the previous step is forgotten, and how much of the new information coming from the signal at the current time step is passed along. The forget gate acts on the previous cell memory c_{t-1}^j, and the input gate adds in new memory content $\tilde{c}_t^j$:

$$c_t^j = f_t^j c_{t-1}^j + i_t^j \tilde{c}_t^j \tag{2.20}$$

The new memory content $\tilde{c}_t^j$ itself is computed using the input x_t and the previous hidden state h_{t-1}, which is the quantity that would represent the hidden state in a regular RNN, as expressed in equation 2.16.

$$\tilde{c}_t^j = tanh(W_c x_t + U_c h_{t-1})^j \tag{2.21}$$

Where W_c and U_c are learned matrices. The values of the forget and input gates are themselves expressed as a sigmoid operation on the signal at time t x_t, the previous hidden state h_{t-1} and the previous cell memory c_{t-1}

$$f_t^j = \sigma(W_f x_t + U_f h_{t-1} + V_f c_{t-1})^j \tag{2.22}$$

$$i_t^j = \sigma(W_i x_t + U_i h_{t-1} + V_i c_{t-1})^j \tag{2.23}$$

Where W_f, U_f, V_f, W_i, U_i and V_i are learned matrices. Following the flow of information from the beginning to the end, we can see that the LSTM first takes the previous cell state, previous hidden state, and current signal, forgets what needs to be forgotten, learns what needs to be learned, and combines all of the information in order to generate a new cell state and a new hidden state to be used for the next steps.

GRU works in a similar way, with the difference coming from the use of a *reset gate* and the absence of the *forget gate* and *input gate* duo, which allows the LSTM to control the portion of the state exposed to the update. Similarly, starting from the end, the activation h_t^j is computed by:

$$h_t^j = (1 - z_t^j)h_{t-1}^j + z_t^j \tilde{h}_t^j \tag{2.24}$$

Where z_t^j represents the update gate that handles how much of the unit updates its activation, and $\tilde{h}_t^j$ is the candidate activation. The update gate is computed using the previous activation state h_{t-1} and the current signal x_t:

$$z_t^j = \sigma(W_z x_t + U_z h_{t-1})^j \tag{2.25}$$

Where W_z and U_z are learned matrices. The candidate activation $\tilde{h}_t^j$ is computed as such:

$$\tilde{h}_t^j = tanh(W x_t + U(r_t \odot h_{t-1}))^j \tag{2.26}$$

Where r_t is the set of reset gates and $\odot$ is an element-wise multiplication. Depending on the value of the reset gates, more or less information from the previous states is discarded. The reset gate is computed using the previous activation and current signal as well:

$$r_t^j = \sigma(W_r x_t + U_r h_{t-1})^j \tag{2.27}$$

As described by Chung et al. [84], the main advantage of LSTM and GRU is that some information from the previous states is kept and mixed with information from the current state. This allows important features to be remembered for a long period of time, as opposed to RNNs, where the activation at each step is replaced. Even if the computation uses the previous step, the information is bound to be forgotten the longer the sequence is. The architecture of these cells also allows the creation of shortcut paths: if the network ends up learning that

a temporal step is not helpful to training, weights can be learned to skip new information coming from that cell (input and update gate), and the information can be forwarded to the next step. This also allows the error to be backpropagated in a more efficient way without the gradient vanishing.

2.4 CHAPTER CONCLUSION

In this chapter, we have introduced general definitions for machine learning, explored supervised, unsupervised and semi-supervised machine learning paradigms, and explored the metrics used to evaluate the performance of a model. We have then covered deep learning algorithms, as they are the more relevant to this thesis. We have seen how complex problems that deal with classes that are not easily linearly separable can benefit from the use of neural networks, and in the particular context of this thesis, CNNs and RNNs seem particularly suited. CNNs because they are designed to work with images, and RNNs, more specifically GRUs and LSTMs implementations, because of their ability to learn temporal relationships in a sequence.

Part III

Related work

CHAPTER III

RELATED WORK IN GAIT RECOGNITION

This part of the thesis is focused on reviewing the related literature in the fields of gait recognition and machine learning for profile recognition. This chapter focuses on gait recognition, starting with general definitions and a description of the standard process of individual gait recognition in the first section. The generalization of gait recognition to profile recognition, such as age and gender, is then discussed in the chapter of this part. A summary of the lessons learned throughout the two chapters of this literature review can be found at the beginning of Chapter 5.

3.1 INDIVIDUAL GAIT RECOGNITION USING MANUAL IMAGE PROCESSING

Gait analysis was defined by Whittle et al. [85] as the systematic study of human locomotion. It has been used in the field of healthcare for the recognition of gait-related health issues [86] and the early detection of falls in the elderly population [44]. Where gait analysis is generally focused on the study of the gait cycle of an individual to detect anomalies or specific patterns, gait recognition is aimed at recognizing, or re-identifying an individual based on their specific gait. It has been used for biometric identification [87], which can be used in airports for travelers identification, or for surveillance purposes [88].

Gait analysis techniques can be split into three main categories: wearable-based, video-based with markers on the subject's joints, and video-based without any markers. This thesis fits in the last category, as images only are used, and no markers are attached to the subject's joints. We will therefore exclude detailed analysis of approaches that fit in the first two categories in this chapter. Additionally, conventional approaches and deep learning methods

have been used to perform gait recognition. In an effort to keep the Related Work section of this thesis short, we will briefly review manual image processing-based approaches before diving in more details in deep learning based approaches. This section also allows to set the foundation necessary to the understanding of gait recognition.

3.1.1 GAIT RECOGNITION PIPELINE

The main difference between manual and deep learning based approaches for gait recognition comes from the image processing part. In all cases, the first step is to acquire images of pedestrians walking using a camera. In most cases, once the images have been collected and organized into sequences of images, background subtraction has to be performed in order to remove the information that is not useful for gait recognition. The image can then be segmented in order to either extract the silhouette or split the silhouettes in different sub-parts that can then be used for feature extraction. A set of features is then extracted for each image or each individual in the dataset, which allows to perform gait recognition or gait analysis depending on the context of application. Figure 3.1 below illustrates the general process of manual gait recognition.

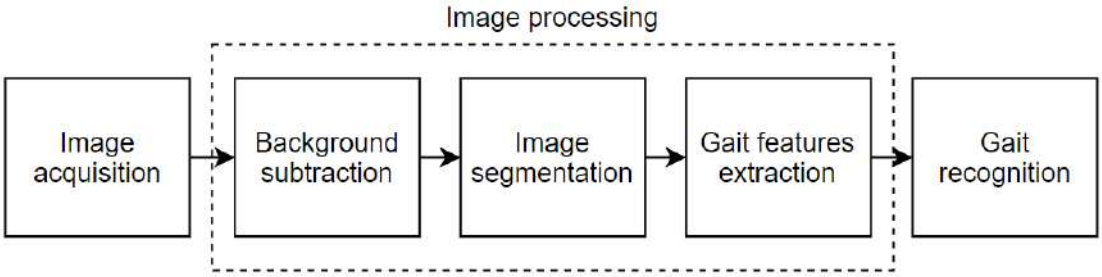

Figure 3.1 : Illustration of the gait recognition process by Leu et al. © Rani Baghezza, 2022.

3.1.2 IMAGE ACQUISITION

The three main types of images used in gait recognition have been RGB (Red, Green and Blue) images [89], RGB-D images [90] (RGB images with a depth component), and IR images (infrared) [91]. These different categories are illustrated in Figure 3.2 below. Limited research has been carried out on thermal images, which is discussed in the next chapter.

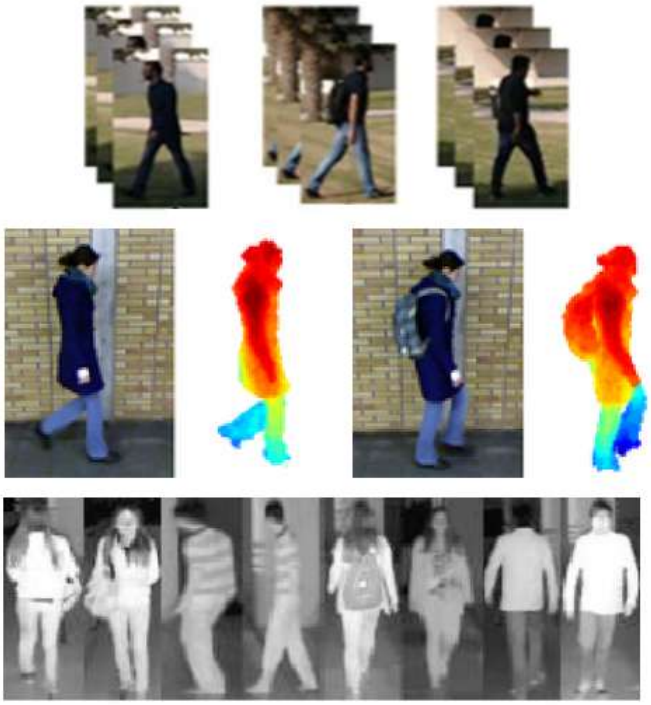

Figure 3.2 : Examples of RGB (top) [89], RGB-D (middle) [92] and IR images (bottom) [91] used for gait recognition. Creative Commons (RGB and RGB-D images) and © 2019 IEEE (IR images).

Depending on the experimental setup, the silhouettes can be captured at different view angles, such as 0°, 90° and 180° [93], -45°, 0° and 45° [94] or anywhere from 0° to 180° in 18° increments [95]. Depending on the task, gait recognition can be performed with better results on a specific view angle: Huang et al. [93] have found that for gender recognition,

better results were achieved at a 90° and 180° angle, whereas Lu et al. [94] have found that a 0 ° gave the best result for individual gait recognition.

3.1.3 BACKGROUND SUBTRACTION

When manual feature extraction methods are used, background subtraction is generally the next step after acquiring the images. The goal of background subtraction is to extract the silhouette from the image by removing the background in order to normalize the data, and remove the noise coming from the background that could interfere in the classification process.

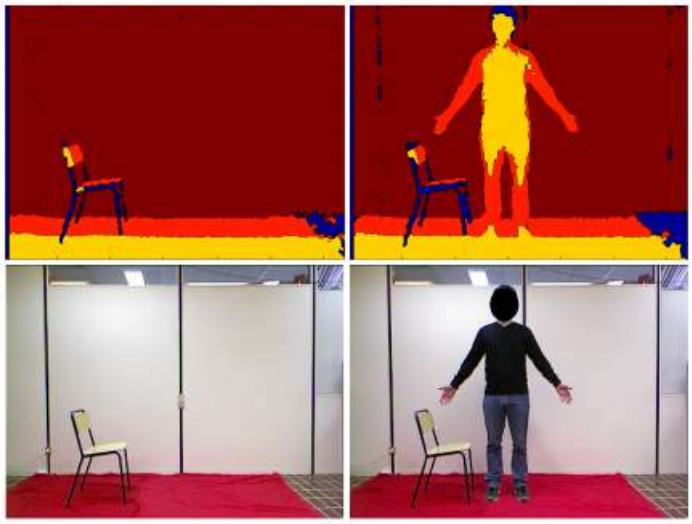

Figure 3.3 : Depth and RGB components of the calibration images used by Cippitelli et al. Left: Background frame of reference. Right: Front-plane pose image for background subtraction [96]. Creative Commons Image.

When data is collected indoors in a controlled environment, it is possible to perform static background subtraction, as shown in Figure 3.3. Cippitelli et al. have used a Microsoft

Kinect camera to collect RGB-D images, allowing them to perform background subtraction using a threshold-based method described in equation (3.1) [96].

$$FF(x,y) = \begin{cases} 0, & |DF(x,y) - BF(x,y)| < Th \\ 1, & otherwise \end{cases} \tag{3.1}$$

Each pixel's depth component in the current frame, $DF(x,y)$, is compared to the same pixel's depth value in the background frame $BF(x,y)$, which is captured before the subject enters the frame. If the difference between these values is above a certain experimental threshold Th, the pixel is considered as a foreground pixel, and belongs to the subject's silhouette.

When gait recognition takes place in an environment where the background is subject to changes, adaptive background subtraction techniques can be used [97]. Similarly to static background subtraction, the current image is compared to the reference image pixel by pixel in order to determine which part belongs to the background and which part belongs to the foreground. The difference in this case is that the reference image is updated at each step, and is made up of a weighted average of all the previous images in the sequence. This is a useful approach for gait recognition in a noisy environment. The weighted average is computed using equation (3.2).

$$A_t(x,y) = (1 - \alpha_b) * A_{t-1}(x,y) + \alpha_b * I_t(x,y) \tag{3.2}$$

Here, $A_t(x,y)$ is the reference image at time t, α_b is the learning rate, and $I_t(x,y)$ is the current frame at time t. A learning rate of 0.8 is used for the first 100 frames, and 0.0005 afterwards, possibly indicating that the learning rate is heavily dependent on the data used. Moreover, the authors have highlighted that this technique suffers from shadows and specularities (reflections).

3.1.4 FEATURE EXTRACTION

Once the silhouette has been extracted, various features can be computed on the image. Some methods, such as the one presented by Cippitelli et al. [96], use static rules about the human body called anthropometric ratios, to determine the position of the subject's joint. However, this approach is not relevant in our case as it emulates a marker-based approach.

Histograms of Oriented Gradients (HOG) have been a popular way of extracting interesting features in images such as edges [98]. The image is split into a grid of regions of a specific size. A 1-D centered mask is used on each pixel ([-1,0,1]), in both horizontal and vertical directions, computing the color gradient at each pixel using RGB data. The 0-360° domain is split into n different bins, and rectangular or elliptical cells are defined, over which a histogram of the gradients is computed. The dominant gradient in the histogram is chosen as the main orientation of that cell. The image below shows the orientation gradients extracted from an image to better illustrate the process (Figure 3.4).

One of the most popular features extracted for gait recognition purposes has been Gait Energy Image (GEI) [100]. Considering a sequence of bounded boxes of the same size containing the gait of interest, GEI simply creates a single signature that averages the values of the pixels over the sequence, as illustrated in equation (3.3).

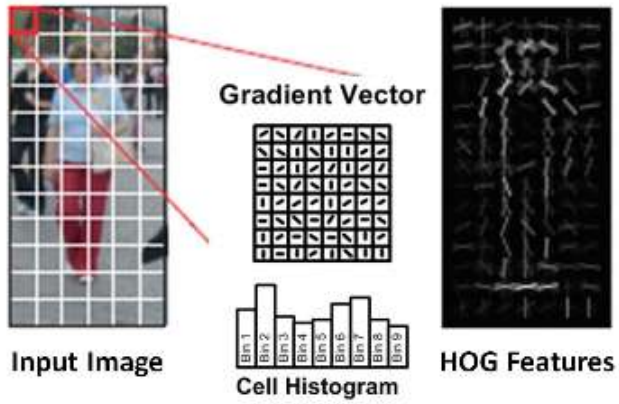

Figure 3.4 : Depiction of the information extracted by the Histograms of Oriented Gradients [99]. © 2017 IEEE.

$$G(x,y) = \frac{1}{N}\sum_{t=1}^{N} B(x,y,t) \tag{3.3}$$

Where $B(x,y,t)$ is the walking binary silhouette sequence, N the total number of frames in the sequence, t the frame number, and x and y the coordinates in the 2D frame. A representation of the final product of the GEI after applying it to a sequence of frames can be found in Figure 3.5.

When considering a sequence of images rather than a single image, Optical Flow (OF) is another popular method [101]. In an image, the optical flow is the measure of the estimation of the displacement of an object between two consecutive frames. It relies on the assumptions that any pixel at a time *t* and a position *(x1,y1)* will appear at time *t+1* at a different position *((x2,y2)*, and that pixels in a neighborhood have a similar motion. The optical flow is represented as a

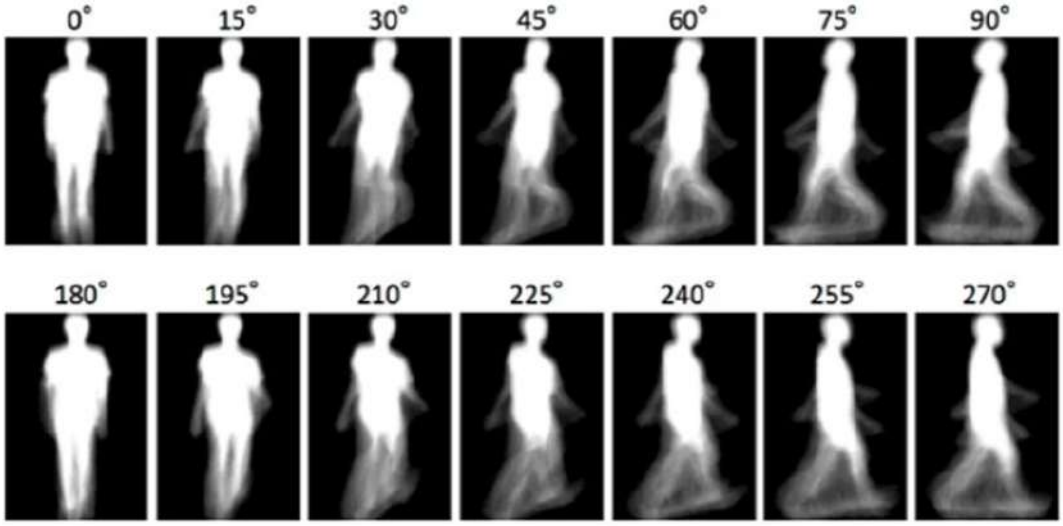

Figure 3.5 : Gait Energy Image at different view angles [89]. © 2021 IEEE.

matrix of the displacement vectors of each pixel of the images for each frame transition, as a measure of the motion in the footage, as illustrated in Figure 3.6.

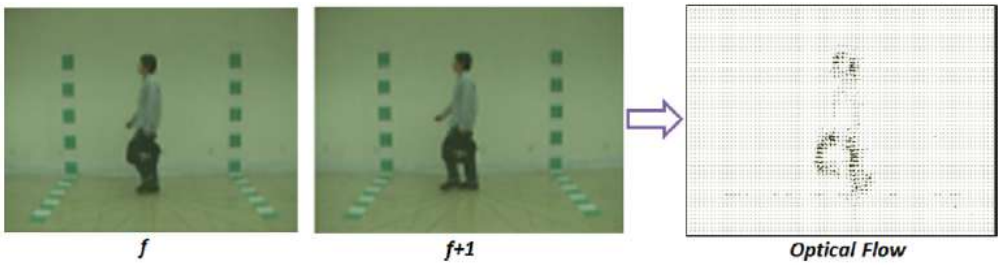

Figure 3.6 : Illustration of the optical flow calculated between two frames on the CASIA dataset [102]. © 2018 Elsevier

Many more features have been used for gait recognition, such as Gait Silhouette Volume (GSV) [103] or various color and texture features that are far too numerous to enumerate here [104].

3.1.5 CLASSIFICATION

Once the images have been collected, and various features have been extracted, the recognition step can take place. The fundamental idea is the recognition of underlying patterns specific to a person's way of walking, allowing us to tell that subject apart from other people. In order to do so, two main approaches have been used: non-learning based methods and learning-based methods. The former category is based on the use of various distance and correlation measures between the instance to classify (test dataset) and the training dataset [105]. The latter category can be divided further in conventional machine learning approaches and deep learning methods.

3.1.5.1 NON LEARNING-BASED APPROACHES

Distance-based methods include the use of the Euclidean Distance [96], described in equation (3.4) or the Manhattan Distance [106], described in equation (3.5).

$$D(A,B) = \sqrt{\sum_{t=1}^{n} (x_i - y_i)^2} \tag{3.4}$$

$$D(A,B) = \sum_{t=1}^{n} |x_i - y_i| \tag{3.5}$$

Dynamic Time Warping (DTW) has been used to find the closest match between data and a known pattern in a time series even when the data and pattern do not match exactly [107]. It has been widely used for speech recognition as well as for gait recognition [87].

Considering the time series S and the template T such that $S = s_1, s_2, ...s_n$ and $^T = t_1, t_2, ...t_m$, we can build a *n-by-m* grid where each point (i, j) is the intersection between a point of S and a point of T. The distance between each point of both series is computed with optional boundary conditions to reduce the search space. We want to find the optimal sequence of grid points $W = w_1, w_2, ...w_k$ called a warping path, such that the cumulative distance over the whole path is the lowest. The problem can be expressed as a minimization over potential all warping paths p:

$$DTW(S,T) = \min_{W}[\sum_{k=1}^{p} \delta(w_k)] \tag{3.6}$$

3.1.5.2 LEARNING-BASED APPROACHES

When it comes to learning-based approaches, different algorithms have been used. Working on sequences of frames of a user climbing up and down some stairs, Snoek et al. [97] have used a probabilistic Bayesian sequence estimation technique to model the position of the feet when they are not visible on the image, and to predict the probability of each foot appearing at a certain position on the following frame. When the feet are visible, HOG can be used to extract the edges, and this data can be fed to a Hidden Markov Model (HMM) classifier [108]. HMM belongs to the category of generative machine learning models, which have been described in the previous chapter.

Bajwa et al. [109] have used a combination of different models to perform gait recognition. RGB images are collected, converted to gray scale images, the background is removed, features are extracted, and three learning setups are compared: using a Support Vector Ma-

chine (SVM) only, using a combination of SVM and k-Nearest Neighbors (k-NN), and a combination of SVM, Neural Networks (NN) and k-NN. As opposed to the HMM mentioned in the previous paragraph, SVM is a discriminative model that aims to find the maximal margin hyperplane that separates the data from two different classes. When several classes are present, a common way of using SVM is to perform 1 vs. all classification, where the instances of the class of interest (a particular individual in the case of gait recognition) are opposed to the instances of all other classes. The k-NN algorithm simply compares the distance between an instance and all of the other instances in the dataset by computing the Euclidean distance between the features of each instance, one by one.

3.1.6 CONCLUSIONS ON MANUAL IMAGE PROCESSING FOR GAIT RECOGNITION

The straightforward way of performing gait recognition on images, which implies image acquisition, background subtraction, feature extraction and a classification step has been thoroughly explored in the scientific literature. However, this approach has a few drawbacks, the first of which is the manual aspect of it. We keep in mind that this thesis aims to achieve a profile recognition system that could be used in real-time in smart cities. Given that constraint, it would not be possible to ensure that image processing could be performed in a consistent way at runtime, as no human could see the images after they have been processed, and before they are fed to the classifier. It has been seen in some instances that experimental thresholds were found and used [96], [97], which require some kind of back and forth between the data and a human expert. There is no guarantee that the experimental threshold would work for unseen instances.

Moreover, performing profile recognition in an urban environment adds the challenge of a dynamic background. Even though some dynamic background subtraction techniques

have been successfully implemented in indoors environments [97], the amount of variability in the data collected in a city could be an issue. The use of thermal cameras in this thesis also implies that the color of the images collected is bound to vary throughout the day as the temperature of the scene fluctuates. Many different factors can lead the background to change, such as cars, pedestrians on the opposite sidewalk or people crossing the street.

Additionally, silhouettes tend to be contained in bounded boxes of uniform size to extract features such as GEI. In an urban environment, with no control over the distance between the cameras and the silhouette, we are faced with a high variability in terms of silhouette size. Pre-processing all of the images to subtract a dynamic background, extract a silhouette, normalize it and extract features from it would prove difficult without a human actor overseeing the process. Moreover, there is no guarantee that a set of fixed features would perform well on such high variability data. All of these factor have led us to move away from manual image pre-processing and handling, and towards deep learning based solutions.

3.2 DEEP LEARNING-BASED GAIT RECOGNITION

Deep learning algorithms have generally been used for more complex tasks that conventional machine learning methods were unable to accomplish [19]. Image recognition has been one of the first fields benefiting from the use of deep convolutional neural networks [110].

In the field of gait recognition, deep learning algorithms have gained a lot of popularity in recent years. Since 2019, most gait recognition methods used in the literature have been based on deep learning (Figure 3.7), overtaking conventional machine learning-based methods [111]. Bashir et al. [51] have highlighted the performance drop of classical approaches when performing gait recognition in an uncontrolled environment.

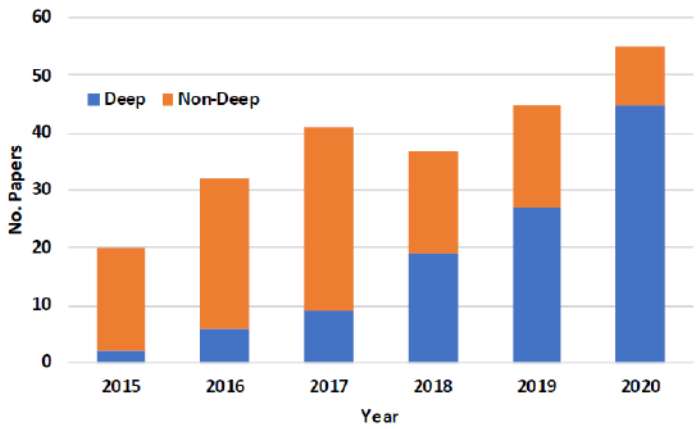

Figure 3.7 : Evolution of the number of deep-learning vs. non-deep-learning based papers for gait recognition over the past years [111]. © 2021 IEEE.

We found the taxonomy for deep learning-based gait recognition presented by Sepas-Moghaddam et al. [111] to be particularly clear and complete. This subsection therefore follows along the general organization of this survey paper with additional insights and references to work that was not covered in the paper. We begin by mentioning the body representation based on the collected images, followed by the way a sequence of frame is represented to take into account the time component before moving to the way features are extracted from the images, and we conclude this section with the variety of deep learning algorithms that have been used for gait recognition, as well as their performance and their limitations.

3.2.1 BODY REPRESENTATION

The starting point of deep learning-based recognition methods is similar: images are acquired using a camera. Many different datasets have been built and used for image-based gait recognition, such as the SOTON database [112], containing indoors and outdoors gait data for 115 subjects, the CASIA-A dataset [113], collected outdoors on 20 subjects walking

at different angles, the USF HumanID gait dataset [114], containing 122 subjects walking outdoors in an elliptical path to account for different angles, the CASIA-B dataset [49], which is one of the most used gait dataset in the literature, containing indoors gait data from 11 different view angles varying from 18 ° to 180 ° for 124 subject, the OU-ISIR dataset [50], containing gait sequences for 4007 subjects captured at 55 °, 65 °, 75 ° and 85 ° angles or the TUM GAID dataset [115] which includes RGB, depth and audio data for 32 subjects under different conditions (walking, walking with a backpack, and walking with shoe covers). Sample images of these datasets can be found in Figure 3.8.

Figure 3.8 : Sample of some of the most used gait datasets in the literature. From top left to bottom right: SOTON [112], CASIA-A [113], USF HumanID [114], CASIA-B [49], OU-ISIR [50], and TUM GAID [115]. © 2004 Springer Nature, © 2013 Elsevier, © 2003, 2005, 2006, 2012 IEEE.

Two main categories have been identified in terms of body representation: silhouettes and skeletons. Silhouettes are generally easier to extract using background subtraction techniques [97], and allow the use of image analysis techniques, such as optical flow, in order to perform

gait recognition. Extracting the silhouettes over a sequence of images allows to extract information about the way the subject walks, such as their speed, cadence, the length of their stride, and various other factors. However, clothing and additional accessories can interfere with gait recognition, as they represent additional noise on top of the silhouette.

On the other hand, skeletons can be extracted from the images, particularly when RGB-D images are used [115]. The depth component of the image allows the extraction of the joints and segments of the subject, leading to the reconstruction of the skeleton, then used for model-based gait recognition. When no depth component is available, pose-estimation methods have been used [116]. Skeleton-based gait recognition approaches have been found to be more resilient when different view points are used [117], as opposed to silhouette-based methods.

3.2.2 TEMPORAL REPRESENTATION

Generally speaking, a sequence of images is used for each subject in the dataset, leading to the matter of handling the temporal aspect of the gait. Sepas et al. [111] have split the way temporality has been handled for deep-learning gait recognition in two distinct categories called *templates* and *volumes*. The former aims to synthesize the information contained in all of the frames of the sequence into a unique map, whereas the latter aims to learn the temporal relationship between the frames by keeping them separate and using learning-based methods.

One of the popular temporal templates used is none other than GEI, which has been illustrated in the previous section [100]. Other templates have been used in the literature, such as Chrono-Gait Image (CGI) [118], which extracts the contour of each silhouette in each image in the sequence and uses a color mapping function to encode each contour in the sequence into a single CGI, Frame-Difference Energy Images (FDEI) [119], which has been useful in the

case of incomplete silhouettes, Gait Entropy Image (GEnI) [120], which computes Shannon's entropy [121] for each pixel of each image over an entire gait cycle, creating a single template containing all the temporal information of the cycle, and Period Energy Image (PEI) [122]. A summary of these different temporal templates can be found in Figure 3.9.

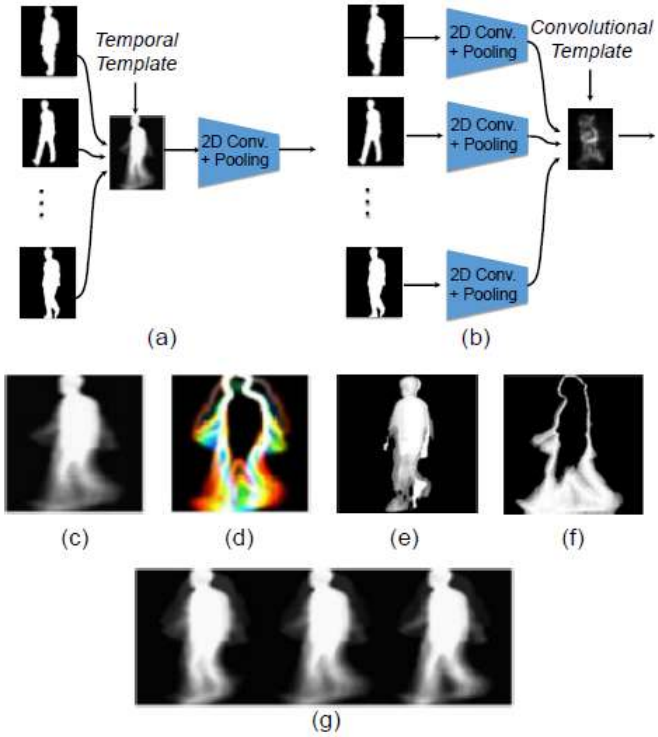

Figure 3.9 : Illustration of the most popular temporal templates used in the literature [111]. (a) illustrates the scenario where a template is computed before being fed to the deep learning network, as opposed to (b), where the convolutional template is computed inside the network. (c) is an example of GEI [100], (d) CGI [118], (e) FDEI [119], (f) GEnI [120] and (g) PEI [122]. 2021 IEEE

Volume approaches are split in two categories, the first one being the use of Recurrent Neural Network (RNN) to learn temporal information between a frame and the previous and following frames in the sequence [123]. The second approach is to create a single 3-D tensor, where the third dimension captures the temporal information of the sequence of images,

and to use models such as 3D Convolution Neural Networks (3D CNNs) [124] or Graph Convolutional Networks (GCNs) [125] on that tensor.

3.2.3 FEATURE REPRESENTATION

When it comes to feature representation, there are two main avenues: *global* and *partial* feature representation. In systems where silhouettes or skeletons are extracted, and each silhouette/skeleton is treated as a single unit, we talk about global feature representation. When deep learning is used, each frame will be used as an atomic unit in the network, and feature maps will be learned on the entire frame.

These approaches are opposed to partial feature representation, where each image is divided in sub-regions in which features are computed, and where the results are combined using different methods in order to reach a global gait recognition solution. Different division strategies can be used, such as patch-based division, horizontal bins, or component-based division. The main benefit of using partial representation has been illustrated in original work by Sepas et al. [126]; extracting partial representations and learning relationships between these relationships allows the model to be more robust to occlusions, camera angles, and variation in clothing. Other models have been used to learn these relationships, such as attention-based networks [127], capsule networks [128], and fully-connected layers [129]. The architecture used in [126] can be found in Figure 3.10.

The first part of the network behaves as a classical CNN and extracts spatial features through 6 convolutional layers and 2 pooling layers for each image in a sequence associated to a subject, and for each viewpoint available for that subject. The results are then aggregated in a Gait Convolutional Energy Map (GCEM), which is the mean of the spatial features at all time steps. This map is then split into different bins which are processed through a fully connected

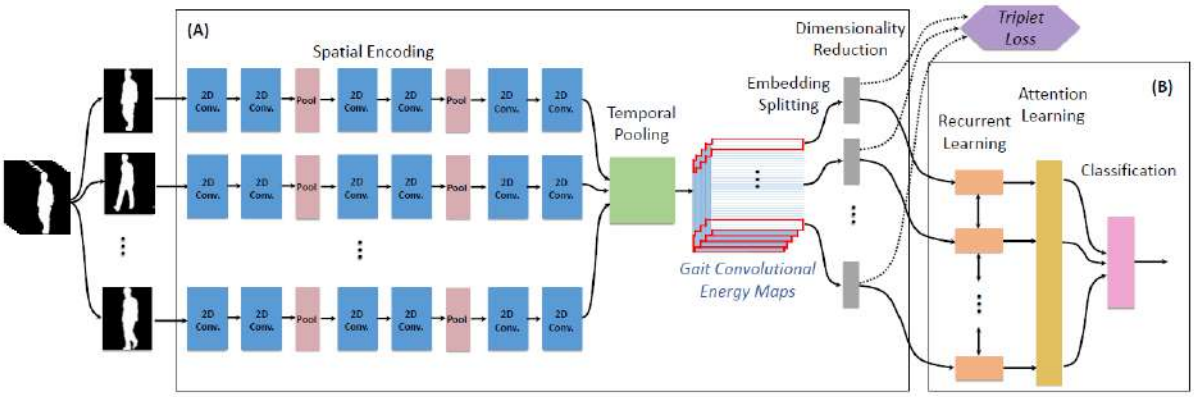

Figure 3.10 : Architecture of the partial feature-based deep learning gait recognition network presented [126]. © 2020 IEEE.

layer to perform dimensionality reduction, allowing the subsequent network to be shallower, thus cutting down computation time. A Recurrent Neural Network (RNN) is then used to learn relationships between the extracted partial features (bins), and an attention layer helps emphasize learning for the most important bins. On average, this method has outperformed previous state-of-the-art methods on the CASIA-B dataset with carried bags and different clothing on the subjects, as well as on the OU-MVLP dataset, demonstrating the robustness of this method to changes in viewpoint and clothing.

3.2.4 DEEP LEARNING MODELS

Many deep learning models have been used for gait recognition. In this section, we give a general overview of the most commonly used models in the field with a brief description for each model.

CNNs have mainly been used for image-based tasks, as described in the previous chapter. Sepas et al. [111] have reviewed the architectures and input sizes used in CNN models

throughout the literature, and they have found that most models operate on inputs of size 64x64 and 128x128, with a slight edge in terms of results for the latter, but an additional computational cost brought by the bigger image size. The total number of layers has been comprised between 5 and 16, with anywhere from 2 to 12 convolutional layers, the rest being made up of pooling and fully connected layers. Classical CNNs have also been extended to 3D Convolutional Neural Networks (3D CNNs) and used for gait recognition. Instead of a single image as an atomic instance, a stack of frames in the form of a 3D tensor is fed to the network [124]. They have proven robust to changes in viewpoints and the appearance of the subjects, however, they lack flexibility when it comes to variable length sequences.

Another way to include the temporal aspect of the subject's gait is to use Recurrent Neural Networks (RNNs). They have been used to learn relationships between partial features as described in the last section [126]. Feng et al. [130] have used LSTMs to perform gait recognition after extracting features for each image using a CNN. Liu et al. [79] have used a memory neuron network to perform gait recognition on the CASIA-A and CASIA-B datasets.

In terms of deep learning, another popular model has been Deep Auto-Encoders (DAE) [131]. They are a specific type of multi-layered neural network that are made of an encoder and a decoder part as shown in Figure 3.11. The encoder takes a specific input and aims to extract bottleneck features by encoding this input into a different feature representation. From these bottleneck features, the decoder aims to learn a way to reconstruct the original input with the highest accuracy possible. The encoded features can then used for classification.

Auto-encoders are useful when dealing with the common issue of handling multi-view data in gait recognition: in real-life datasets, the angle of the subject to the camera will vary greatly from subject to subject, and it is important to build gait recognition models that can

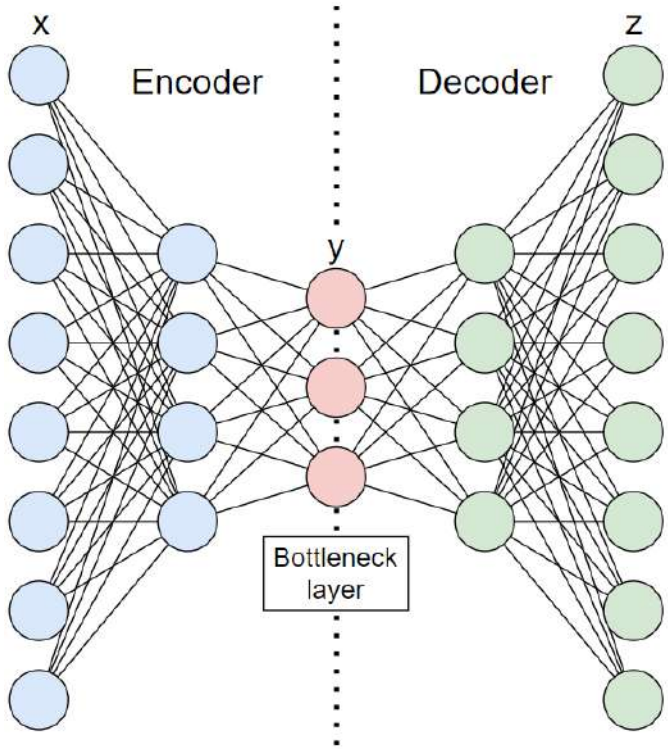

Figure 3.11 : Deep Auto-Encoder architecture. © Rani Baghezza, 2022.

provide a good recognition accuracy regardless of the angle between the camera and the subject. Yu et al. [132] have capitalized on the ability of auto-encoders to extract features in a different space by using an auto-encoder with 7 fully connected layers for view angle invariant feature extraction. The CASIA-B dataset used in this paper contains data for all subject at angles ranging from 18° to 180°, allowing the authors to stack encoders that learn to successively translate the data to a 90° angle observation, leading to better results. The GEI is extracted before being fed to the auto-encoder. An overview of the model is illustrated in Figure 3.12.

Deep Belief Networks (DBN) [133] have also been used for gait recognition. They differ from the models mentioned above as they belong to the category of generative models. A DBN is a probabilistic model that mixes directed and undirected connections between nodes.

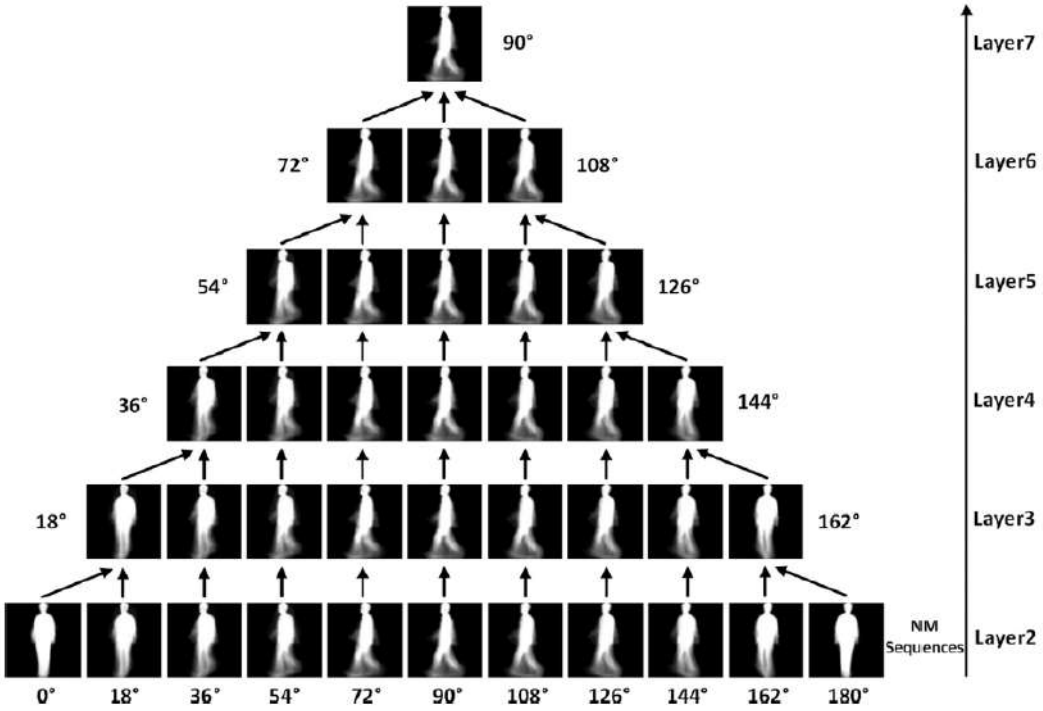

Figure 3.12 : Auto-encoder architecture used by Yu et al. to perform view invariant feature extraction for gait recognition [132]. © 2017 Elsevier.

It is built by successively stacking Restricted Boltzmann Machines (RBM) to improve the performance of the model. Each RBM is composed of a visible layer and a hidden layer containing units, which are associated with a weight and a bias term. As a generative model, DBN does not aim to find the boundary between classes, but it aims to capture and learn the probabilistic distribution of the instances in each class, which is captured in the value of the weight and biases after training. An illustration of the DBN architecture can be found in Figure 3.13.

Benouis et al. [134] have used DBN to perform gait recognition on the CASIA-B dataset. Since DBN is not specifically built for 2D data such as images, features have to be extracted from the images and encapsulated into a 1D vector that is then fed to the model

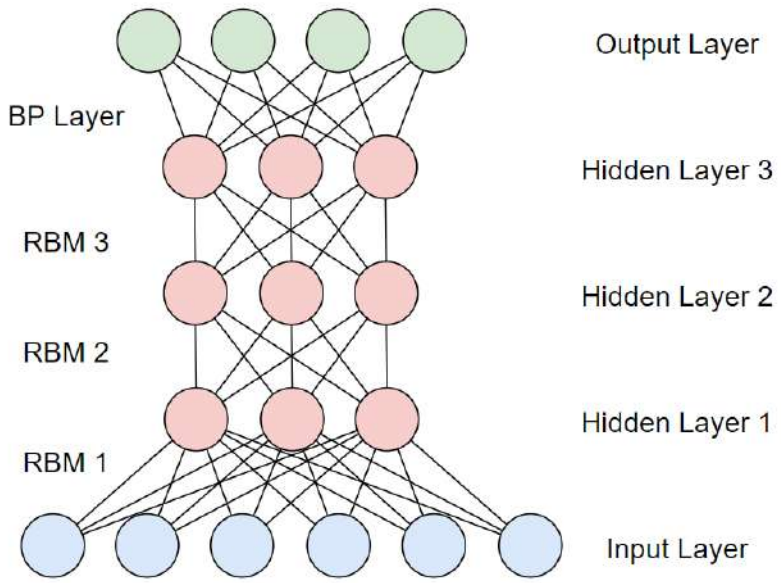

Figure 3.13 : Example of a Deep Belief Network architecture. © Rani Baghezza, 2022.

in order to perform training and classification. Since DBN is a generative model, we expect it to learn a good representation of each class (in this case, we expect the model to learn the probability distribution that encapsulates the way a certain individual walks), and to be robust to some degree of variation in the data. Indeed, the authors have shown that their method generally outperformed other methods on the same dataset, except in the case where no variation in conditions (view angle and clothing) occurred. The model did not perform well on big variations between the training data and the testing data, such as training the model on subjects wearing a backpack, and testing it under normal walking conditions.

Generative Adversarial Networks (GANs) have been introduced by Goodfellow et al. [135] as a two-player minimax game between a generative model (the generator) and a discriminative model (the discriminator). Deep neural networks are typically used. The idea behind GANs is for the generator G to generate synthetic data from a random input z, and for the discriminator D to be able to tell the difference between real and fake data. As the

generator gets better, the discriminator's performance should slowly reach a point where it is unable to tell the difference between real and synthetic data. GANs have experienced a tremendous improvement since their inception in 2014 [135], due to improvements in the models, as well as advances in deep learning in general as well hardware improvement [136]. Early examples of generated data can be found in Figure 3.14.

Figure 3.14 : Early synthetic data generated by GANs. The yellow rectangle highlights the closest image in the training set, ensuring that the model does not simply generate known data [135]. © 2014 Curran Associates.

GANs have been used in gait recognition to address the typical view angle, clothing and occlusion challenges that are met in gait datasets. Yu et al. [137] have presented their GaitGAN method, which relies on a GAN to generate invariant gait images at a side-view angle while preserving the human and identity information in the image. Their model is composed of one generator and two discriminators. First, the GEI is extracted from a sequence of images for each subject, clothing variation and view angle in the CASIA-B dataset. The overall architecture of the system is shown in Figure 3.15.

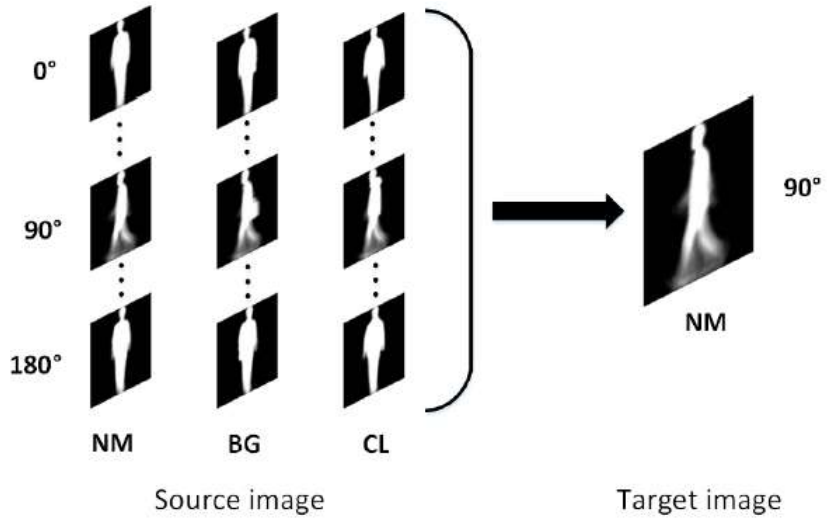

Figure 3.15 : Source and target GEI, where the source images are GEI computed on sequences captured at various angles, and under various clothing conditions, and the target is a 90 ° GEI of the subject under normal conditions [137]. © 2019 IEEE.

The authors have found that this method outperformed state-of-the-art methods in the field, especially in cases where there is a high variation between the training angle and the probing angle, as opposed to the DBN-based method previously mentioned [134], where a high variation led to worse performance. Both approaches used generative methods, however, the addition of the identification discriminator and the use of GEI rather than more specific body part-based features seems to lead to a better capacity for generalization for different view angles. GANs have also been used in other work to deal with the view angle variation issue [138] as well as to deal with subjects carrying objects [139].

Going back to discriminative models, Capsule Networks (CapsNet) [140] have recently been introduced as a way to address the issues created by the use of pooling layers in CNNs. A CNN is made up of convolutional layers that apply a certain transformation to the image using a filter which is learned through training. These convolutional layers are interleaved with pooling layers, which usually take the max, or the mean of the activation value of pixels in a small region in the image, usually a 2x2 region, which is replaced by a single value. This

operation reduces the dimension of the feature maps in subsequent layers, leading to a reduced computational cost, and they also provide the CNN with useful properties such as invariance to orientation and translation. However, they also lose the information about a shape's orientation and coordinates. Moreover, the relationship between a part of the image and the whole are slowly learned in deeper layers of CNNs, by extracting more abstract features from low-level features, but they are not directly learned at each layer. In CapsNet, the concept of *capsules* is introduced, where a capsule is a group of neurons, or a vector, whose parameters give information about the instantiation parameters of an entity or an object. The overall original architecture of the CapsNet can be found in Figure 3.16.

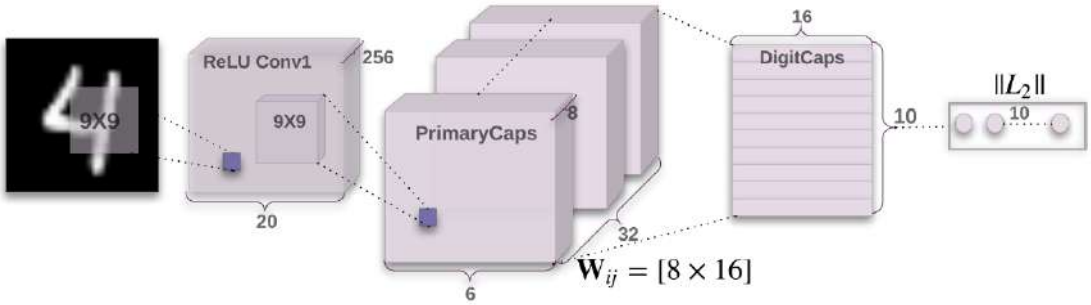

Figure 3.16 : Capsule Network architecture presented by Sabour et al. [140]. © 2017 Curran Associates.

CapsNets have been used for gait recognition by Xu et al. [141] on the CASIA-B and OU-ISIR treadmill datasets. The architecture of the two models presented by the authors can be found in Figure 3.17. The basic idea is to first extract features using classical convolutional layers on the GEI for each sequence, before using CapsNet in order to perform inverse rendering on the images to construct better representations for each entity. In this example, the similarity between two images is measured using a margin loss function.

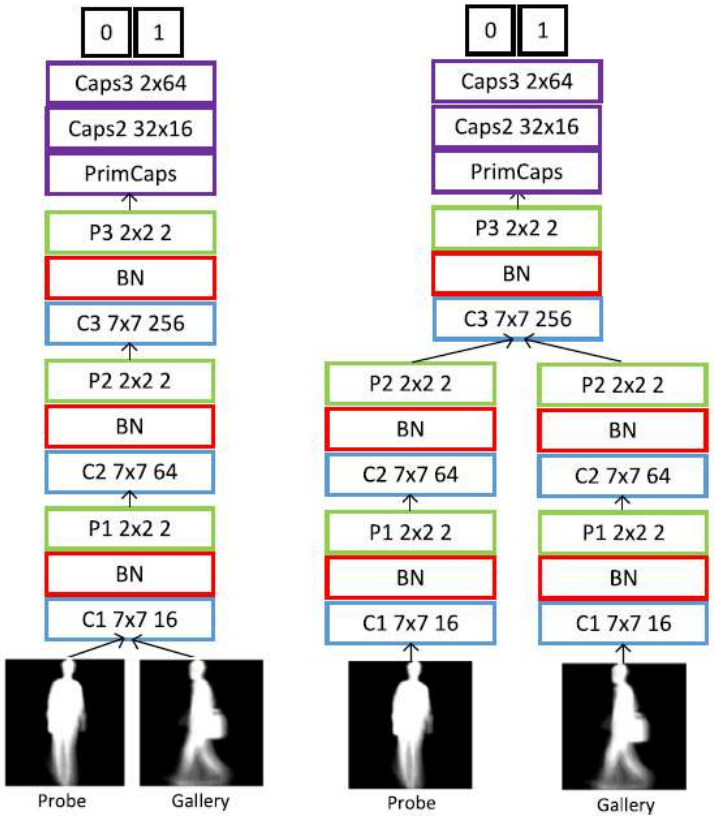

Figure 3.17 : Two architectures of CapsNet for gait recognition. The first architecture (left), aims to measure the similarity between low-level features of two images by coupling them at the first convolution layer (LBC). The second one (right) first performs a couple of convolution steps on each images and measures the similarities using medium level features (MMC) [141]. © 2019 Elsevier.

The authors have found that CapsNets outperformed other methods in the literature, and that the LBC (early coupling) yielded better results than the MMC (late coupling). The use of 4 capsule layers seemed to give the best results overall, even though the results for subjects walking under normal conditions did not seem to improve past 3 capsules, whereas 4 capsules improved the results when clothing variations were involved. It was observed that wearing a cap or a rain coat had the biggest negative impact on gait recognition results in these settings.

CapsNet have also been used as a part of the architecture for Sepas et al.'s gait recognition system based on partial representations [128].

To conclude this subsection, we describe a few hybrid methods that have been used in recent years, where two or more deep learning architectures have been combined in order to improve gait recognition performance. The first example is the use of CNNs combined with RNNs with two main scenarios: in the first, most straightforward approach, a CNN is used to extract features for each sequence in a sequence of images for gait recognition [142]. The RNN is then used to find temporal features in the sequence of images. The architecture of this approach is described in Figure 3.18. In this case, a Long Short-Term Memory (LSTM) network is used [82].

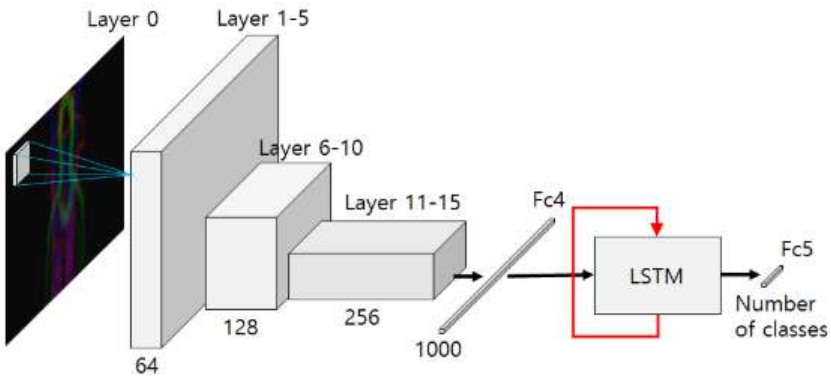

Figure 3.18 : CNN-LSTM architecture used by Batchuluun et al. to perform gait recognition on a sequence of images [142]. © 2018 IEEE.

The second approach has been to split an image into 4 horizontal partitions that are individually fed to a CNN before a LSTM is used to learn spatio-temporal features [143]. In this case, the LSTM is used as a temporal attention mechanism, and allows the model to focus on more discriminative parts of the image. This approach is combined with a custom loss

function called the Angle Center Loss (ACL) function, which has shown better performance than GEI. The architecture used by Zhang et al. [143] can be found in Figure 3.19.

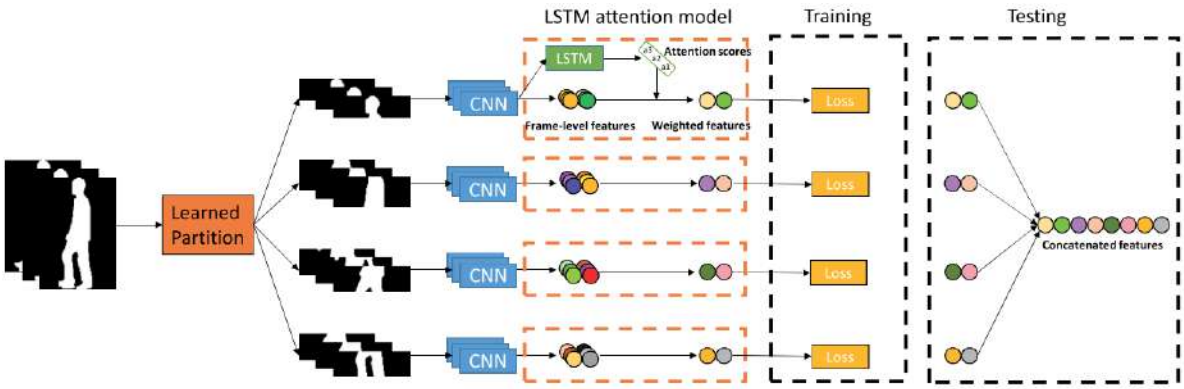

Figure 3.19 : CNN-LSTM architecture used on horizontal portions of images [143]. © 2020 IEEE.

DAE have been used on their own, as well as in combination with GANs. Indeed, the generator part of a GAN aims to generate a fake image that looks like it could belong to the real dataset. In the case of gait recognition, and image-oriented tasks, the input is simply a random, noisy image of the same dimension as the image we want to generate as the output. This naturally leads to the construction of a neural network that has the same input and output size, and that performs transformations on the image to go from noise to a realistic image, leading to the use of Deep Auto-Encoders for this task. DAEs have also been combined with RNNs, where an auto-encoder called GaitNet is used to disentangle gait features into three latent representations: appearance, canonical and pose features, and a LSTM is used on the latter [144]. The architecture is shown in Figure 3.20. One of the biggest challenges in gait recognition is to extract gait features that are discriminative among subjects, but that are invariant to various confounding factors, such as clothing or view angle variation. Depending on the method and the dataset, two different subjects wearing similar clothes could be wrongly

identified as being the same person, because of the model relying on the appearance features more than the actual gait. The LSTM learns from the dynamic (pose) features, while the DAE allows to learn static (canonical) features. The combination of both types of features has demonstrated superior performance compared to the state-of-the-art on datasets such as CASIA-B and USF. The authors have also built and used a custom gait dataset (FVG) made of more challenging images captured from front-pose angles, which are known to contain less dynamic gait information compared to side-view images.

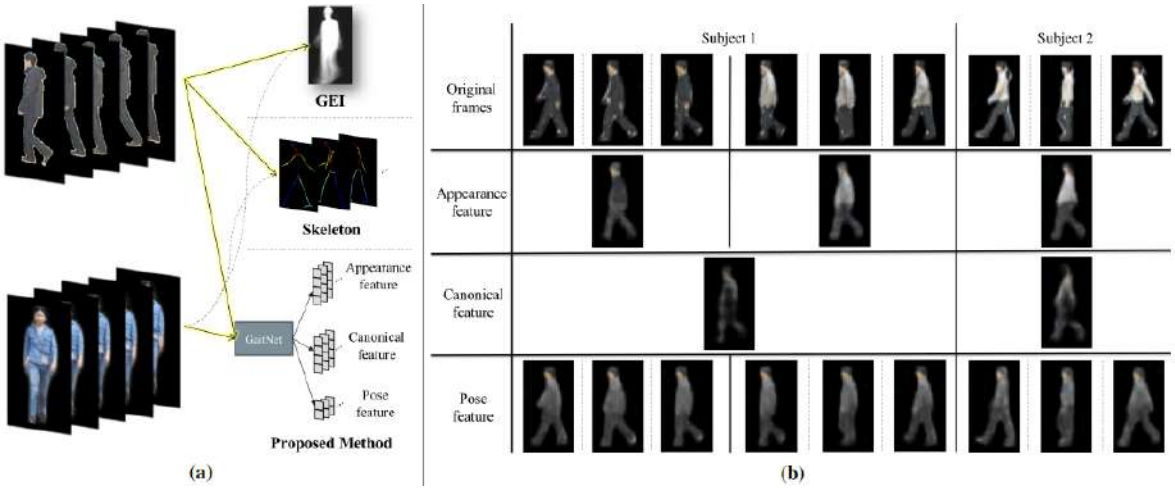

Figure 3.20 : Architecture presented by Zhang et al.The authors use GaitNet instead of GEI or skeleton-based features (a). Appearance, canonical and pose features are extracted using a DAE (b) [144]. © 2020 IEEE.

Sepas et al. have combined RNNs and CapsNet to perform gait recognition on partial features. [128]. The architecture used by the authors is illustrated in Figure 3.21. Multi-scale partial features are extracted from the gait sequence using GaitSet [129] which are then fed to Bidirectional Gated Recurrent Units (BGRU) [145] in order to learn correlations between different partial features. CapsNets are then used to learn part-whole relationships, allowing the model to generalize better to different view angles and clothing conditions. The authors

have shown that their method consistently outperformed previous methods, especially in conditions with variations in clothing or when subjects are wearing bags on the CASIA-B and OU-MVLP datasets.

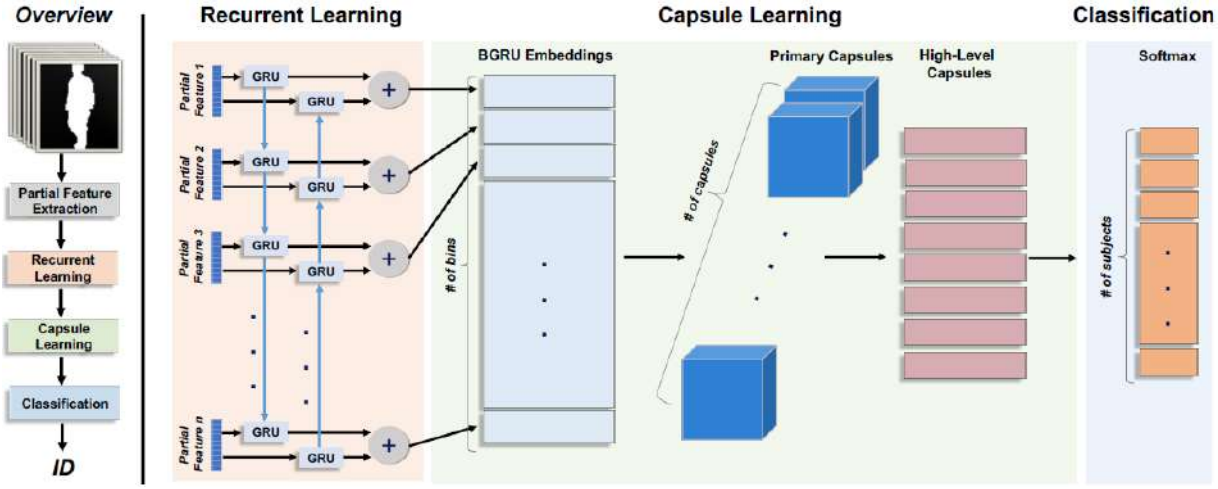

Figure 3.21 : Architecture combining Recurrent Neural Networks and CapsNet [128]. © 2020 IEEE.

3.2.5 CONCLUSIONS ON DEEP LEARNING-BASED GAIT RECOGNITION

As described in the previous section, many different types of deep learning algorithms have been used to perform gait recognition, from simple CNN-based architecture, to improvements using CapsNets, to generative models (DBN), and the creation of synthetic data (GAN), to sequential models (RNN) aiming to capture the temporal aspect of gait. Even though most approaches rely on the use of global silhouettes to extract features, some promising approaches using partial features have been used to deal with issues such as silhouette occlusion. The main challenge in deep learning-based gait recognition is to build a model that can perform

well on subjects walking at different view angles and wearing different accessories, clothing, or carrying a bag or a backpack.

The number of different algorithms and possible avenues to explore gait recognition using deep learning is very wide, it is therefore important to prioritize what is important for this thesis. The long term vision of the system presented in this thesis is to perform real-time profile recognition on embedded nodes in the city using thermal images. Given these constraints, the algorithms used should remain simple enough to run on limited resources nodes. Additionally, most deep learning approaches presented in the previous section perform some kind of pre-processing on the data. In a real-time environment, especially dealing with noisy data with a high variability, using automatic pre-processing methods such as algorithmic background subtraction could prove to be challenging. Indeed, variation in silhouette size, temperature (affecting the image's color), as well as background activity (cars, motorcycle, trucks), would make it impossible to implement a one size fits all pre-processing method that could extract the silhouette, and normalize it to work with uniformly sized binary bounded silhouettes, which are used in most of these approaches.

3.3 CHAPTER CONCLUSION

In this chapter, we have given an overview of the landscape of gait recognition in the past couple decades, from the first approaches using a very manual pipeline consisting in the manual extraction of a set of features, and distance/conventional machine learning-based classification, to more recent approaches using complex multi-part deep learning models. This chapter was important to set the basis of gait recognition, present some of the most used datasets, and provide the reader with a good understanding of the context of the research, as well as the main challenges in the field.

CHAPTER IV

RELATED WORK IN PROFILE RECOGNITION

In the previous chapter, we have reviewed standard gait recognition methods, which aim to recognize a single person based on their gait in a dataset containing sequences of images capturing many different people. In this thesis, however, we want to explore whether it is possible to recognize a certain category of people based on their gait using thermal images in the city. Indeed, if ambient assisted living is to be extended to smart cities in the future, and if pervasive intelligence is deployed in cities to improve the quality of life of the citizens, one of the first necessary steps would be to recognize different categories of people in order to cater to their needs. From a healthcare perspective, a useful application would be to recognize the elderly as well as persons with reduced mobility, which could lead to valuable insights for accessibility and inclusivity in the city. Children could be added to potentially vulnerable profiles in the city, given their frailty and their increased risks of serious injuries in traffic accidents [43].

The terms *profile* and *vulnerable profile* can be vague, subjective, and context-dependent. The profile of an individual is multi-faceted, making it important to define the aspect of the profile we are interested in. As has been established in the previous chapter and in the introduction, when considering gait recognition, we are not relying on wearable sensors, and we are not considering model-based methods that rely on the reconstruction of the individual's skeleton. Because of the observational nature of the experiments of this thesis described in the next chapter, it is not possible for us to have detailed information about pedestrians, such as their exact age, possible health conditions and so on. Therefore, we have to rely on observable characteristics of the individual's profiles, such as estimated age category, observed gender, and observed mobility.

Ultimately, our goal is to see whether it is possible to perform age, gender, and mobility estimation using noisy, low-resolution thermal images in the city and an appearance-based approach. Recognizing a broader category of people based on their gait is somewhat of a new challenge in the gait research community, as the focus of the community has been directed towards individual gait recognition in the past few decades. In recent years however, age and gender have emerged as the main gait-based profile recognition areas of interest, which we will review in this chapter. Gait-based mobility recognition has mainly been studied in detailed analysis of the gait cycle for people with specific ailments, but it has not been studied in systematic ways on larger databases. In this chapter, we review related work in the literature for gait-based age estimation, gait-based gender estimation, before defining thermal images, and exploring relevant work done on thermal images in the literature.

4.1 AGE ESTIMATION

Human age estimation is an important application in many fields, such as security, healthcare, and identification. In terms of healthcare, it has been reported that 35% of adults over the age of 70 are affected by an abnormal gait [146], which can lead to falls and increased mortality [147]. Age estimation methods relying on facial images [148] or voice recordings [149] have proven successful in the past. However, these approaches are met with limitations: in order to estimate the age of an unknown individual, it is necessary to collect a good quality image of their face or to record their voice, which is not always possible, especially in an urban environment. Gait has therefore been seen as a more practical avenue to address the age estimation challenge, as low resolution images can be used as well as video footage captured from far away. The field of gait-based age estimation is still nascent, as illustrated on Figure 4.1. In fact, the number of papers tackling gait-based age estimation in 2019 and 2020 combined is superior to the sum of all the papers published before these years. In this section,

we focus on vision-based age estimation using gait, and exclude approaches using wearable sensors.

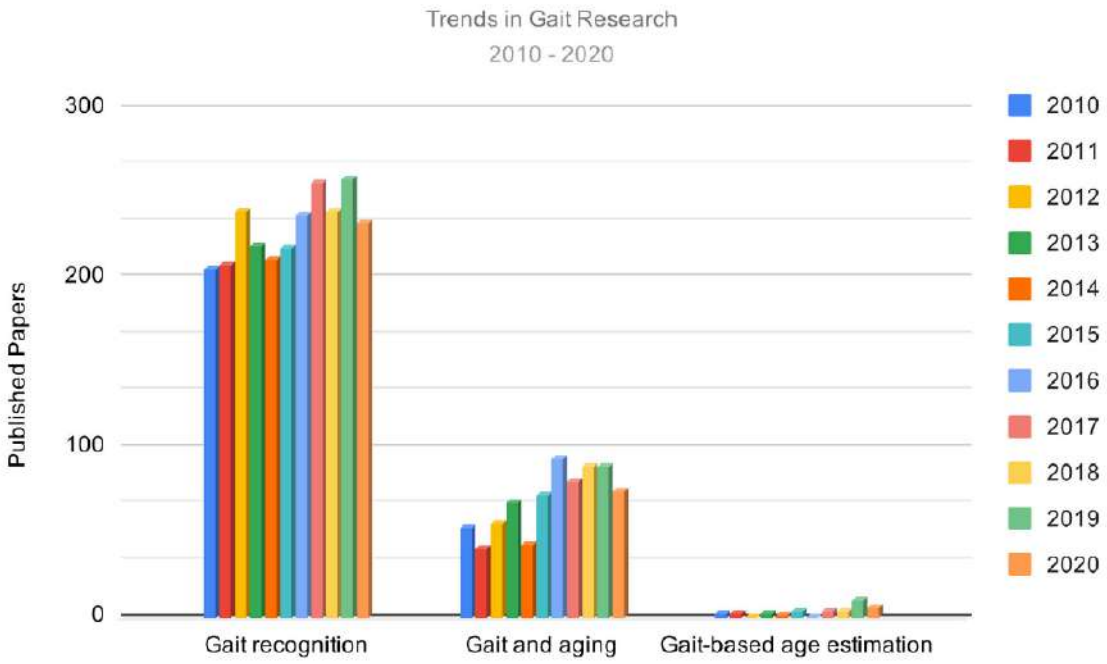

Figure 4.1 : Diagram illustrating the focus of gait recognition research in the past years in terms of published papers [150]. © 2021 IEEE.

Gait-based age estimation can be challenging because of the covariates that have been highlighted in the previous chapter, such as change in view angle and variation in clothing. As always, the challenge is to reliably estimate the age of an individual based on their gait, excluding the effect of external, non-gait influences: the model should learn that someone is an elder because of the way they walk, not based on how they are dressed. The authors in [150] have summarized the main sources affecting gait variability in Figure 4.2.

There is no absolute way of dealing with internal factors since something as simple as mood could affect the way a pedestrian walks. The impact of external factors can be mitigated

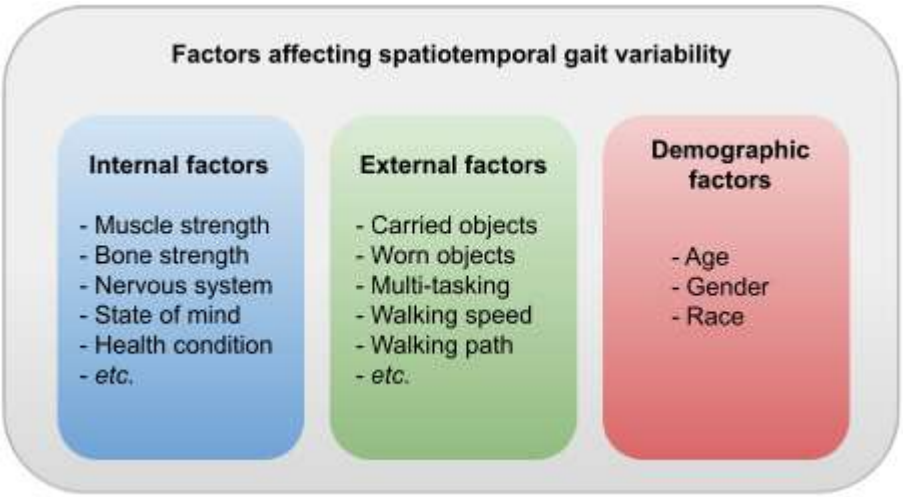

Figure 4.2 : Summary of the factors affecting gait variability [150]. © 2021 IEEE.

with some of the methods presented in the previous chapter, relying on combinations of deep learning models, and learning to transform gait sequences to different view points or clothing variations. The demographic factors, such as age and gender, represent the target we want to uncover based on the observed gait. However, even these factors are not independent, as Callisaya et al. [151] have shown that gender actually modifies the relationship between age and gait. For men, the relationship is more straightforward and linear than for women, as the various accessories, carried objects, and additional factors such as pregnancy can heavily affect the estimations of any model [52].

4.1.1 DATASETS

In order to perform age estimation, the datasets used need to contain age information about the subjects, which has not always been the case when building datasets for individual gait recognition. The number of public gait datasets including age has increased in the past decade, however, the age range varies a lot from experiment to experiment, and some datasets appear to be more gender imbalanced than others, which can impact age estimation

performance. A summary of the main publicly available, image-based datasets used for age estimation is given in Table 4.1.

Table 4.1 : Summary of the main publicly available image-based gait datasets with age metadata used for age estimation. The list of dataset was adapted from [150], removing inertial sensor-based datasets. For each dataset, the number of male (M), female (F) and total number of subjects appears in the associated columns.

Dataset	Year	Modalities	M	F	Total	Age range
USF [114]	2005	Video/image	85	37	122	19-59
OU-ISIR [152]	2011	Video/image	88	80	168	4-75
OULP [50]	2012	Video/Image	2135	1872	4007	1-94
TUM-GAID [90]	2014	Kinect (RGB-D)	186	119	305	18-55
OULP-Age [153]	2017	Video/image	31093	32753	63846	2-90
OU-MVLP [154]	2018	Video/image	5144	5193	10307	2-87
VersatileGait [155]	2021	Synthetic gait	-	-	11000	-

As illustrated above, early datasets such as USF [114] and OU-ISIR [152] contain very few subjects, 122 and 168 respectively. OU-ISIR appears to be more gender balanced, and includes subjects of different age and gender walking on a treadmill, captured at 25 different view angles. However, the clothing remains the same throughout the experiment, leading to less covariate factors than USF, which includes changes in walking surface (concrete, glass), two types of shoes, and a variation in carrying conditions, as the subject are either empty handed, or carrying a briefcase. The age group repartition for these two datasets is illustrated in Figure 4.3.

The OULP [50] dataset has been one of the biggest gait datasets in the literature for a few years with a total of 4007 gender-balanced subjects (53.3% male, 46.7% female). It is also the dataset with the biggest age range, with subjects ranging from 1 to 94 years old, as illustrated in Figure 4.4.

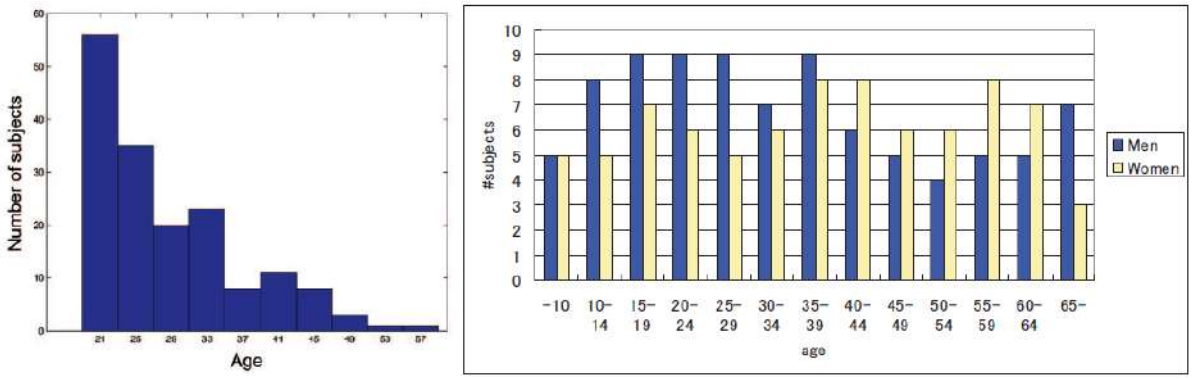

Figure 4.3 : Overview of the age group repartition of the subjects for the 2 early gait-based age estimation datasets found in the literature: USF [114] (left) and OU-ISIR [152] (right). © 2005 IEEE. © 2011 Springer Nature.

The large majority of the subjects fall into the 5-49 age range, as illustrated in the diagram below (Figure 4.5). Since the dataset was collected over the course of several different events, no controlled gait covariate have been captured: each subject simply walks in, dressed the way they are, without carrying any bags. Each subject is captured at 4 different camera angles, 55°, 65°, 75° and 85°.

The TUM-GAID [90] dataset differs from the other datasets in this list as it has been built using a Kinect sensor, which captures a depth component for each pixel of the image, as well as the standard RGB data. The dataset is composed of 305 subjects, 61% of which are male, and 39% female. The subjects are recorded in different conditions: normal, carrying a backpack, and wearing coating shoes (protective mesh surrounding the shoes, similar to the ones used in clinics or hospitals for hygienic purposes). The age repartition of the subjects in the TUM-GAID dataset is narrower than in the OU-ISIR and OULP datasets, as illustrated in Figure 4.6. This dataset is smaller than the OULP dataset, and not as gender-balanced, but the use of a RGB-D camera and the inclusion of gait covariate, as well as audio data

Figure 4.4 : Example of subjects of different age in the OULP-C1V1 dataset [50]. © 2012 IEEE.

captured by the Kinect allows for interesting analysis, such as the exploration of multi-modal gait recognition [156].

The OULP-Age [153] is by far the biggest gait-based age estimation dataset publicly available, with a total of 63,846 subjects ranging from 2 to 90 years old, 31,093 of which are male, and 32,753 female, making it gender-balanced. The age repartition for this dataset is shown in Figure 4.7. As most previous datasets, we can see that it is mainly composed of adults, however, compared to most other gait datasets, the number of children and teens is considerable in OULP-Age. There are relatively fewer elders compared to adults and children, however, because of the overall size of the dataset, the amount of elders is still significant. Because of the sheer size of this dataset, no gait covariate or different view angles are captured, as the subjects simply walk in and out of the frame, with a single camera capturing their gait at a 90 ° angle.

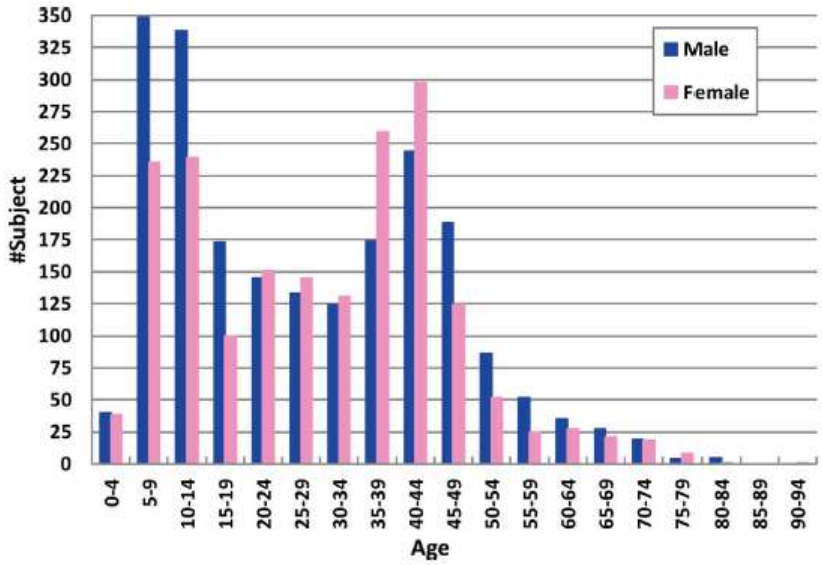

Figure 4.5 : Age repartition of the subjects in the OULP-C1V1 dataset [50]. © 2012 IEEE.

The OU-MVLP [154] dataset is gender balanced as well, and contains a total of 10,307 subjects recorded at many different camera angles from 0° to 90° in 15° increments. Since the subjects walk back and forth, the authors were able to collect data corresponding to angles from 180° to 270° in 15° increments as well, leading to a good coverage of all view angles. The age repartition for OU-MVLP is shown in Figure 4.8.

The last reviewed dataset, VersatileGait [155], differs from the previous datasets as it is a synthetic gait dataset created using a game engine. The motivation for a synthetic gait dataset comes from the shortcomings of real-life datasets, such as the small number of participants in some cases, or the lack of covariates and different view angles in bigger datasets. Collecting a gait dataset is also a long and expensive process: it took 11 months to collect the OU-MVLP dataset. The collected datasets lack variety and they can contain damaged silhouettes due to limited data pre-processing.

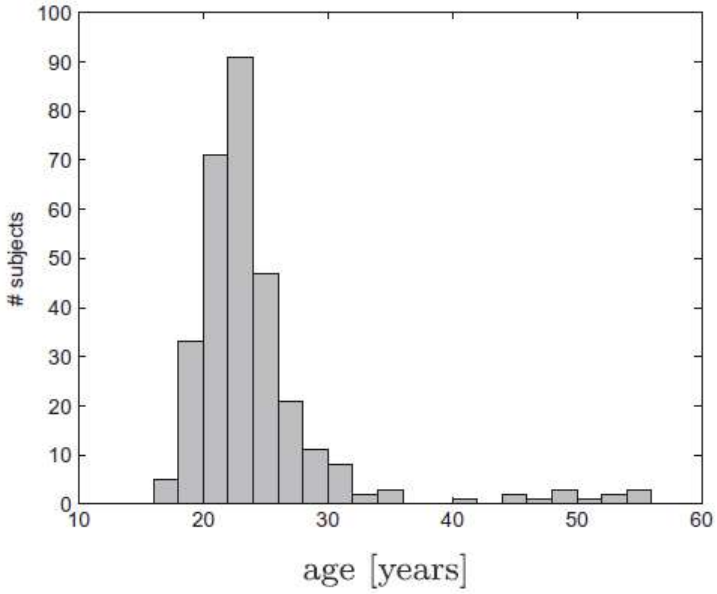

Figure 4.6 : Age repartition of the subjects in the TUM-GAID dataset [90]. © 2014 Elsevier.

The use of a game engine to simulate and record a walking behaviors allows a very fine-grained annotation, as all of the parameters for each animation are hand-picked, and allows the creation of complicated and customized scenarios. No privacy issues arise as the data is not collected on real people. VersatileGait consists of a million silhouettes of 11,000 subjects with different attributes, walking in various scenarios. The labels are much more fine-grained than for real gait datasets, and there is more freedom when it comes to camera positioning since the scene is virtual, as illustrated in Figure 4.9. Other useful applications coming from this dataset are multi-person gait recognition, where several subjects are considered at the same time, as well as using VersatileGait to pre-train deep learning models, leading to a 1.1% increase in accuracy on the CASIA-B dataset. The combination of real and synthetic data has shown promising, as using 50% of synthetic data and 50% real data has lead to the same performance as using 100% real data, which is useful in a field where data collection and annotation is expensive and time consuming.

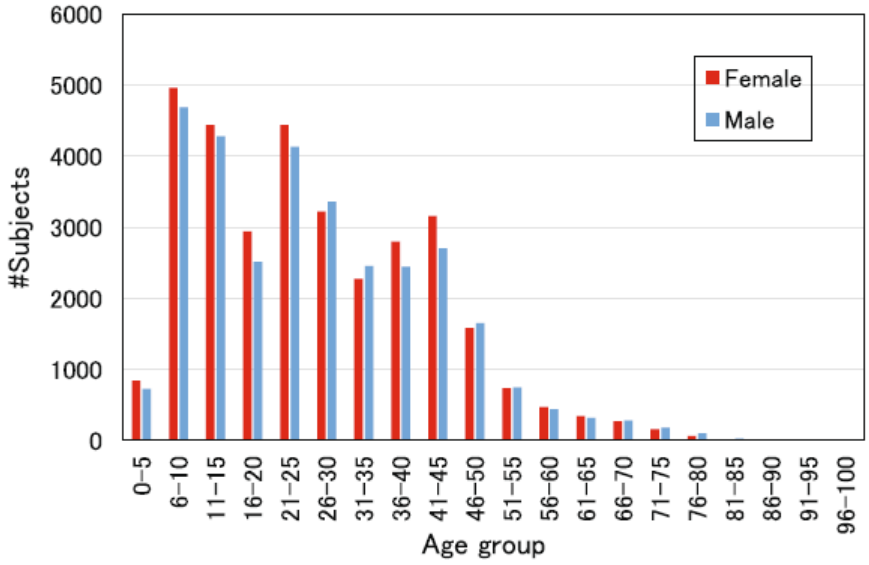

Figure 4.7 : Age repartition of the subjects in the OULP-Age dataset [153]. Creative Commons Image.

Overall, datasets in the literature either include a high number of different subjects and a low number of gait covariate, or the opposite. We can see in the age repartition figures throughout this section that most dataset contain much more adults and younger individuals than elders, which is an additional challenge for age classification. These datasets are collected in a controlled environment as well, either indoors using a green screen and ideal lighting, or outdoors with reference points and a fixed background. In terms of gait-based age estimation using public or custom datasets, there are two main avenues to explore: age classification and age regression, which are the focus of the following subsections.

4.1.2 AGE CLASSIFICATION

Age classification is the simplest way of tackling the age estimation problem. Instead of trying to estimate the specific age in years of an individual, age classification aims to estimate if an individual belongs to a broad age category. The age range can be broken down into its

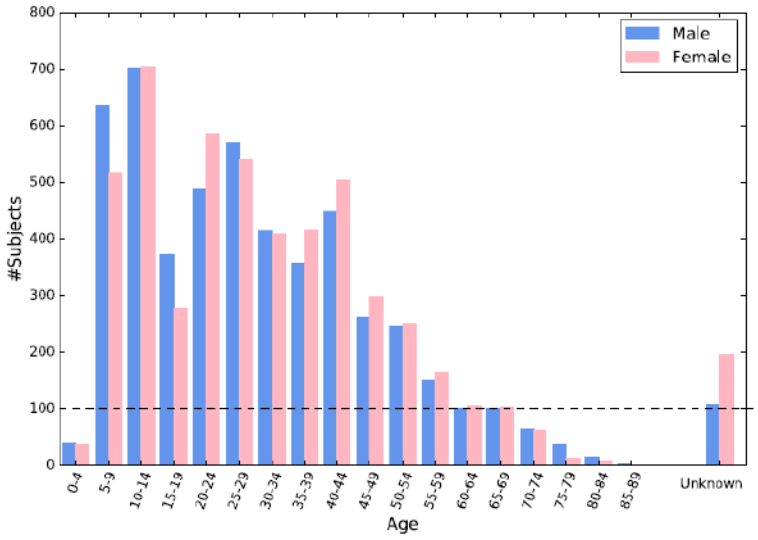

Figure 4.8 : Age repartition of the subjects in the OU-MVLP dataset [154]. Creative Commons Image.

three simplest components: child (generally from 0 to 15 years old), adult (15 to 65) and elder (65 and older). Generally speaking, gait maturity is reached between the age of 2 and 7 in children [150]. Past the age of 15, it becomes increasingly difficult to tell the difference between a teenager and an adult from gait and static parameters such as height and width of the silhouette. It has been observed that the gait of an individual typically starts declining after the age of 60 [157], and the age of 65 is often used in the literature to refer to senior citizens and the elderly [27]. In a similar way to gait recognition, there are two main approaches for gait-based age estimation: model-based and model-free. The first approach aims to build a model of the skeleton of the subject, using static features, such as joints angle and segment lengths. In the context of this thesis, we will focus on model-free approaches, which rely on the use of video and images to perform age estimation.

The first effort to tackle the challenge of gait-based age classification is attributed to Davis [158] back in 2001. For this work, the dataset consists of 6 children (age 3 to 5) and 9

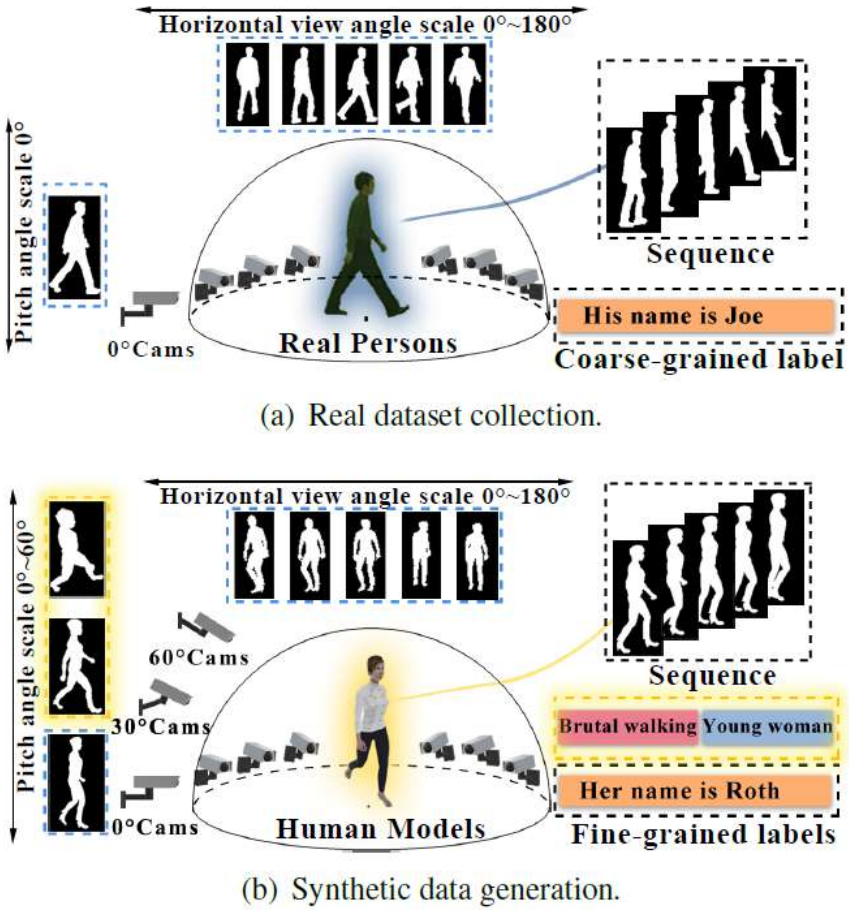

Figure 4.9 : Comparison of real and synthetic gait dataset generation [155]. Creative Commons Image.

adults (age 30 to 52), where the adults walk on a treadmill, and the children walk across the room while being recorded by a camera. The used human locomotion features are calculated on each walking cycle, such as cycle time, stride length (distance between footfalls of the same foot), and stature (height). The stride length can be normalized by the person's height, giving a relative stride L', which better captures the specific gait signature of the person or category of person under consideration. The stride frequency f is used as well. The experimental setup is shown in Figure 4.10.

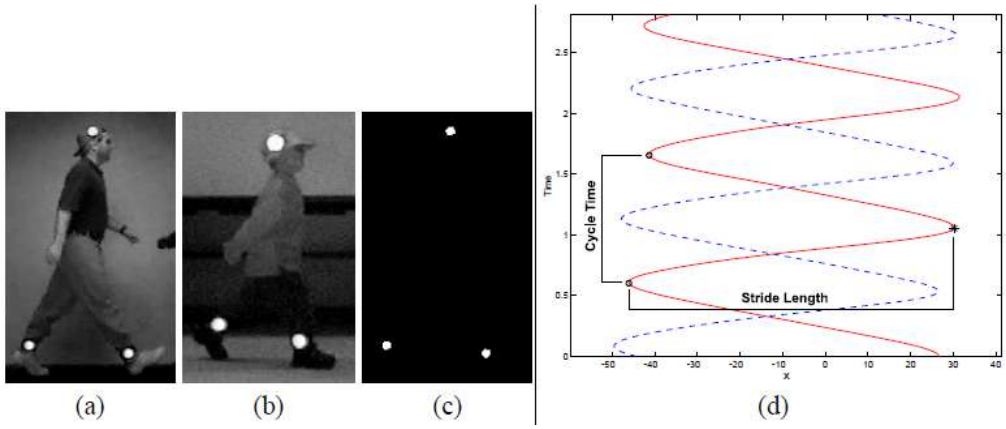

Figure 4.10 : Recording of an adult (a), child (b) and the reflective ankle and head markers (c) on the left. The stride length and gait cycle time can be read on the graph on the right (d) [158]. © 2001 Springer Nature.

In this case, the computed stride frequencies for adults and children were very distinct, and separable using a two-class linear perceptron discriminator of the form:

$$d(x) = \sum_{i=1}^{n} \omega_i x_i + \omega_{n+1} \tag{4.1}$$

The authors have estimated the accuracy of this method to be around 93 to 95% for binary age recognition. The separation between the classes appears clearly on Figure 4.11 below.

The method used by Davis falls into the category of marker-based gait recognition, where markers are placed on the subject, images are captured, and the evolution of the position of these markers in time is used to perform gait recognition. This approach is different from the work of this thesis, however, this paper is important to mention as the first gait-based age

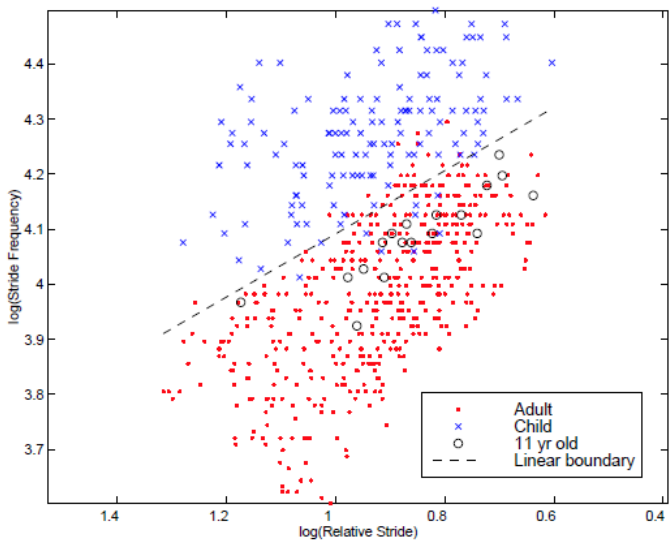

Figure 4.11 : The two classes are linearly separable using stride frequency and relative stride only [158]. © 2001 Springer Nature.

classification effort. The used features are simple, the subjects belong to two very distinct classes, leading to an easy separation between classes.

The first work performing age classification based on human gait without the use of any markers or sensors has been presented by Zhang et al. [159] in 2010. The authors have built a dataset with two categories of people: young (25-30) and elderly (60-65), made up of 14 subjects (7 for each category, 4 male and 3 female). From this custom dataset, silhouettes are extracted, the background is removed, and the contour of the silhouette is extracted before being reduced to a lower dimensional space. A Hidden Markov Model (HMM) is then used to perform binary age classification, achieving an accuracy of 83.33%. Examples of the extracted silhouettes and contour can be seen on Figure 4.12.

Makihara et al. [152] have used a bigger dataset (OU-ISIR) to perform multi-class classification, namely: children (C), adult males (AM), adult females (AF) and elderly (E). In

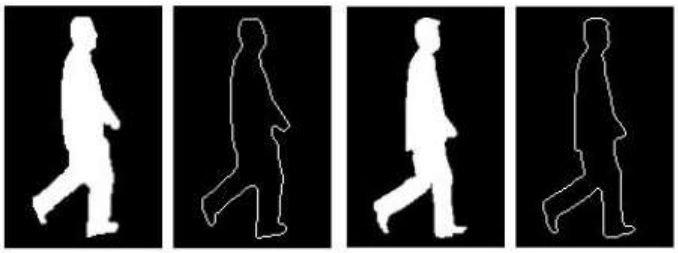

Figure 4.12 : Custom dataset sample for binary age classification: the silhouette of an elder and its associated contour can be observed on the left, and the equivalent for an adult on the right [159]. © 2010 IEEE.

this paper and some of the following papers in the literature, the correlation between age and gender is a reoccurring theme. Indeed, it would seem like knowing the gender of the individual could lead to increase in performance in age classification, based on the fact that each gender's gait evolves differently as the individuals age. In this work, subjects are recorded at different view angles, the silhouettes are extracted in order to construct the Gait Silhouette Volume (GSV) before extracting frequency domain features, performing dimensionality reduction and using k-NN to perform classification. Three binary classification tasks are evaluated: children-adult (C-A), adult male-adult female (AM-AF), adult-elder (A-E).

Figure 4.13 shows the difference in features between different classes at various view angles. Analyzing this figure leads to a few interesting observations: for example, the top left observation shows that children tend to have their head further forward than adults, and their body is leaning more forward overall, which coincides with observations stating that children tend to lean forward more in their gait [150]. Their arm swing is also less than the arm swing in adults. The second row shows that adult males tend to have a wider stature than adult females, while the third row illustrates how elders tend to bend forward more than adults. The classification results show that using overhead view is not as efficient as using side views, which capture posture and leg movement better. The authors have observed that

classification performance is affected by the age range under consideration: the wider the age range, the worse the performance of the classifier. the more diverse the age range is, the worse classification performance is. Indeed, using the whole dataset, the classification accuracy for AM-AF is 80% and 74% for C-A, but it increases to 91% and 94% respectively when restricting the adult age range to 25-34 years.

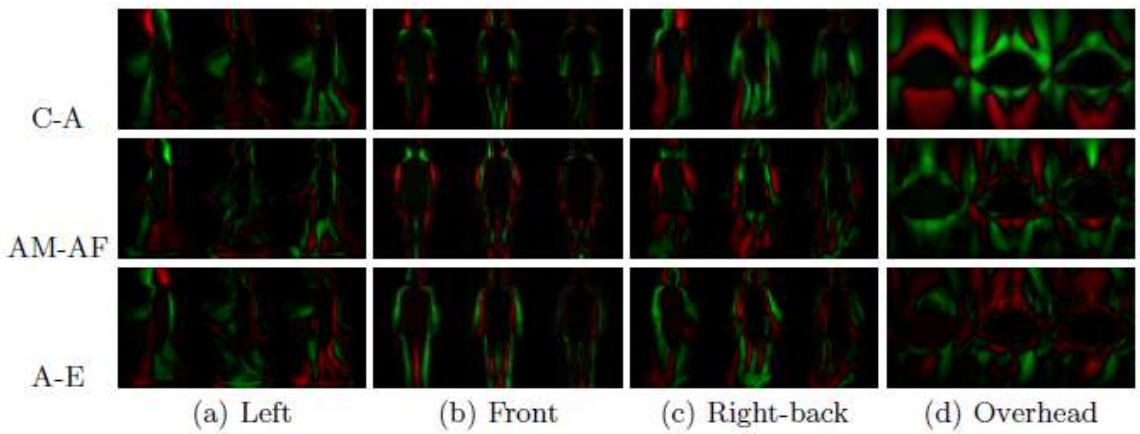

Figure 4.13 : Illustration of the feature differences between classes for each binary classification task under different view angles [152]. Red indicates a feature belonging to the leftmost class (C in C-A, AM in AM-AF and A in A-E), while green indicates a feature belonging to the rightmost class. © 2011 Springer Nature.

Hu et al. [160] have used Gabor filters [161] to decompose the body shape into local orientations and scales on silhouettes from the CASIA dataset. Gabor filters are essentially bio-inspired, mathematically generated filters applied to an image in order to detect different features of interest, such as specific textures. In the general, complex formulation, a gabor filter is expressed as follows:

$$g(x,y;\lambda,\theta,\psi,\sigma,\gamma) = exp\left(\frac{x'^2+\gamma^2 y'^2}{2\sigma^2}\right) exp\left(i\left(2\pi\frac{x'}{\lambda}+\psi\right)\right) \tag{4.2}$$

Where x and y are the dimensions of the filter, λ the wavelength, θ the angle, ψ the offset, σ the standard deviation, γ the aspect ratio. The filter is built by the repetition of a wave pattern constrained by these parameters. Many different scales, sizes and angles can be achieved by tweaking these variables, as illustrated on Figure 4.14. Gabor filters have been used for edge detection [162], texture segmentation [163], as well as face recognition [164] among various other applications.

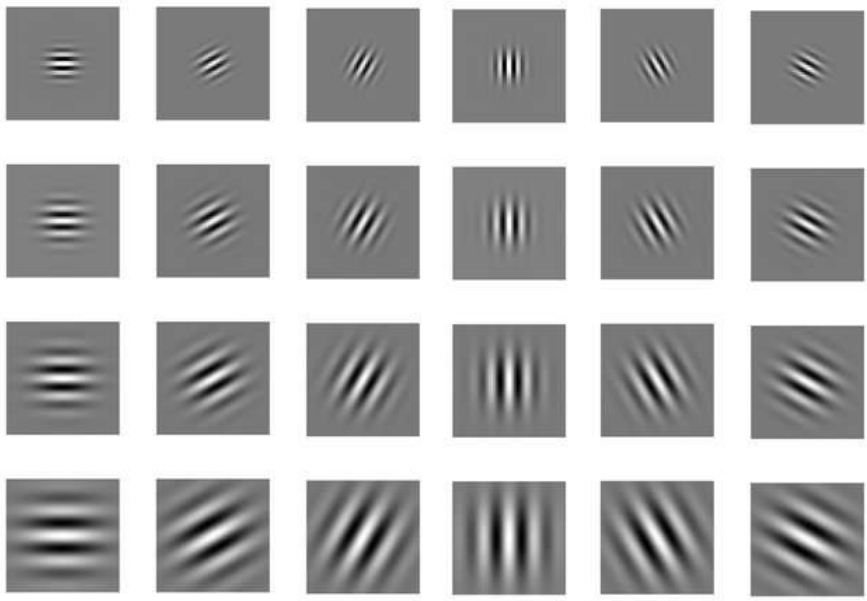

Figure 4.14 : Example of a set of Gabor filters of different scales and orientations used for car classification [165]. © 2015 IEEE.

In this paper, 3 different scales and 6 different orientations were used to extract features from the silhouettes using Gabor filters before training two HMMs, one for each age category (young and elderly), achieving a 85.71% accuracy in age classification.

Mansouri et al. [52] have performed binary age classification on the OULP dataset using a custom descriptor based on Silhouette Longitudinal Projection (SLP) and Silhouette

Transverse Projection (STP). It was highlighted from previous work that one of the factors that can affect the gait of the elderly, and thus, help classify gait signature as belonging to an elder rather than an adult, is the decrease of gait velocity caused by muscle degeneration [159]. The fear of falling also affects head pitch frequency as well as the general speed and cadence of the elderly [166]. Based on these observations, the authors have decided to isolate a gait cycle, and use SLP and STP to observe the variation in width and height of the silhouette throughout the gait cycle, respectively. The proposed approach is summed up in Figure 4.15.

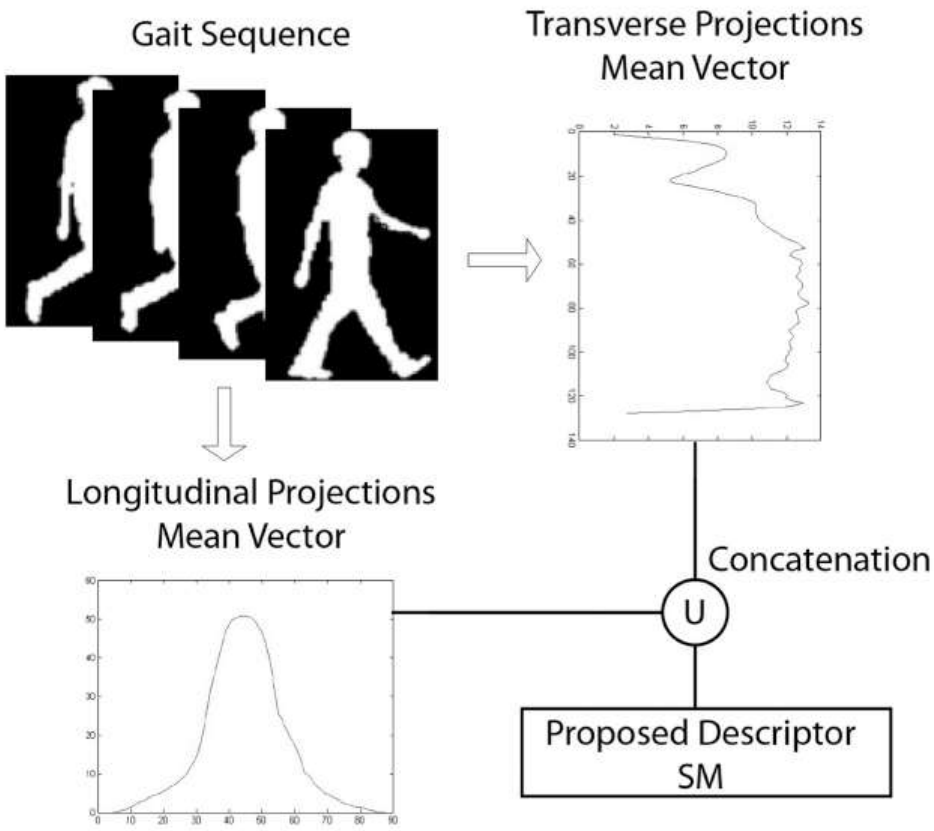

Figure 4.15 : Description of the SLP and STP based approach proposed by Mansouri et al. [52].
© 2018 IET.

The projections are computed in a simple way: considering binary silhouettes appearing white on a black background, the number of white pixels in a column (for SLP) or in a row (for STP) are counted and added up. The SLP and STP are then concatenated and a SVM is used to perform binary age classification. The difference in STP for young and elderly subjects can be observed in Figure 4.16.

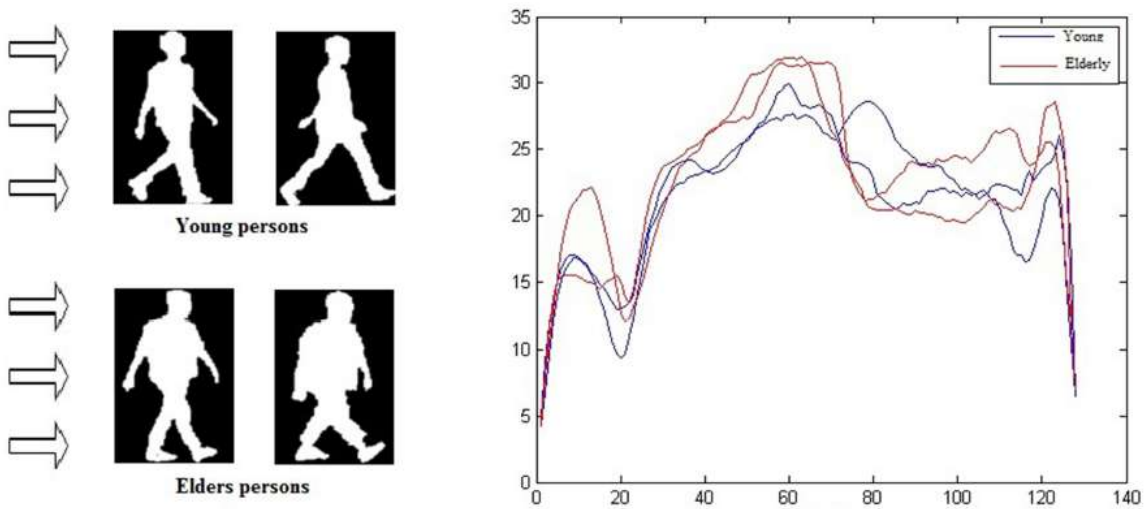

Figure 4.16 : Example of the STP for 4 subjects: two young and two elderly [52]. We can see the difference between the two groups: the STP for the elders seems to vary much more and reach higher peaks than for the younger subjects. © 2018 IET.

Using this method, the authors have achieved a 74.47% correct classification rate on the OULP dataset. Some interesting insights are also discussed in the paper, such as the impact of gender on the accuracy of the system: using only male subjects, the classifier reaches an accuracy of 83.33%, as opposed to 58.33% on female subjects only. The authors have attributed this difference to factors such as pregnancy that could significantly affect a silhouette's width and the subject's stride, leading the system to classify an adult pregnant woman as an elder. However, given the size of the dataset (4,007 subjects, 2,135 male, 1,872 female), it seems like pregnancy cannot be the only factor at play, as it would imply that

several hundred pregnant women were part of the dataset. Since OULP was collected over the course of several public, large scale, scientific events, it also seems unlikely that a lot of women at an advanced enough stage of pregnancy that would impact their gait significantly would be present and willing participants. However, the original OULP paper does specify that participants wore their own clothes and their own shoes during gait collection, which is probably closer to the source of the difference in results observed by Mansouri [52]. Factors such as hats, accessories, taller or smaller shoes, as well as the general difference in height between men and women most likely explain the difference in performance better.

4.1.3 AGE REGRESSION

With the appearance of larger, more detailed gait datasets containing age and gender metadata, the trend has naturally shifted towards age regression rather than age classification, with the first efforts being published as early as 2010 [167]. As illustrated previously in this chapter, most gait datasets are not balanced in terms of age repartition, and the emphasis is often times put on adults or younger subjects rather than elders. Combining several datasets can also be challenging because of the difference in image format, as well as view angle and gait covariate.

4.1.3.1 NON DEEP-LEARNING APPROACHES

Lu et al. [167] have worked on the USF dataset to perform age estimation. Being aware of the difference in silhouette appearance between male and female, as illustrated in Figure 4.17, the authors have encoded both age and gender labels in a binary format, shown in Figure 4.18. In the paper, Multi-Label Guided (MLG) subspace learning is used, where $X = [x_1, x_2, ..., x_N]$ represents the gait feature set (in this case, GEI), $L = [L_1, L_2, ..., L_N]$ are the

corresponding labels, and the goal is to learn a low-dimensional subspace W that maximizes the dependency between X and L. Since the dataset used does not always include the gender information for every instance, multi-label classification was used in the form of ML-KNN [168]. Multi-label classification is employed when an instance could belong to several classes rather than a single class. However, using the instances where gender is known, two separate subspaces can be found, one for each gender, where age estimation can be performed. Using ML-KNN then allows to perform classification in a robust way, even when the gender is not known for a specific test instance.

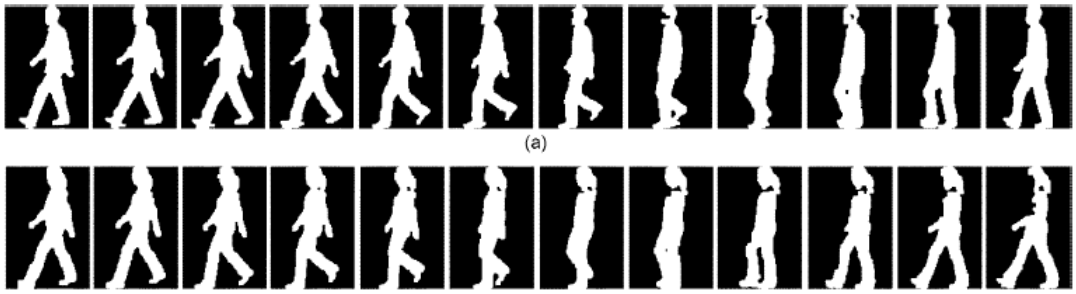

Figure 4.17 : Gait cycle for a 26 years old male (top) and a 26 years old female (bottom) [167]. © 2010 IEEE.

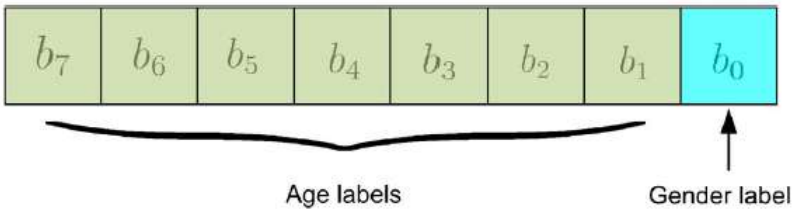

Figure 4.18 : Binary age and gender encoding [167]. © 2010 IEEE.

Using GEI only, a Mean Average Error (MAE) of 5.45 years was achieved on the USF database, and 3.86 years on a custom gait database. Using Gabor wavelets on the extracted GEI and using either the magnitude (GM), the phase (GP), or a fusion of both (GM + GP) has lead

to a MAE of 3.45, 3.34 and 3.02 years respectively on a custom gait database, demonstrating the efficiency of performing feature fusion.

Makihara et al. have performed age estimation on a bigger dataset containing 1728 subjects, as opposed to the 122 subjects of the USF dataset [95]. The authors have used Gaussian Process Regression (GPR) [169] because of its success in the field of face-based age estimation. Three silhouette-based features are used in this case: GEI, frequency-domain features (FREQ), and gait periods (GP). Some examples of these features are shown in Figure 4.19. The authors have achieved a MAE of 8.2 years, with 68% and 85% of the absolute error under 10 and 15 years respectively, with a slight, but insignificant difference in performance for male subjects (8.3 years MAE) and female subjects (8.2 years MAE). The frequency domain features seemed to give a very slight edge in performance compared to time-domain features (GEI).

Figure 4.19 : Frequency-domain feature examples, from left to right in each picture: averaged silhouettes, one-time frequency elements, two-time frequency elements of amplitude spectra over a gait period. [95]. © 2011 Springer Nature.

Although the MAE is much higher in this paper than in previously described work, this is largely explained by the dataset used. Indeed, a 5.45 years MAE was achieved on the USF dataset used by Lu et al. [167], however, that dataset is close to 15 times smaller, and the age range is much more restricted (19-59, as opposed to 2-94 in Makihara's work [95]). Besides, no feature fusion method was used in Makihara's work, which could also influence

the difference in results, even though the difference in age range seems like the most probable explanation.

Li et al. [14] have combined both classification and regression to perform gait-based age estimation. The OULP dataset is used in this work, which contains over 60,000 subjects. In order to perform both classification and regression, the authors have split the dataset into 5 years ranges increments from 0 to 20, 10 years from 20 to 60, and they have grouped up all elders over 60 in a single category. This split is based on the fact that gait evolves a lot more in younger years than in adult years, and 60 years old generally marks the moment where gait starts changing from an adult gait to an elder gait. GEIs for different age ranges are illustrated in Figure 4.20.

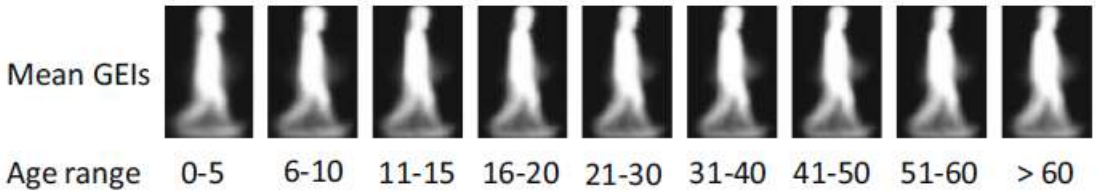

Figure 4.20 : Gait Energy Image for different age ranges in the OULP dataset [14]. Creative Commons Image.

The first step towards age regression is to use Direct Acyclic-Graph Support Vector Machines (DAGSVMs) to perform multi-class age classification [170]. Once the age has been narrowed down to a specific range, age group-dependent manifold learning is carried out to map the originally extracted GEI into a low-dimensional subspace more relevant to this specific age group. Non linear Support Vector Regression (SVR) is then used in each age group to perform age regression. The proposed approach is illustrated in Figure 4.21.

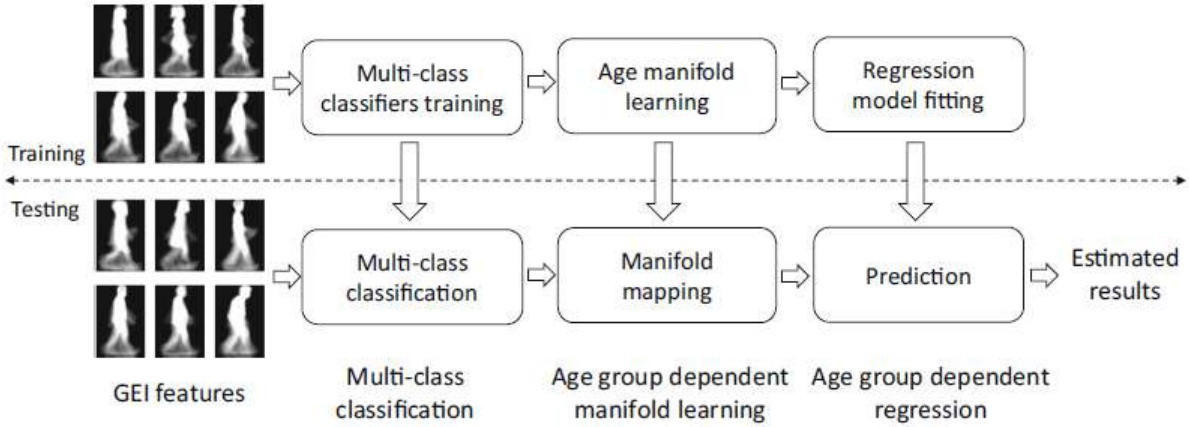

Figure 4.21 : Method proposed by Li et al. for human age estimation using classification and regression [14]. Creative Commons Image.

Since SVMs are designed for binary classification, DAGSVM creates $K(K-1)/2$ binary classifiers, where K is the total number of classes. Once these classifiers are trained, a tree is built to perform multi-class classification, as illustrated in Figure 4.22. Each node represents a trained classifier, where the root node is chosen to be the easiest classification task, in this case, the classification between the lowest age range (0-5) and the largest age range (>60).

After initially splitting the dataset population into 9 different age groups, the authors have computed the L2 distance of the average GEI for each pair of adjacent age groups, as shown in Figure 4.23. They have then grouped up similar age ranges based on the value of the computed L2 distance, and using a threshold of $8.0x10^4$, they have built 5 more relevant age groups, namely 0-5, 6-10, 11-15, 16-60 and >60. The DAGSVM is then built using these age groups. The highest Correct Classification Rate (CCR) of 79.84% was achieved for the age group 0-5, whereas the lowest CCR (63.48%) was reached for age group 11-15. As could be expected, a good portion of the misclassification happens in neighbouring age groups: 20.03% of the misclassified instances for the age group 0-5 are mistakenly classified as belonging to

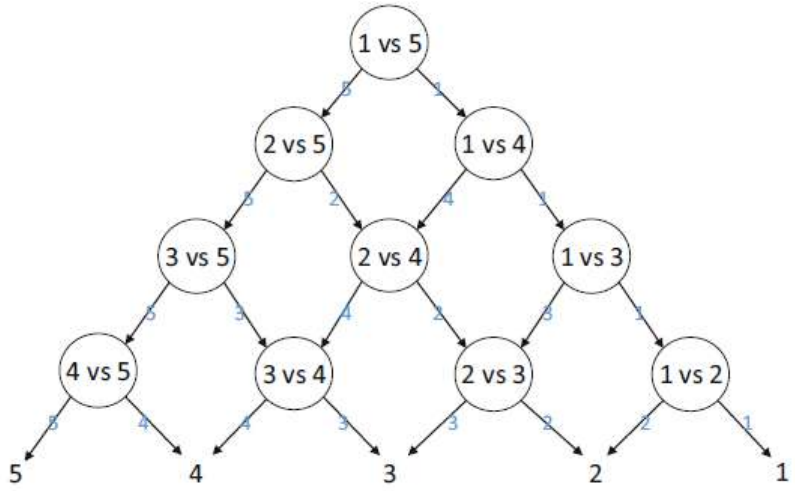

Figure 4.22 : Example of the DAGSVM tree used at classification time in the case where 5 classes are present. [14]. Creative Commons Image.

the 6-10 age group. In terms of regression, the authors have achieved a MAE of 6.78 years on the OULP dataset. On top of that, the proposed method has achieved an average computational time per sample of 2.69ms.

4.1.3.2 DEEP-LEARNING APPROACHES

As observed in the previous chapter when exploring general gait recognition approaches, the trend towards deep learning is noticeable in the field of gait-based estimation as well. At this point, it also becomes more difficult to isolate age-based paper, as the correlations between age and gender are often explored and as most used datasets provide metadata for both categories. In this subsection, we focus on the age portion when both modalities are explored.

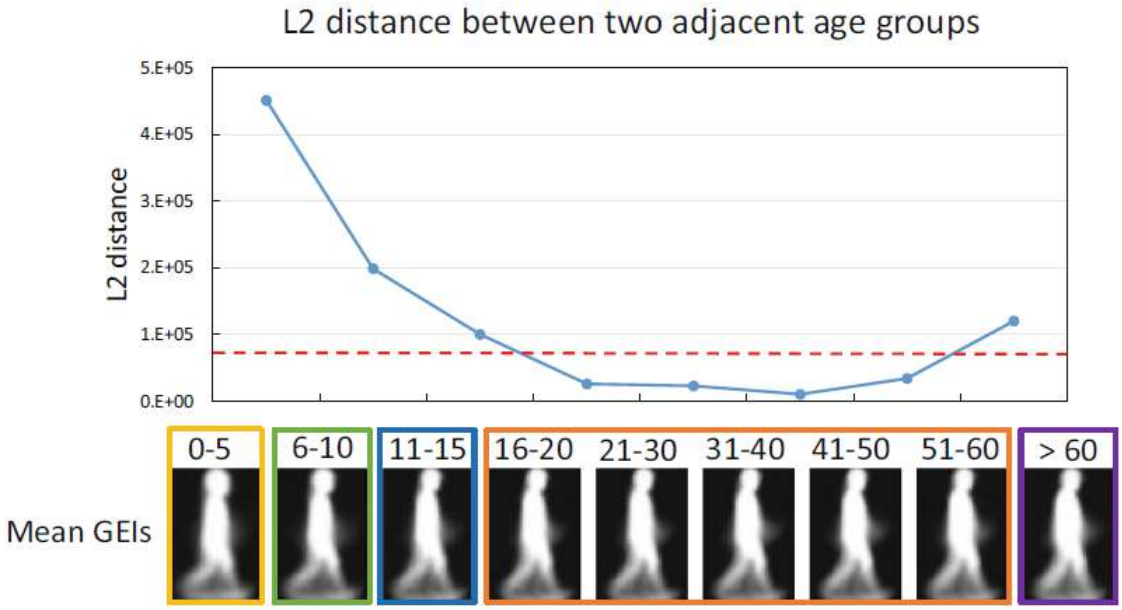

Figure 4.23 : Example of the DAGSVM tree used at classification time in the case where 5 classes are present [14]. Creative Commons Image.

Sakata et al. [171] have presented the first work using deep learning on a large gait dataset (OULP) for gait-based age estimation. The DenseNet [172] architecture used is illustrated in Figure 4.24. The use of skip connections between layers and all the way from the input of the models through several layers allows to combat the vanishing gradient issue that can occur when trying to train deep networks: multiplying the gradient by values close to 0 repeatedly can lead the change in the weight of the earlier layers to be 0, which prevents early layers from training and learning.

The authors have used Stochastic Gradient Descent (SGD) [173] over 100 epochs with a batch size of 64, as well as a decreasing learning rate based on the epoch (0.1 from epoch 0 to 50, 0.01 from 50 to 75, and 0.001 from 75 to 100) and a Nesterof momentum of 0.9 [174]. The number of dense blocks M and layers per block N is evaluated to find the best performing architecture, with $M \in \{2,3,4,5,6,7\}$ and $N \in \{4,5,6\}$. A significant performance drop was

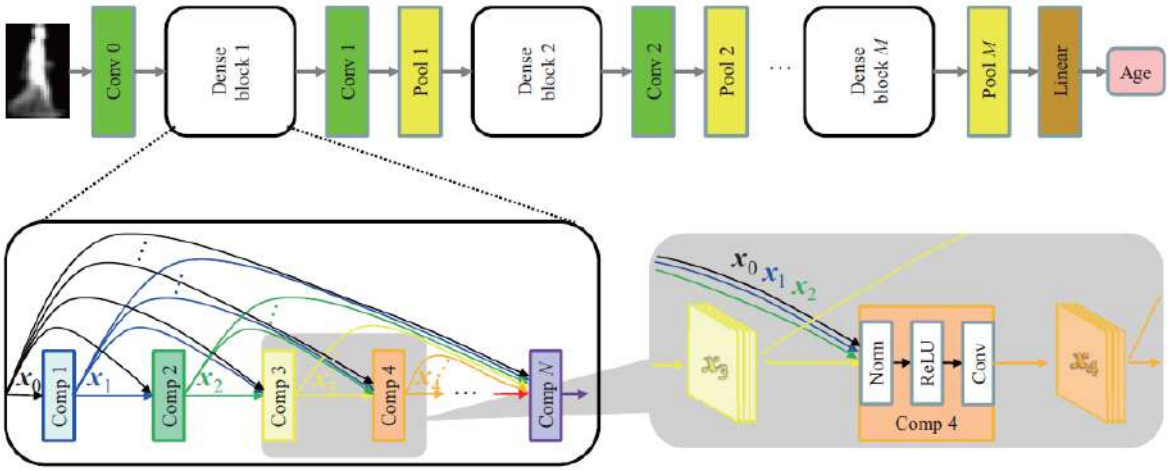

Figure 4.24 : DenseNet architecture proposed by Sakata et al. for gait-based age estimation [171]. © 2019 Springer Nature

observed when using less than 5 blocks, whereas the number of layers did not seem to have a significant impact. A MAE of 5.79 years was achieved on the OULP dataset, with 22.5% of the estimations falling within a 1 year range, 55.9% within 5 years, and 80.4% within 10 years.

Zhang et al. [175] have performed age estimation using a ResNet [176] inspired architecture. The shallowest ResNet architecture (ResNet18) was used as a base and adapted to the problem of gait-based estimation. The main difference between DenseNet and ResNet comes from the fact that skip connections are restricted to a single block in the ResNet. The proposed architecture is illustrated in Figure 4.25. Using the Adam optimization [177] with a learning rate of 0.0001 for gradient descent over 200 epochs with a batch size of 128, the authors have achieved a MAE of 5.58 years on the OULP dataset without taking gender into account, and 5.47 years when taking it into account.

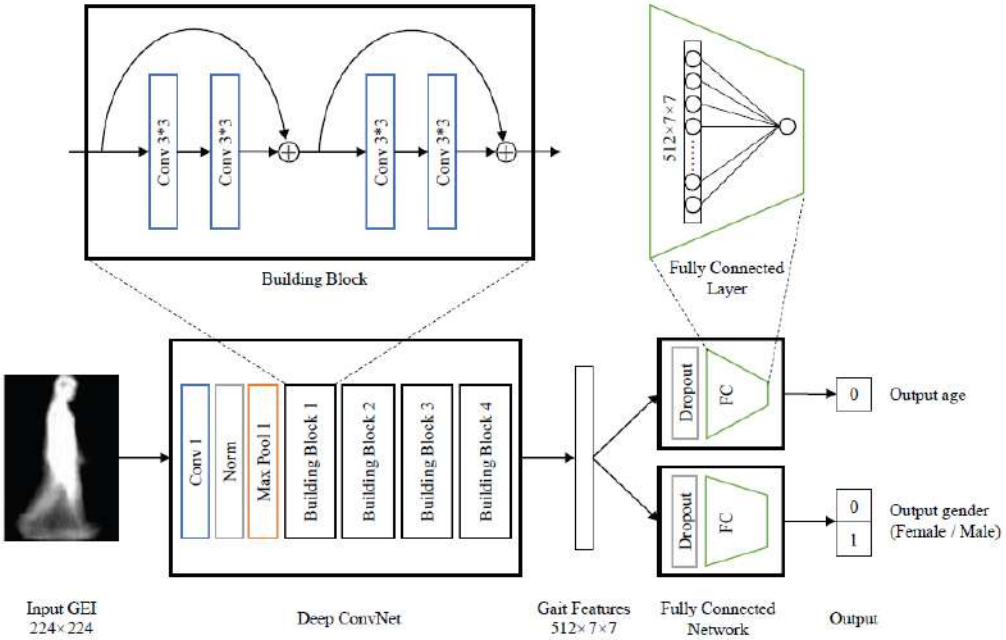

Figure 4.25 : CNN architecture proposed by Zhang et al. for gait-based age estimation [175]. © 2019 IEEE.

Li et al. [178] have used GANs to deal with gait covariates. More specifically, they have used a GAN that takes as an input an image of a subject walking with a bag, and gives the version of that same person walking without the bag as an output. At test time, the network is fed with a GEI of the subject, and the GAN first gives a GEI without any carried object as an output (regardless of the input), and feeds this generated GEI to the age regression part of the network that estimate the age of the subject. An overview of the system is shown in Figure 4.26. They have achieved a MAE of 6.15 and 7.07 years respectively on subjects without and with carried objects, using the OULP dataset. As mentioned before in this thesis, dealing with gait covariates is a big challenge in gait recognition as well as profile recognition. Even though this method is successful, it does imply having access to the same subject walking with and without a backpack in order to train the GAN.

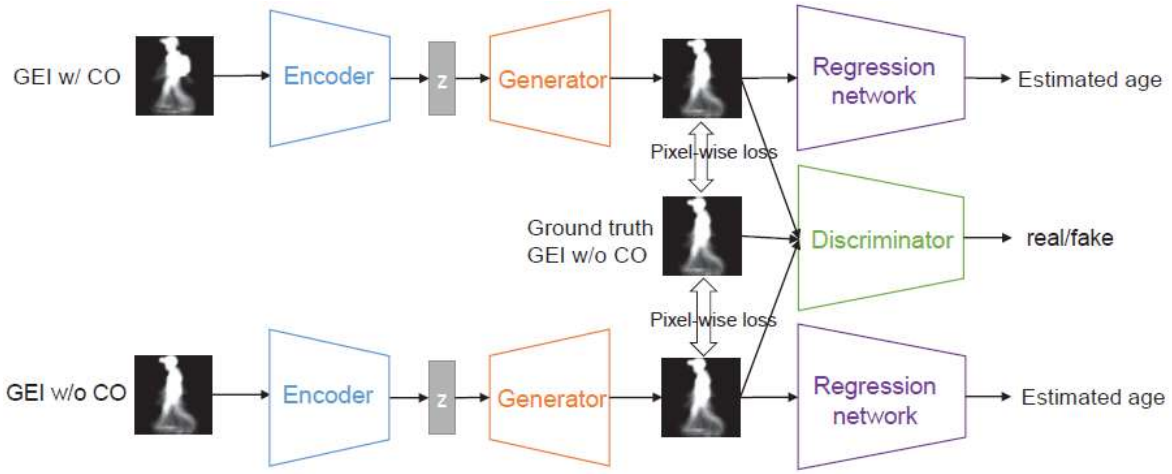

Figure 4.26 : Illustration of the architecture of the GAN-based age estimation method proposed by Li et al. [178]. © 2019 IEEE.

Zhu et al. [179] have combined a CNN working on full GEI images, as well as 3 CNNs working on partial features to perform gait-based age estimation. In this case, the authors have considered the task of age estimation to be an ordinal regression task, where the age is estimated through a series of binary classification problems, which allows to capitalize on the ordinal relationship of age labels. For K age ranks, $K-1$ binary classifiers are used to predict the age rank of a specific image, using a custom distribution loss to capitalize on the relationships between different age ranks. The proposed architecture is shown in Figure 4.27. VGG16 [180] was used as the backbone for the CNN, and trained using Adam with a learning rate of 0.0001, a batch size of 300 and 300 epochs. Using this approach, the authors have achieved a MAE of 5.12 years on the OULP-Age dataset, with 66.95% of estimations falling within a 5 year range.

A recurring theme that has been observed throughout this section is the link between age and gender. In recent papers, it would seem like this trend is only confirmed, as new

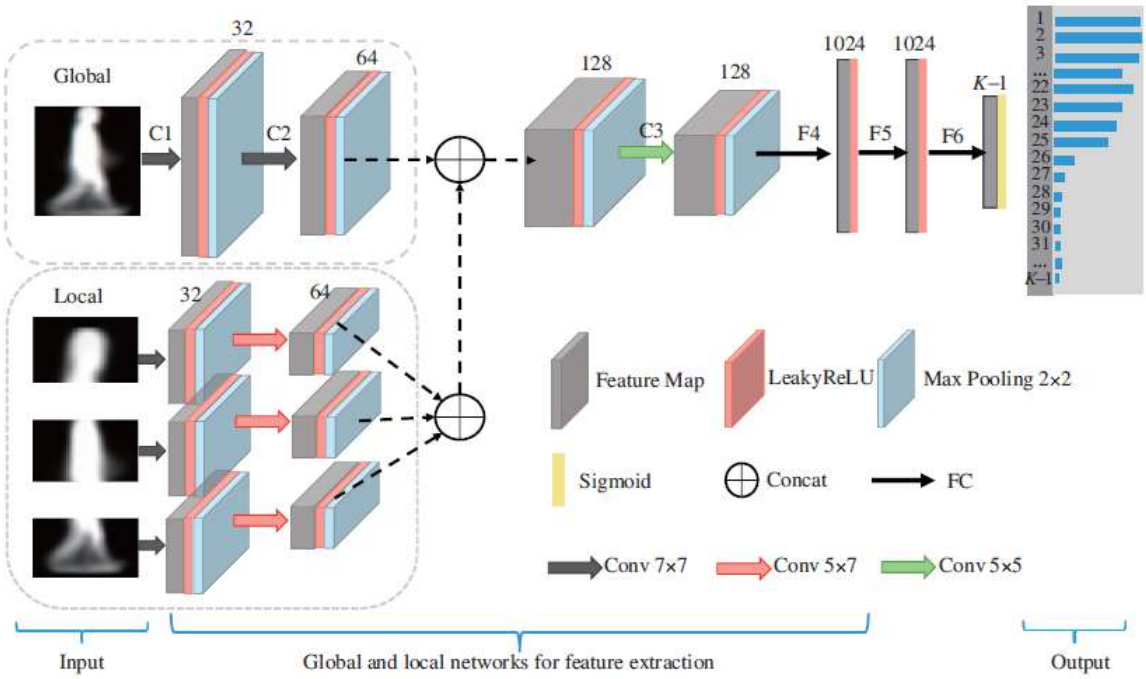

Figure 4.27 : ODR-GLCNN architecture proposed by Zhu et al. [179]. © 2020 Springer Nature.

approaches take into account both parameters. The direction of the correlation seems to be identical across these works as well: gender helps improve age classification and/or regression, but age is not used to improve gender recognition accuracy. This intuitively makes sense, as gender seems to affect gait and how it evolves as individuals age. Marin-Jimenez et al. [181] have shown that jointly training several gait-based tasks, namely age, identity and gender, the identification task converges faster, as illustrated in Figure 4.28.

Abirami et al. [182] have used a similar age and gender label encoding to what was presented in the previous subsection [167], and have achieved a 3.7 years MAE on the USF dataset using k-NN. Xu et al. have gone a step further in age and gender estimation by starting with a single image, reconstructing an entire gait cycle, and performing estimation on that

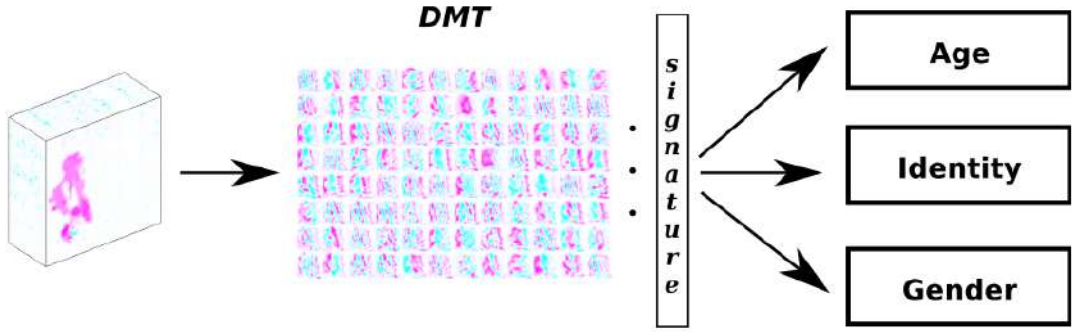

Figure 4.28 : Optical flow and CNN-based architecture for multi-task learning presented by Marin-Jimenez et al. [181]. © 2017 IEEE.

sequence of images [183]. They have achieved a 94.27% CCR for gender recognition and a 8.39 years MAE on the OU-MVLP dataset.

Even more recently, Russel et al. [184] have extended the task to gender, age group as well as carried object recognition using GEI and a complex parallel CNN architecture. They have achieved up to 100% gender recognition accuracy under different circumstances and using different custom parameters on the CASIA-B and the OU-ISIR dataset, with better performance on the former on average. In terms of age group classification and considering 5 groups (0-5, 5-11, 11-15, 16-60, and >60) they have achieved 95.99%, 96.29% and 93.53% accuracies for female only, male only, and both at the same time, respectively on the OU-ISIR gait dataset. These results highlight the age-gender correlation once again. Additionally, they have been able to perform carried object detection (absence or presence of an object) with up to 99.79% accuracy, and classification between 7 types of objects with diagonal confusion matrix values ranging from 41.27% to 86.53% accuracies, depending on the object.

4.1.4 CONCLUSION ON GAIT-BASED AGE ESTIMATION

Gait-based age estimation has come a long way in the last two decades, starting from marker-based binary age classification on a very small dataset of subjects belonging to two drastically different age ranges, to gait-based multi-task classification and regression methods combining several biometrics aspects and relying on complex deep learning architectures.

The trend in age estimation has followed that of gait recognition where the use of deep learning has supplanted non-deep learning approaches, as work with images and sequence of images are particularly suited to these algorithms, and as the complexity of the classification and regression problems require the use of more advanced models. Generally speaking, the wider the age range between different groups, the better the classification results. In this way, it is easy to distinguish small children from adults and elders but most of the challenge occurs when dealing with subjects belonging to neighboring age ranges. Gait covariates still play an important role, whether it is a variation in clothing, view angle, accessories, or carried objects. Generative-based methods have been used to try and circumvent that issue.

As highlighted throughout the previous sections, the correlation between age and gender has been pointed out in many different papers, which comes back to the fundamental challenge of trying to encapsulate what a *profile* is, in order to classify different individuals. The results obtained by relying on a single biometric aspect seem to be capped, we therefore expect more research in this field to move towards the fusion of different aspects of profile and the use of known biometrics to estimate unknown ones. With advances in deep learning and the massive amount of data collected worldwide, we could also expect a shift in perspective, from the manual annotation and minute analysis of existing labeled datasets, to the use of massive data and deep learning algorithms to discover natural clusters and gain a new understanding when it comes to human profiles.

4.2 GENDER RECOGNITION

Gender recognition has been another area of interest when it comes to gait-based biometrics. As described in the previous section, age and gender have been found to be correlated, therefore, being able to perform gender recognition can have positive implications for age estimation. Since gait in general has been used for individual identification in applications where the individual's face is not available, gender recognition is useful for various applications. In the context of this thesis, we are interested in seeing if gender can be derived from low resolution thermal images in the city, partly for the technical challenge, and partly for the implications in terms of smart cities. Indeed, on top of accessibility measures that could be derived from age and mobility recognition, gender recognition could give insights into the usage of the infrastructure of the city, or the safety of certain portions of the city.

4.2.1 NON DEEP-LEARNING APPROACHES

Early gait-based gender recognition efforts seem to have originated from Davis [185] as well, in 2004. Point-light images were used by the authors, where each walking sequence is made of a set of postures associated with a gender. The point-light images are illustrated in Figure 4.29. The authors have achieved a 95.5% gender recognition accuracy using this method on a custom dataset of 40 subjects.

Huang et al. [93] have performed ellipse-based feature extraction on silhouettes captured at three angles (0 °, 90 ° and 180 °), as illustrated in Figure 4.30. For side silhouettes, 7 ellipses are used, as opposed to 5 for front and back silhouettes. The intuition for this split, as explained by the authors, is that men tend to exhibit more upper body lateral sway than women, which is captured by the two ellipses used for the upper body. The parameters of the ellipses (centroid, elongation of the ellipse, orientation of the major axis) are used as features,

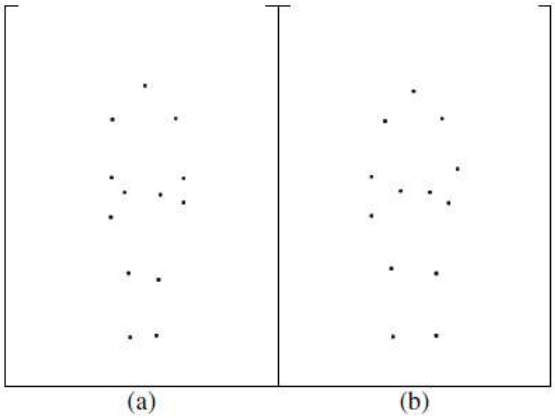

Figure 4.29 : Point-light images used by Davis et al. for gender classification. (a) female walker, (b) male walker [185]. © 2004 IEEE.

and a similarity measure is computed before using a SVM to perform gender classification. The authors have achieved a 89.5% accuracy with a fusion from all three view angles on the CASIA dataset using this approach.

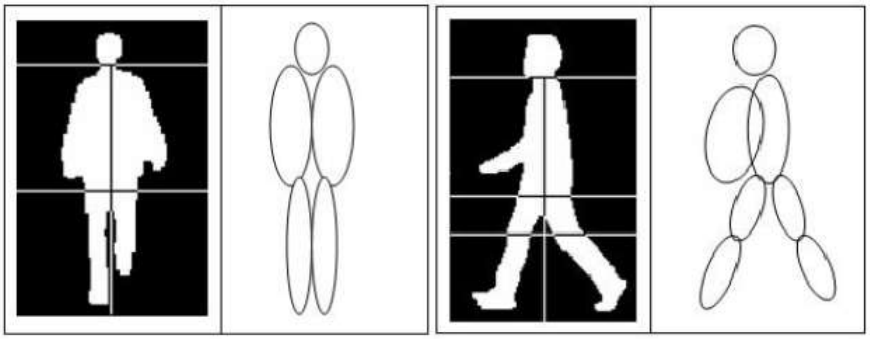

Figure 4.30 : Ellipse-based silhouette division used by Huang et al. [93]. © 2007 Springer Nature.

Based on the literature review, the general consensus in gait recognition seems to be that capturing silhouettes at a 90° angle is optimal, as it captures the most dynamic gait information. However, results presented by Huang et al. [93] seem to show that the use of silhouettes captured at a 0° and 180° angle give better results for gender recognition, leading

to the observation that a side view angle might not be optimal for gender classification. A year later, Zhang et al. [186] have investigated that issue and found that an oblique angle, in their case, 60°, yields better results, with a 91.7% accuracy using SVM on a custom gait dataset. However, the results for the 0° and 180° were lower (78.7% and 82.3%), which still leads to the conclusion that any conclusion on view angle is very heavily datat-dependent. The inclusion of face or hair from the front and back view could also be a big giveaway when it comes to gender.

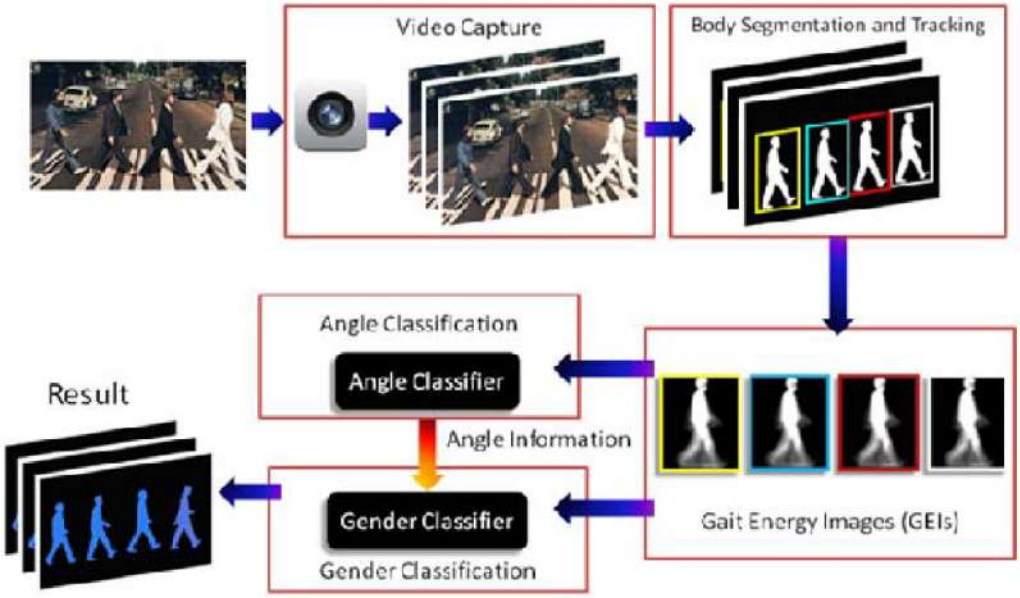

Figure 4.31 : Architecture of the real-time gender classification system presented by Chang et al. [187]. © 2009 IEEE.

Following this trend, Chang et al. [187] have built a gender classification system that takes into account the view angle of the silhouettes. In this case, the CASIA dataset was used, with view angles ranging from 0° to 180° in 18° increments. The idea is to extract the GEI of gait sequences for all 11 angles, use Linear Discriminant Analysis (LDA) to perform dimensionality reduction, and build a gender classification model for each angle. In an effort

to build a real-time classification systems, the authors have performed angle classification: based on a new sequence of images, the GEI is computed and the view angle is estimated before picking the right gender classifier to perform angle-targeted gender classification. The architecture of the system is illustrated in Figure 4.31. The highest accuracy was reached at 90° and 108° angles (99.05%), and the lowest at 0° (93.70%).

The main issue with this approach is that the angle classification (determining the view angle of a previously unseen gait sequence) only reached an accuracy of 66.02%, leading the system to misclassify the angle more than a third of the time, which leads to the use of the wrong classifier. Reducing the number of view angle groups from 11 to 5 leads to an angle classification accuracy of 96.79% but comes with the cost of a slightly worse accuracy in gender classification. Overall, the authors have achieved an average gender classification accuracy of 84.38% using 10 custom gait video sequences showcasing 2 males and 2 females, thus demonstrating the validity of their approach. Using C++ and OpenCV, their system is capable of handling and processing 63 frames per second (320x120 resolution) on a personal computer with an Intel Core 2 Quad CPU. These results are encouraging for real-time profile recognition, even though in this case, a personal computer is used.

In terms of view angle, Makihara et al. [95] have achieved their highest gender classification results on the 30° and 60° view angles, around 73% accuracy on the OU-ISIR dataset using k-NN. Using Gabor filters, which were presented in the previous section, Hu et al. [160] have achieved a 96.77% for 90° to 90° gender classification. When performing one-to-many view angle gender classification (training on instances at a specific angle, and testing on all angles), they have achieved the best performance using a 54° view angle, with 82.11% accuracy. These results seem to confirm the intuition that an oblique view is more efficient when performing gender classification.

Most approaches in the literature rely on GEI, which has been extensively mentioned in this chapter and in the previous one. Arai et al. [188] have used GEM instead (Gait Energy Motion). GEI simply averages the value of each pixel over a gait cycle, whereas GEM averages the difference in pixels from adjacent frames at each time step (starting from the second frame) over the whole cycle. GEM therefore captures a more dynamic aspect of the gait, and includes a temporal component, where the movement between each frame is emphasized, rather than the position at each frame. This is an effort from the authors to extract spatio-temporal components of the individual's gait in order to perform age classification. The equation for the GEM over a sequence of frame is expressed in equation 4.3.

$$F(i,j) = \frac{1}{T}\sum_{t=2}^{T} |I(i,j,t) - I(i,j,t-1)| \tag{4.3}$$

Where T is the number of frames in a sequence, and $I(i,j,t)$ is the pixel intensity at the coordinates (i,j) in frame t. Using this method, they have achieved a 97.63% accuracy for gender classification on a dataset comprising 31 males and 31 females, with a slight improvement in results when including spatial and temporal information, rather than just spatial information (97.47%).

Following the trend of exploiting dynamic information, and concluding this subsection covering conventional approaches, Liu et al. [189] have used Dynamic GEI (D-GEI) to capure more dynamic information about the arms and legs swing to perform gender recognition. The GEI is first extracted over a gait cycle, like other approaches in the field, from which the D-GEI is computed, from which HOG features are extracted, which have been described in the previous chapter. The process of extracting the D-GEI is illustrated in Figure 4.32.

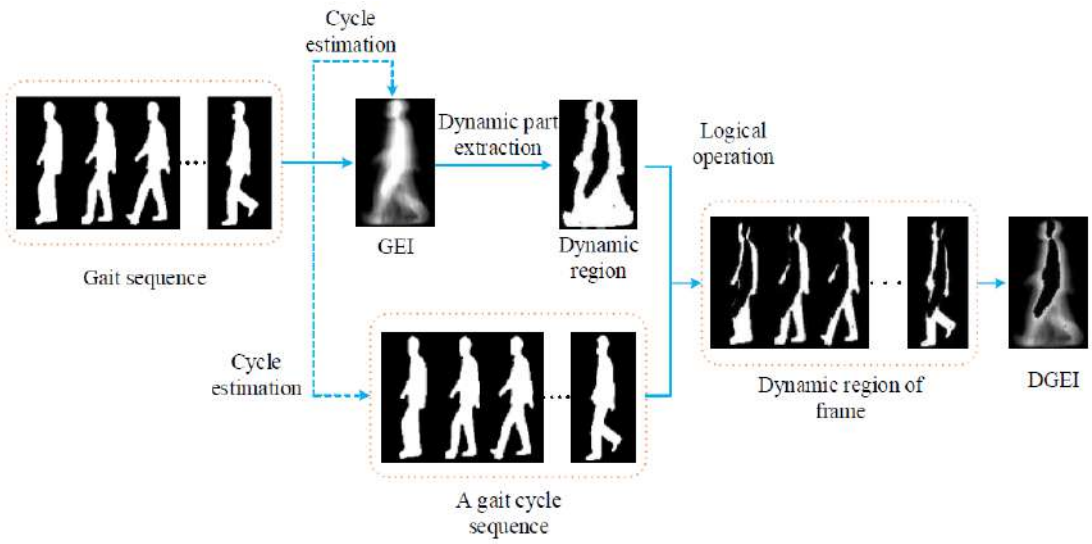

Figure 4.32 : Extraction of Dynamic Gait Energy Image (D-GEI) over a gait cycle [189]. © 2018 IEEE.

Using a SVM for classification, the authors have found that their methods performs better than methods relying on the GEI, especially considering clothing variations, where they have achieved a 88.25% accuracy using HOG on D-GEI, as opposed to a 79.06% accuracy using HOG on GEI. This method is on par in terms of performance when considering normal conditions (97.42% vs 97.48%) and individuals with a backpack (93.91% vs 93.86%) on the CASIA B dataset.

4.2.2 DEEP-LEARNING APPROACHES

Following the trend in gait recognition and age estimation, recent approaches have turned towards deep learning. Liu et al. [190] have experimented with CNNs by changing the classification part of the network from the conventionally used sigmoid to a SVM. The CNN takes GEI of gait sequences as an input and learns sets of filters to extract feature maps, as described in the first chapter. These feature maps are flattened when reaching the dense layers

of the CNNs, which are then used to train a SVM. The authors have used the VGG16 [180] CNN architecture presented in Figure 4.33.

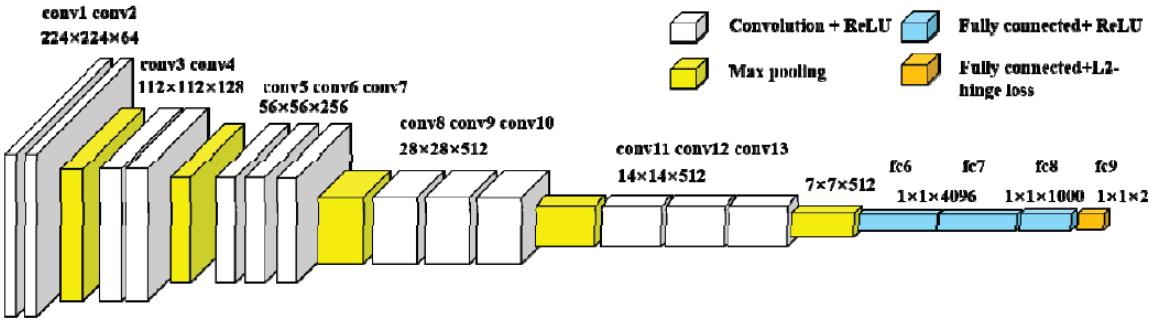

Figure 4.33 : VGG16 CNN architecture used by Liu et al. [190] The features in the dense layers are used to train a SVM to perform gait-based gender recognition. © 2019 IEEE.

The authors have found that the use of a SVM trained on extracted features, instead of using a standard softmax activation in the last layer yielded better results for gender classification on the CASIA dataset (89.62% vs 87.10%). Another interesting observation with preliminary results has been that early dense layers (fully connected layers) seem to give more expressive features, as a SVM trained on the first dense layers performed better (87.94%) than using feature from the second layer (81.92%) or the third layer (80.79%). These layers have a dimension of 4096, 4096 and 1000 respectively.

Following a concept explored in the previous section, Kitchat et al. [191] have exploited view angle to perform gender recognition. In this case, the angle is determined based on the pre-processed images, and once the angle is determined, the GEI is extracted before performing gender classification using a CNN. The authors have tested their system on the CASIA dataset as well as on a custom *freestyle* gait dataset that they have collected using a

Kinect camera, called SIIT-CN-B. An example of the images of this dataset together with the extracted GEI are shown in Figure 4.34.

Figure 4.34 : Freestyle dataset collected by Kitchat et al. [191] with the associated extracted GEI. © 2019 IEEE.

The authors have found that without taking angle into account, they achieve a 79.93% CCR for gender recognition. This accuracy reaches 90.74% when taking both gender and view angle into account, which highlights once again the importance of view angle in classification results. The CNN architecture used in this work is presented in Figure 4.35. This model has also been compared with the previous VGG16 model mentioned, used by Liu et al. [190], pre-trained with ImageNet weights. VGG16 only reached an accuracy of 56.17% of the custom dataset. Based on this observation, as well as conclusions from the work of Smith et al. [192], studying transfer learning with deep CNNs for gender recognition and age estimation, the authors have conjectured that large dimensionality dense layers seem to lead to worse

performance when it comes to gender recognition. This is also reinforced by the fact that the VGG16 architecture uses a first dense layer of dimensionality 4096, whereas this work uses a first dense layer containing 1024 units only, and performs much better.

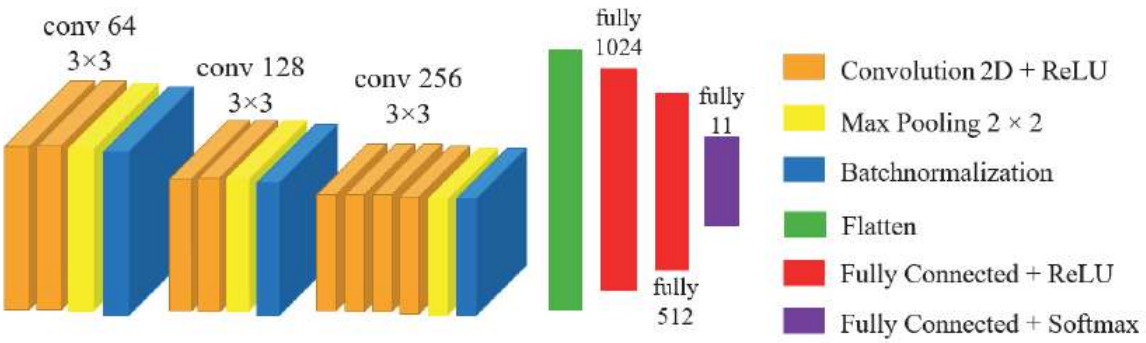

Figure 4.35 : CNN architecture used by Kitchat et al. [191]. © 2019 IEEE.

On the CASIA dataset, an accuracy of 97.58% was achieved *without* using the angle, and it dropped to 89.25% when including angle estimation. These results may seem to contradict the idea that knowing the angle leads to a steady improvement of the models, however, the authors have mentioned that the CASIA dataset contains the same number of subjects for each view angle, removing any possible imbalance that could lead to an improvement in results by taking view angle into account. CASIA also contains full silhouettes only, whereas SIIT-CN-B contains a lot of partial silhouettes. Taking angle into account could therefore be more efficient on difficult datasets with partial silhouettes and possible imbalances.

Revisiting a paper that was mentioned in the age regression subsection [175], the emphasis is now put on gender recognition, which was carried out as an auxiliary task by the authors. A ResNet-based CNN architecture was used, and the authors have reached a 98.26% CCR with optimal parameters on the OULP dataset, which is a much bigger dataset than the CASIA dataset used in the few previous papers (63846 vs 124 subjects). Looking at

the architecture of the CNN, the authors have removed the last 2 layers of ResNet-18, which are the dense layer and an average pooling layer. The last layer before the classification unit is a 512x7x7 set of feature maps, which is connected to an output layer containing 2 units, each returning a probability of an instance belonging to the male or the female class.

The system presented by Xu et al. [183] was described in the previous section as well. A GAN is used to reconstruct a full gait cycle from a single image in order to perform age estimation and gender recognition. The authors have reached a 94.27% CCR on the OU-MVLP dataset (10307 subjects) which was chosen because of its high variation in view angles and its large size. Another interesting aspect of this paper, despite the ability to reconstruct a full gait cycle from a single image and to perform efficient gender recognition, is the online gender recognition system presented by the authors, which is relevant to the long-term vision of this thesis.

They have presented two systems, a client-server architecture, and a standalone system. In the first architecture, the client sends an image over to the server, which returns its prediction about the gender of the observed individual. However, the communication latency makes it difficult to perform real-time classification at a rate that would match the recording rate of the camera used (Microsoft Kinect v2, 30 fps). This can be solved by using only a few frames, which still leads to good results, because the system is designed to work with a single frame. The standalone system relies on a computer equipped with a NVIDIA GeForce RTX 2080 TI and is able to perform real-time gender classification at the framerate of the camera. The image is captured, processed, the background is automatically subtracted using the depth component of the camera, resized, before the gait cycle can be reconstructed and the classification can be performed. Snapshots of the standalone systems are shown in Figure 4.36.

Figure 4.36 : Screenshots of the online standalone gender recognition system presented by Xu et al. [183]. © 2021 IEEE.

4.2.3 CONCLUSION ON GAIT-BASED GENDER RECOGNITION

In the previous section, we have covered gait-based gender recognition approaches, and observed a similar trend towards deep learning more recent years. Even though gait covariates, such as clothing and accessories are still a challenge, the question of the view angle seems to have been studied more for gender than for age estimation.

In the case of age estimation, parameters such as head tilt, arm and leg swing as well as the general posture are a good indicator of the general age range an individual belongs to. When it comes to gender however, upper body lateral sway seems to play an important role, and an oblique angle seems to capture more relevant dynamic features than a side view. Nonetheless, some papers seem to have reached opposite conclusions when it comes to view angles, which is understandable because of the nature of the problem: depending on the way the problem is framed, the dataset used (size, data format, full or partial silhouettes, view angles...), and the method used (extracted features, deep learning model architecture), there is a lot of room for variability, and it is not realistic to expect a general consensus to be reached with so many different parameters at play.

4.3 RELATED WORK WITH THERMAL IMAGES

To conclude this chapter and the related work part of this thesis, we review existing work performing gait related tasks using thermal images. We begin by defining what *thermal* images are before reviewing related work and its limitations, as well as how it applies to this thesis.

4.3.1 THERMAL IMAGES

Thermal cameras used to be a specialized and expensive tool used by professionals such as electricians or plumbers, to detect overheating or leaking pipes. They have also been used by mechanical engineers in buildings, as well as in airports, to detect fever in passengers, which has been useful in the context of the current pandemic, at the time of writing this thesis. These use cases are illustrated in figure 4.37.

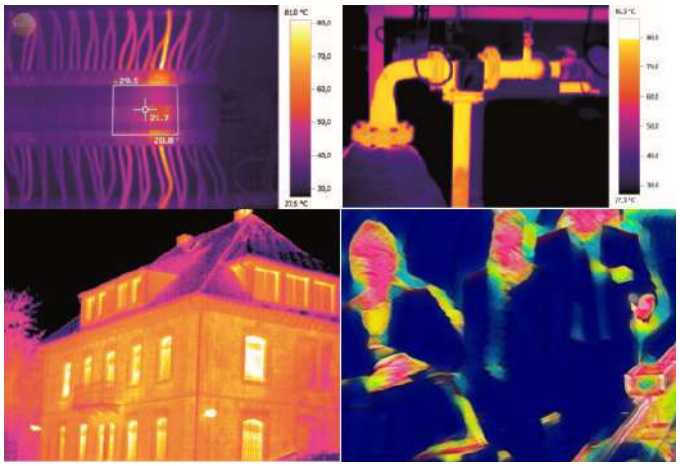

Figure 4.37 : Example of thermal images applications, from left to right and top to bottom: electrical engineering, plumbing, mechanical engineering, and airports [193]. © 2018 Pyrosales.

Recently, these devices have become cheaper, smaller, and more accessible, which has opened up a lot of opportunities, this thesis being one of them.

4.3.1.1 DEFINITIONS

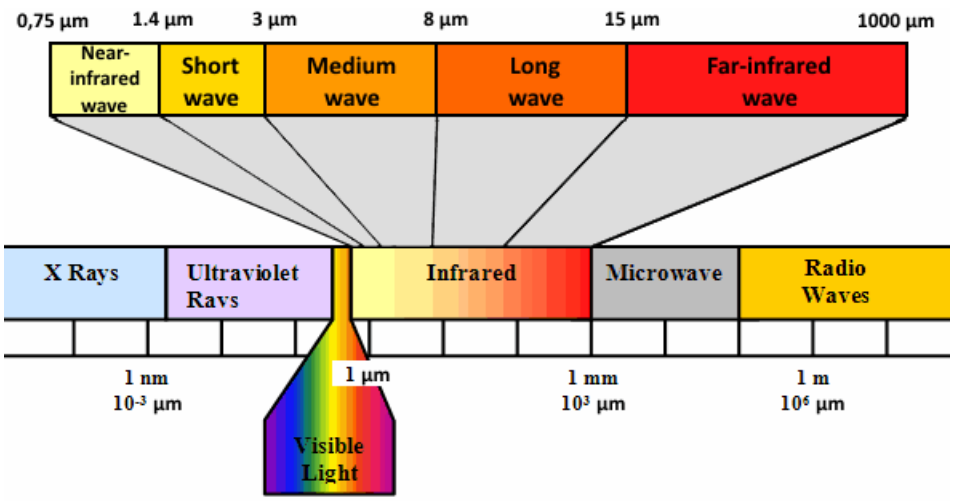

Figure 4.38 : Electromagnetic spectrum with an emphasis on the infrared region [194]. © 2022 Ellis Amalgamated.

Standard RGB cameras, which have provided the images for most of the work in the field of gait recognition, work in the visible light spectrum, where light bouncing off objects in the scene is captured to form an image. Each pixel in the camera's sensor captures the red, blue and green components of the light, forming a RGB image. Thermal cameras however, are more commonly referred to as thermal sensors, as they do not work with light, but they detect the difference in heat in the observed scene in order to form an image, based on the radiation emitted by every object or subject in the scene. The electromagnetic radiation spectrum is shown in Figure 4.38, with an emphasis on the infrared range, sitting just above the visible light spectrum.

The infrared range is divided in five categories depending on the wavelength of the radiation [194]:

- Near-Infrared Radiation (NIR): 0.75 to 1.4μm wavelength
- Short-Wavelength Infrared Radiation (SWIR): 1.4 to 3μm wavelength
- Mid-Wavelength Infrared Radiation (MWIR): 3 to 8μm wavelength
- Long-Wavelength Infrared Radiation (LWIR): 8 to 15μm wavelength
- Far Infrared Radiation (FIR): 15 to 1000μm wavelength

Generally speaking, we talk about *thermal* images for longer infrared wavelength, and *infrared* images for shorter wavelength, even though they all fall within the infrared range. However, It is important to point out the difference between thermal and infrared images. Even though both types of image operate within the infrared range, NIR cameras project infrared light on a scene, and generate an image based on the infrared energy that is reflected back to the sensor. This makes NIR cameras active, whereas thermal cameras, such as the FLIR Lepton used in this thesis, are passive, and reconstruct an image purely based on heat information.

4.3.1.2 MOTIVATION AND CHALLENGES

The main interest of using thermal images rather than standard, visible light images, is the fact that any object above the absolute zero temperature emits infrared radiations. It means that even without the presence of light, objects can be detected by infrared sensors, which is useful at night or in low light environments, as explored in the next section. Thermal imaging systems built on top of a thermal sensor construct an image using a color gradient where colder regions are darker, and warmer regions are brighter. This is illustrated by some of the images collected for this thesis, using a FLIR Lepton 2.5, which captures EMR in the LWIR range (8 to 14μm wavelength), in Figure 4.39.

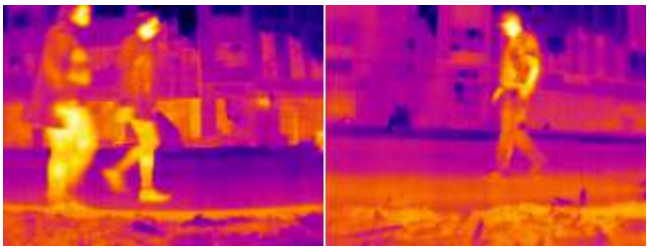

Figure 4.39 : Thermal images collected by the FLIR Lepton 2.5 during the summer 2020 experiment. © Rani Baghezza, 2022.

In the context of gait recognition in a public space such as the city, the use of thermal images allows to protect the privacy of citizens. Indeed, a dense deployment of RGB cameras would be a huge privacy concern, as anyone having access to the images collected by the system could identify a specific person, as well as their location and the time they have stepped in front of the camera. By removing the identity component with the use of thermal images, and by moving towards a real-time system where images are not stored longer than necessary, such a system will, by design, be much more privacy-aware.

Using thermal images also allows to avoid some of the challenges faced with RGB images, such as low lighting conditions (low ambient light, shadows), or on the other end of the spectrum, over exposure during a bright, sunny day. As explored throughout this literature review, gait covariates, such as clothing and accessories are a challenge. An unfortunate data distribution using RGB images could learn wrong correlations, such as linking a specific clothing color or pattern with a specific profile, which could just be a complete coincidence, or a temporary fashion trend. Since thermal images only capture difference in heat, these issues are much less prevalent.

However, the use of thermal images also brings additional challenges, the first of which is the variation in temperature. Images captured in different weather conditions can present a very high variation in color, as illustrated in figure 4.40. This can be an issue on many

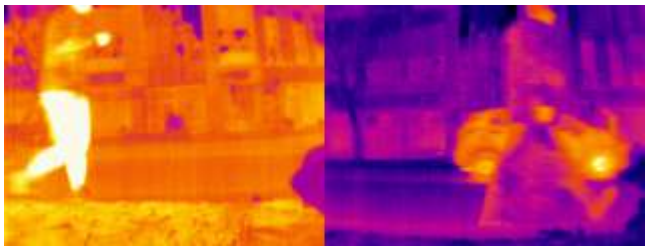

Figure 4.40 : Example of the variation in temperature in thermal images. Both images are captured in the same location, on different days, leading to a high difference in temperature, and therefore, color in the images. © Rani Baghezza, 2022.

different levels. When it comes to deep learning, a model trained with a specific distribution could be led to believe that bright images are correlated with a specific profile. From an image analysis point of view, as observed in the picture on the left, very hot parts of the image tend to lose all texture component, and appear as a yellow or a white patch that can be difficult to apprehend by the model. The amount of heat picked up by the sensor is also clothing-dependent. Insulation layers worn by the pedestrian on the right make the silhouette darker, whereas the silhouette on the left appears much brighter thanks to a combination of a higher external temperature, possibly darker and thinner clothes, allowing a higher amount of heat to be picked up by the sensor.

Additionally, thermal images are less sharp than infrared images, due to the fact that heat information, not reflected light, is used to reconstruct the image. This leads images to capture less texture information and less visible detail, which is translated by less information, and a different type of information being contained in the images.

4.3.2 RELATED WORK USING THERMAL IMAGES

Most of the work in gait recognition in the literature has been carried out using RGB images, even though more recent work seems to be focusing on the inclusion of infrared images, and the fusion of both RGB and infrared images.

4.3.2.1 SILHOUETTE EXTRACTION AND GAIT RECOGNITION USING IR IMAGES

The earliest work in IR-based gait recognition seems to be from Tan et al. [195], who built the CASIA Infrared Night Gait Dataset containing 153 subjects (130 males, 23 females), with 320x240 images captured at a 25fps rate. A sample of these images is shown in Figure 4.41. A Gaussian filter is used to remove noise from the image, before subtracting the background and normalizing the silhouette. The authors have then compared the performance of GEI and HTI (Head Torso Image) using a nearest neighbor classifier. At best, they have achieved a 96% recognition rate using GEI, and 94% using HTI. This has shown that gait recognition was possible on infrared images captured at night. However, the background is uniform, and a lot of pre-processing takes place, virtually leading to a similar setup used in most gait lab-based gait recognition approaches.

In 2006 as well, Goubet et al. [196] have tackled the challenge of pedestrian detection using fusion of infrared and RGB images. They have used a Thermal-eye 2000B operating at a 8-12µm wavelength range (LWIR), capturing infrared 320x240 pixel images, and a Handycam PC105 capturing RGB images at a 640x480 resolution. As mentioned in the previous subsection, thermal images tend to capture less texture as well, which is illustrated in Figure 4.42.

Figure 4.41 : Example of infrared images captured at night under different weather conditions by Tan et al. [195]. From left to right: clear day, cloudy day, low contrast on a cloudy day. © 2006 IEEE.

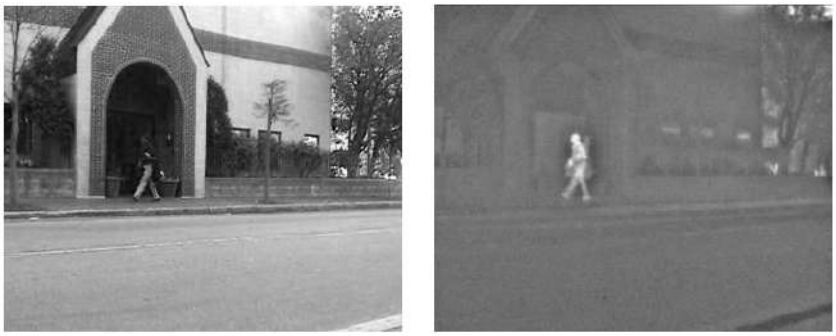

Figure 4.42 : Example of RGB (left) and infrared images (right) captured by Goubet et al. [196]. © 2006 SPIE.

The authors have highlighted some of the challenges of using thermal images, especially when it comes to detecting silhouettes: generally speaking, one can assume that the silhouette will be warmer than the background. However, in summer on a hot day, the background could be warmer, rendering a simple temperature threshold-based background subtraction inefficient. Alternative methods have been mentioned, such as relying on both temperature, and the contrast between pixels. This way, columns of pixels can be extracted, pixels above a certain threshold can be counted, and the contrast between bright and dark pixels can be computed, thus isolating a silhouette, column by column. However, such methods can be computationally intensive.

They have also highlighted that the amount of infrared light coming from a pedestrian highly depends on the temperature and infrared absorption coefficient of that pedestrian's clothes, which was observed during the experiments of this thesis. In order to combine the best of both worlds, they have performed fusion of RGB and IR images by performing a weighted sum of the value of each pixel in both images, as illustrated in Figure 4.43. This allows the presence of both texture and temperature information in the same image. The authors have found that infrared images allowed to extract the foreground pixels (silhouettes) much better than using visible images only, or a fusion of both RGB and IR images.

Figure 4.43 : Example of two different fusions of IR and RGB images, where different coefficients have been used to combine the images [196]. © 2006 SPIE.

In 2009, Ming et al. [197] have used a A40M infrared camera (7.5 to 13 µm wavelength) to capture gait data from a set of 23 subjects under four different conditions: normal walking, carrying a 5kg load, carrying a volleyball, and wearing cotton clothes. Samples are shown in Figure 4.44.

The authors have extracted various gait features and extracted the skeleton of the silhouette to perform gait recognition using a SVM. They have achieved a 92% accuracy at

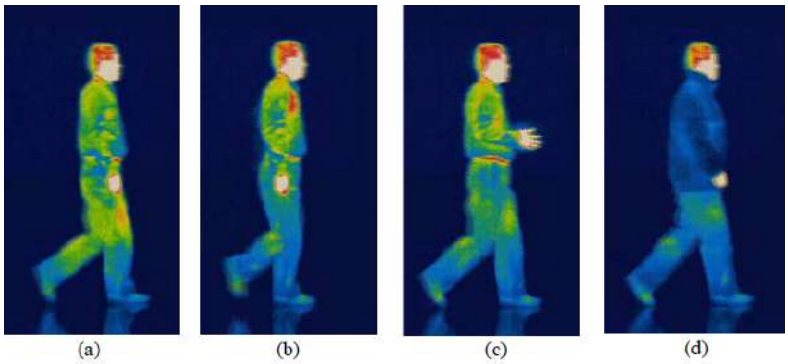

Figure 4.44 : Images collected by Ming et al. [197] under different conditions: normal walking (a), carrying a 5kg load (b), carrying a volleyball, and wearing cotton clothes. © 2009 IEEE.

best, and they have found that wearing cotton clothing affected the results more than carrying a volleyball or a 5kg load.

Decann et al. [198] have used the CASIA Night Gait Dataset originally created by Tan et al. [195] to perform silhouette extraction and backpack detection. The process of extracting a binary silhouette is carried out after adjusting the contrast, as shown in Figure 4.45, as thermal images generally show a lower silhouette to background contrast, making it difficult to extract a silhouette without pre-processing.

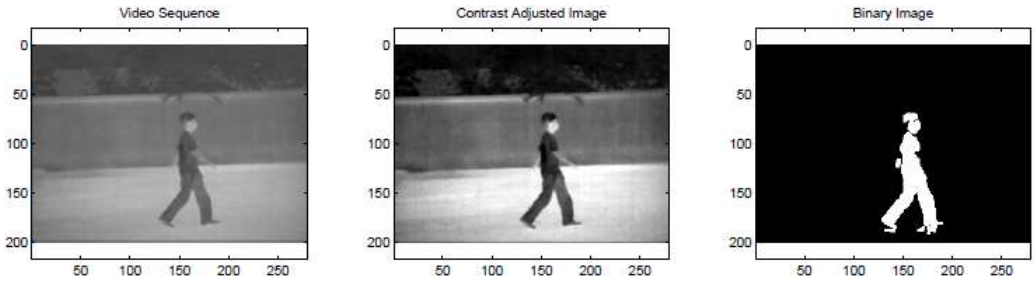

Figure 4.45 : Process of contrast adjustment and binary silhouette extraction by Decann et al. [198]. © 2010 SPIE.

Once the silhouette is extracted, static features (silhouette height, coronal plane) and spatiotemporal features (gait curve) are extracted. The average gait curve is then projected against the coronal plane (vertical plane dividing the body into two halves) to create a 1-D curve that helps identifying the presence of a backpack in the silhouette. Using this method, the authors have reached a maximum accuracy of 98.4% for backpack detection on the CASIA Night Gait Database. Using a similar approach, Bourlai et al. [199] have achieved a 97.7% human gait recognition on the same dataset.

4.3.2.2 GENDER RECOGNITION USING IR IMAGES

The last subsection has summarized gait recognition approaches using infrared images. In order to close the gap between the literature and this thesis, it is important to explore existing profile recognition approaches using gait and infrared images. From the state of the literature, it seems that the only soft biometrics recognition task has been gender recognition, with a first approach from Arai et al. [200] in 2012 on the CASIA Night Gait Database, where a 92.9% accuracy was achieved using 2D-Discrete Wavelet Transform [201].

A few years later, Liu et al. [79] have used a body link model to perform gender recognition on the same dataset. The images are pre-processed in order to extract the silhouette from the background, and to extract the contour of the silhouettes. This process is illustrated in Figure 4.46.

Once the contour has been extracted, the authors have used the body link model illustrated in Figure 4.47. This model is based off the human anatomy and allows to use various ratios to extract joint positions. This approach fits into the model-based approaches for gait recognition, which have been voluntarily excluded from this thesis in general, with the excep-

Figure 4.46 : Image pre-processing for gender recognition on IR images presented by Liu et al. [79]. Creative Commons Image.

tion of this paper, as it is one of the few existing papers dealing with infrared images-based gender recognition.

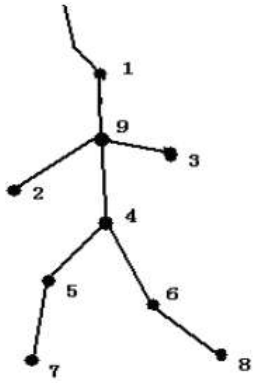

Figure 4.47 : Body link model used by Liu et al. [79]. Creative Commons Image.

Using a custom gait dataset including 20 people between the age of 20 and 40 recorded under different conditions (view angle, carrying a backpack), the authors have compared the results of different machine learning models (SVM, NN, k-NN with $k = 3$). The highest gender classification accuracy (92%) was achieved with a SVM on subjects walking under normal conditions, without carrying a backpack. All in all, SVM scored between 80 and 92%, NN between 69 and 75% and k-NN between 73 and 82%.

All subsequent approaches in gender recognition using infrared images have relied on a combination of visible light and infrared images, in order to combine the best of both worlds. In 2016, Nguyen et al. have used a SVM-based method to combine information from both modalities [202]. A custom dataset is built by capturing gait sequences with a custom-made node including a RGB camera as well as a FIR camera, as illustrated in Figure 4.48. It contains 5852 images for each modality, collected over a pool of 103 people (66 males, 37 females) at various view angles. The camera node is located at a 6m height in an uncontrolled environment.

Figure 4.48 : Samples of the custom-made dataset with a RGB image on the left and a FIR image on the right [202]. Creative Commons Image.

Pedestrians are then located on the image using a method presented by Lee et al. [203], which allows to track pedestrians on both visible and thermal images. Features are then extracted from the silhouette using HOG and MLBP (Multi-Level Local Binary Pattern), which was deemed better than using raw pixels by the authors, because of the high variation in terms of clothing, hairstyle and illumination changes. Two types of fusion were then used: feature fusion and score fusion. In the first case, features are extracted individually from the visible image and its thermal equivalent, the feature vectors are concatenated, PCA is used to reduce the dimensionality, and a SVM is used on the resulting vector. In the latter, score

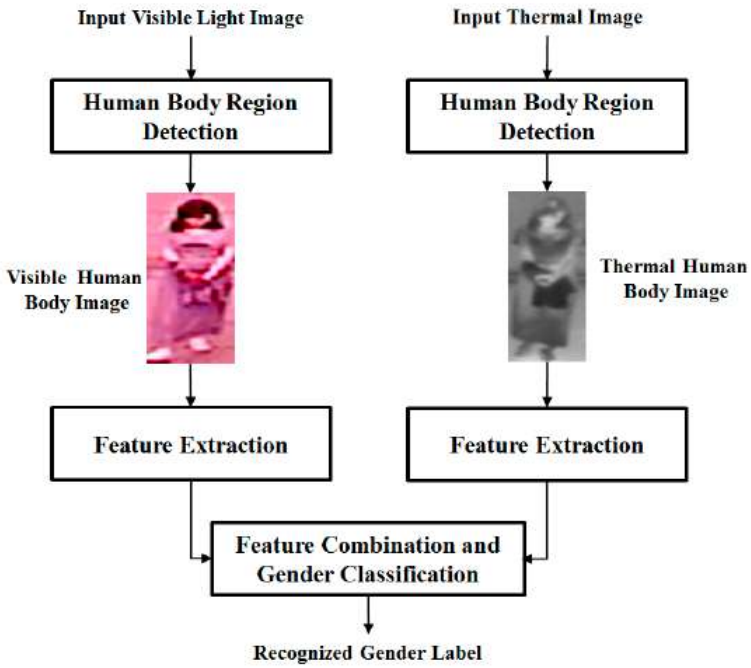

Figure 4.49 : SVM-based system proposed by Nguyen et al. [202]. Creative Commons Image.

fusion is used on the classification result of two individual SVM working on each type of image, in order to reach a final classification decision. An overview of the proposed system is shown in Figure 4.49.

Nguyen et al. [202] have summarized the results obtained under different circumstances: using only visible images, they have achieved an accuracy of 83.46% using HOG, and 80.42% using thermal images and HOG. Using feature level fusion, they have increased the accuracy to 84.05%. Using score level fusion, they have improved these results further, to 85.33%. This shows that the combination of information from both visible and infrared images seems to lead to an increase in gender recognition accuracy. The authors have also evaluate the processing time of their method. Using a desktop computer with an Intel Core i7 CPU (3.5GHz) and 8GB of RAM, they have found that the processing of a visible and a thermal image, as well as

the classification of the individual in these images took on average 27.59ms, leading to the possibility of using this system in real-time with a frame rate of up to 36 fps.

In subsequent work a year later (2017), Nguyen et al. [204] have worked with visible and thermal images as well, using CNNs this time. The architecture used in this work is similar to previous work, as shown in Figure 4.50. As highlighted by the authors, the trend is shifting towards deep learning approaches for several reasons, one of which is that the use of static feature extraction methods such as HOG or LBP (Local Binary Patterns) is starting to show its limitations on more challenging datasets presenting high variations in illuminations and various gait covariates. Deep learning is more suited to these challenging tasks, because of the ability of these methods to discover features by themselves throughout the training process.

The authors have used a custom CNN architecture shown in Figure 4.50, with an input resolution of 183x119 pixels, 5 convolution layers (96, 128, 256, 256, 128) and two fully connected layers (4096, 1024), excluding the final output layer. In order to deal with overfitting issues, the authors have used image augmentation and they have added dropout. In terms of image augmentation, they have randomly cropped pixels on the top, bottom, right and left of the image. In this work again, feature fusion was used, as illustrated in Figure 4.51. A separate CNN is used for each type of image, where the last fully connected layer before the output layer is used as the feature vector. Each image is therefore summarized into a 1024 vector that is concatenated with the vector from the other modality before PCA is used and final gender recognition is performed using SVM. For the score fusion part, the same initial method is used, except a SVM is used on each 1024-dimension vector to obtain an initial score for each modality, before performing score fusion using a second SVM layer and obtaining the final classification result.

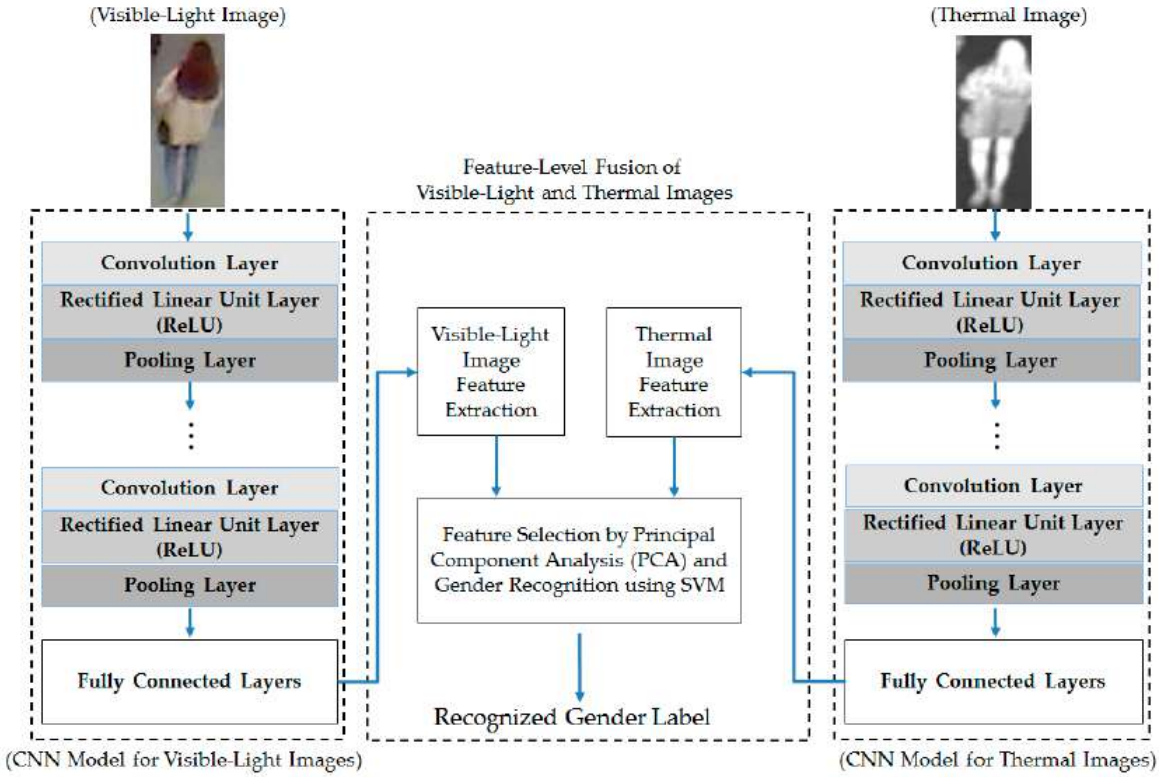

Figure 4.50 : CNN-based system proposed by Nguyen et al. [204]. Creative Commons Image.

For this work, the authors have captured another custom database containing 412 people, 254 males, 158 females. For each person, 10 visible light images and 10 thermal images are captured, for a total of 8240 raw images. Using data augmentation the final training database contains 14,9640 images, and the test database contains 36,600 images. The CNN is trained for 60 epochs with a learning rate of 0.01 that is divided by 10 after every 20 epochs. Using only visible images, an accuracy of 82.78% was reached, as opposed to 83.39% using only thermal images. Using both images, without PCA, an accuracy of 88.32% was reached using feature-level fusion and 88.15% using score-level fusion. These results have been slightly improved using PCA, to 88.56% and 88.29% respectively.

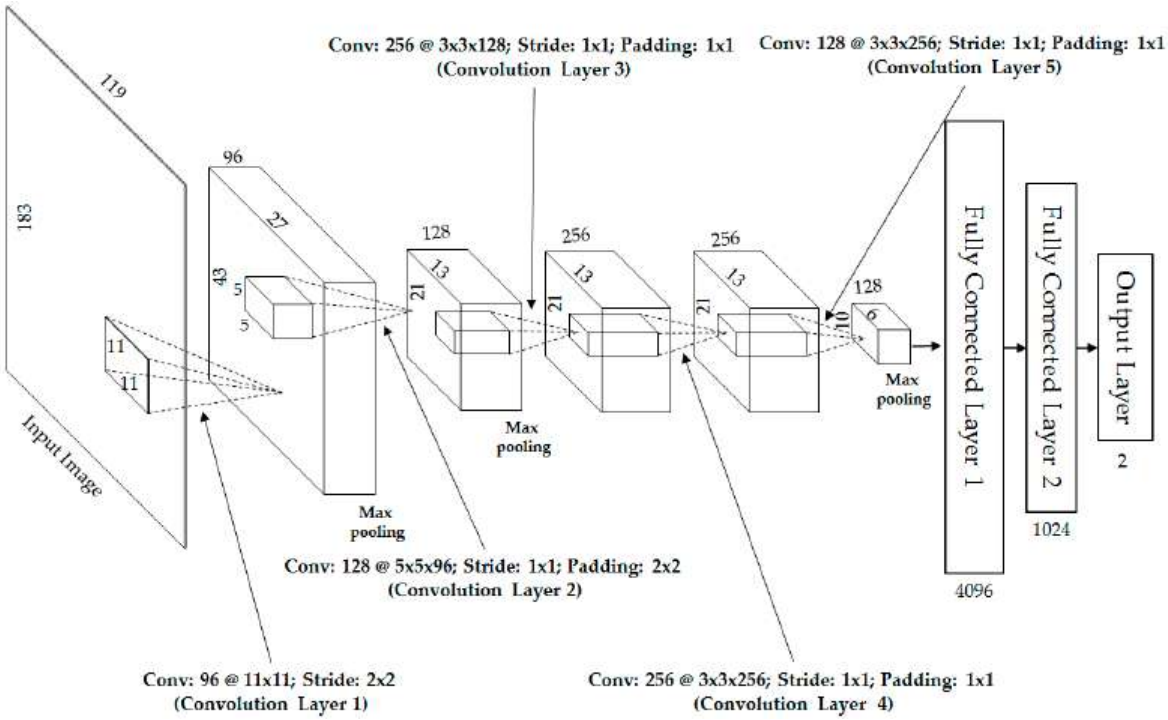

Figure 4.51 : CNN architecture proposed by Nguyen et al. [204]. Creative Commons Image.

Overall, using deep learning seems to show better performance on thermal images than on visible-light images, as opposed to the HOG-based method presented in Nguyen et al.'s previous work [202]. Here again, the fusion of both modalities seems to achieve better results, and CNNs perform better than conventional learning methods on gender recognition using visible-light and thermal images.

In 2019, Baek et al. [91] have experimented with a more complex system, combining visible light images and IR images. They have used two different databases, the first one being the SYSU-MM01 database [205], containing visible light images and NIR images, the second one being a custom-made database (DBGender-DB2) containing visible light images and thermal images (more specifically LWIR images, with infrared wavelength in the 8-12µm range). The overall architecture of the system is shown in Figure 4.52.

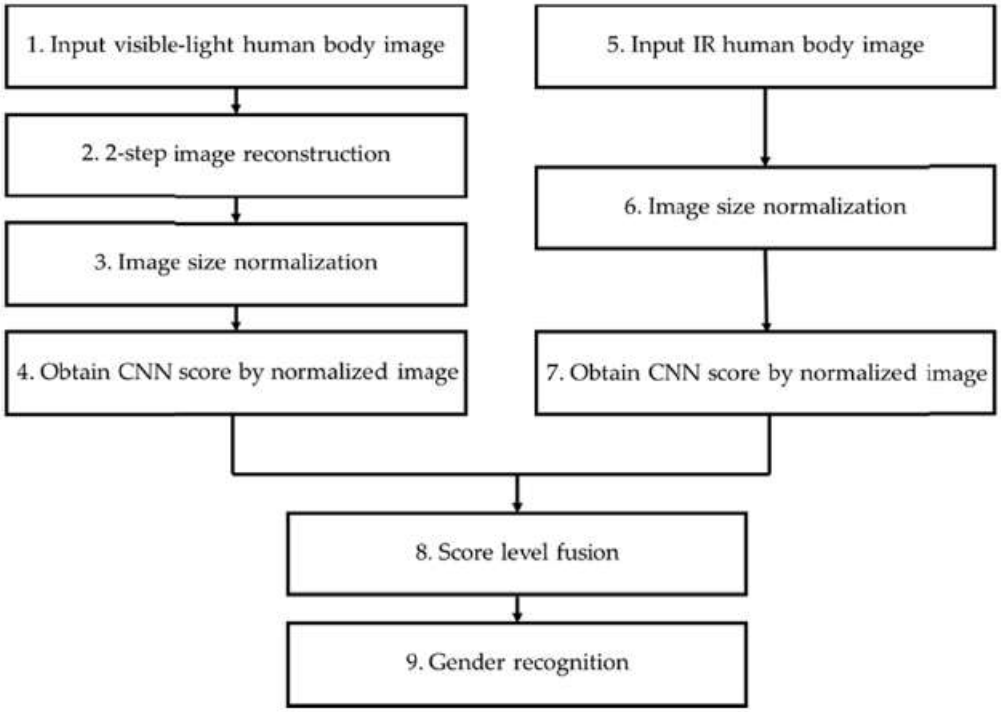

Figure 4.52 : CNN architecture proposed by Baek et al. [91]. © 2019 IEEE.

The main difference between this work and previous work, and what makes it more complex is the fact that the authors have used a two-step reconstruction process to pre-process the images and remove optical blur, motion blur, noise, and to increase the resolution of the images. The first step focuses on denoising the images using an IRCNN, and the second step reconstructs a higher resolution image using a very deep convolutional network (VDSR). Even though this approach could be useful to improve performance in this thesis, the main goal remains to limit pre-processing as much as possible in order to move towards a real-time system. Some example of reconstructed images are shown in Figure 4.53

Once the images are reconstructed, they are fed to a ResNet-101, where the images in input have a resolution of 197x447 pixels, which is much higher than images used in previous work. The authors have found that using the original 224x224, square resolution leads to a

Figure 4.53 : Examples of images obtained through super resolution reconstruction [91]: in each case: image before reconstruction (left), images after reconstruction (right). © 2019 IEEE.

high loss of information when it comes to various body parts ratios and body shape. The ResNet architecture is shown in Figure 4.54.

The SYSU-MM01 database contains 287,628 visible-light images and 15,792 NIR images, a sample of which are shown in Figure 4.55. There are 245 males and 245 females. Using image augmentation (horizontal flipping, image shifting and cropping), the authors have brought the numbers of images for each category from 7,724 and 7,771 to 227,944 and 229,030 respectively. Because of the big difference in number of RGB and NIR images, the authors have discarded easy cases of high resolution close-up RGB images from the dataset until they reached an equal number of both RGB and NIR images.

Some interesting findings have been reported in terms of image reconstruction. Using only visible-light images, and using both IRCNN and VDSR for image denoising and image super resolution reconstruction, the authors have achieved an accuracy of 86.37% for gender recognition. The best results were obtained using image reconstruction. For NIR images however, the best results were obtained when the original image was used, without any reconstruction, with an even higher accuracy than for visible-light images (90.98%). Using

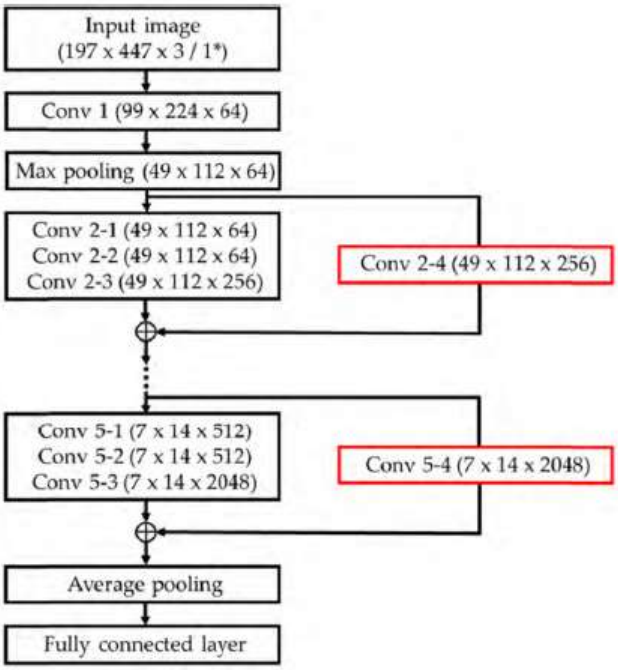

Figure 4.54 : ResNet-101 architecture used for gender recognition by Baek et al. [91]. © 2019 IEEE.

score fusion using a weighted sum of the individual classification score using images from both modalities, an accuracy of 94.73% was reached on the SYSU-MM01 dataset. For this dataset, two-fold cross validation was used, with a ResNet-101 trained for 20 epochs using SGD and a momentum of 0.9, a weight decay of 0.0001, and a learning rate of 0.1.

Since the SYSU-MM01 dataset is made up of images captured both indoors and outdoors, it contains close-up images of people, which make gender recognition easier, especially when the face, hair and clothing are clearly visible. The authors have therefore built a more challenging dataset, the DBGender-DB2 dataset. Low resolution images are collected outdoors only, using one visible-light camera and one thermal camera. Some samples are shown in Figure 4.56. This dataset includes 412 people, 4120 visible-light images and 4120 thermal images. In this case, five-fold cross validation was used with 60 epochs and identical

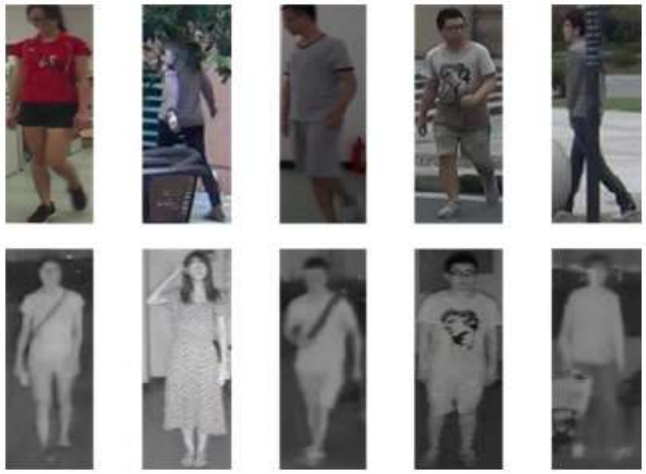

Figure 4.55 : Samples of the SYSU-MM01 dataset used by Baek et al. [91]. Visible-light images (top), NIR images (bottom). © 2019 IEEE.

parameters as used for the previous dataset. On this dataset, the accuracies obtained using visible-light images only, thermal images only, and a combination of both were 85.38%, 86.13% and 89.02% respectively. And identical behavior was observed with original thermal images performing better than reconstructed ones, and a fusion of both modalities leading to a higher accuracy, even though the nature of the dataset has led to a lower overall accuracy.

The authors have also tested the processing time of their method in a desktop environment (Intel i7-7700 CPU, 24GB RAM, and a NVIDIA GeForce GTX 1070Ti with 8GB of VRAM), and have found that the average processing time for an image was 31ms, leading the system to be able to support a framerate of 32fps. In order to evaluate the potential for embedding, they have also tested their method on a Jetson TX2 embedded system, commonly used for deep learning applications in autonomous vehicles. It is equipped with a GPU from the NVIDIA Pascal family and 8GB of memory shared between the CPU and GPU. They have found that the processing time on this device is around 99.26ms per image, leading to a

Figure 4.56 : Samples of the custom DBGender-DB2 dataset used by Baek et al. [91]. Visible-light images (top), thermal images (bottom). © 2019 IEEE.

system that can handle a frame rate of 10 images per second at most. This demonstrates that this approach could be used in real-time, on an embedded, albeit specialized device.

Finally, an analysis of the extracted feature maps in the CNN has taught the authors that the CNN working on visible light images generally emphasizes hairstyles more, whereas the CNN working on IR images shows a stronger emphasis on body-related features. This explains how combining both visible-light and IR images can lead to an improvement in result: the visible-light network takes advantage of visible characteristics that can discriminate between male and female, such as hairstyle or clothing, whereas the IR network relies on body shape and body-based features. This also explains why the visible-light model performs better after image reconstruction, as it allows features such as hairstyle to be more distinct from the background and more noticeable for the CNN. This allows for a robust system that can take advantage of both modalities.

In more recent work, Baek et al. [206] have sought to remove the need for IR images completely, because of the cost of some IR cameras, making it challenging to build recording systems including a thermal camera. They have used an attention-guided GAN in order to synthetically generate equivalent IR images based on visible-light images. Both visible and synthetic IR images are then used with a score fusion method, and a custom loss function used for the GAN in order to perform gender recognition. Even though it is interesting, this paper is not investigated in details, as it goes beyond the scope of this thesis because of the removal of raw thermal images from the equation. It is however interesting to point out that an accuracy of 87.05% was achieved on the SYSU-MM01 dataset, and 90.95% on the RegDB dataset using this method, demonstrating its efficiency. The accuracy on the SYSU-MM01 dataset is lower than the previous method mentioned using actual IR images and two CNNs (87.05% vs 94.73%). This highlights the potential of raw IR images in the field of gender recognition and more work in the field will surely follow.

4.3.3 CONCLUSION ON GAIT AND GENDER RECOGNITION USING THERMAL IMAGES

This concludes the section exploring gait and gender recognition using thermal images. This specific niche is still nascent, as illustrated by the lower number of papers specifically using thermal images for gait recognition purposes compared to approaches using visible-light images only. Even though it was less gradual, there has been a trend towards the use of deep learning methods in recent years as well in this field. The emphasis has recently been the combination of both visible-light and IR images in order to take advantage of the texture and colors of the former and the more robust and invariant aspects of the latter.

Very promising results have been obtained and more research seems geared towards the investigation of embedded, real-time implementations. Using IR images comes with its own

set of challenges, such as the loss of texture and color information, as well as noise coming from the environment and the background, and a bigger impact of the weather on the collected data. It also comes with benefits, such as the ability of thermal cameras to operate during the night and in low-light environment, and the increase in personal privacy coming from the use of thermal images alone.

4.4 CHAPTER CONCLUSION

In this chapter, literature addressing the wider problem of profile recognition, more specifically age and gender recognition, has been reviewed. The most common datasets have been presented, and the evolution from static, handcrafted methods towards deep learning methods has been observed as well. This chapter was concluded with the exploration of the nascent field of thermal-based gait recognition, as well as the use of both visible-light and infrared images to perform gender recognition.

Part IV

Contributions

CHAPTER V

BUILDING A CUSTOM GAIT DATASET IN THE CITY

The first step towards performing real-time embedded profile recognition in the city using thermal images is to collect a dataset of various types of profiles of people in the city. In this chapter, previous results obtained using a first generation prototype to collect data in the city are quickly presented before diving into the lessons learned from this first experiment, and how they have guided the fabrication of the second generation of prototypes and the design of the second set of experiments. The new prototypes are then presented, along with the sensors used, the experimental design, the lessons learned from the experiments, the collected dataset, and the pre-processing carried out in order to make this dataset usable for deep learning purposes. In following chapters, deep learning methods are explored, implemented, algorithms are tuned, choices are justified, and results are analyzed and dissected in order to gain a better understanding of the problem, the challenges to face, and future avenues to explore. The embedded implementation of these algorithms are tackled later in the thesis.

5.1 LESSONS LEARNED FROM RELATED WORK: A SUMMARY

Before diving into the work and contributions of this thesis, it seems important to give a quick summary of what has been learned in the previous part. This is useful for readers who do not have time to read the detailed literature review and who are looking for a condensed version of the lessons learned in order to understand the choices made in the rest of the thesis.

The literature review has covered the various fields related to the thesis, namely: smart cities, deep learning, profile recognition, thermal vision, and embedded implementation. On top of that, profile recognition itself is a sub-field of gait recognition, but one cannot talk

about estimating people's age and gender without first understanding the ins and out of gait recognition. This is why the related work part of this thesis gradually moves from basic gait recognition knowledge towards deep learning-based gait recognition, to profile recognition, to approaches using thermal images, with a few mentions of embedded implementations.

As highlighted in the previous part, the main trend observed in all gait recognition related work, and from a larger perspective, in science in general, has been the move towards deep learning algorithms. Their ability to model any function, separate classes that cannot be linearly separated by conventional machine learning algorithms, and their flexibility has made them the *de facto* standard for complex learning tasks.

In terms of gait recognition in general, the main challenges have been dealing with gait covariates, such as clothing variations, accessories, and view angles. Simple cases with little variation in the walking conditions have led to good results, however, something as simple as carrying a backpack, wearing accessories, or walking with different shoes have been shown to negatively affect gait recognition systems.

For age recognition, the general consensus is that the further apart age groups are, the easier it is to classify someone as belonging one group or the other. Three classes seem specifically distinct, children between the age of 0 and 15, adults, and elders above the age of 60 (or 65 in some cases). However, when more accurate age regression has to be performed, things can get more complicated as gait covariates and people's specific way of walking, as well as their physical health, and various demographic factors can account for more variability in gait than just their age. In those cases, combining a first classification step followed by an age regression has seemed to prove efficient, as having several specialized models for different, small age ranges seems to work better. Lastly, age and gender have been found to be correlated: knowing someone's gender helps improve age estimation results.

For gender recognition, view angle seems to be a more important factor than for age recognition or gait recognition in general. This is due to several factors: seeing someone's face and their hairstyle can positively affect classification results when it comes to gender recognition. Men have shown to have a higher upper body lateral sway than women, which is better captured from the front, back and oblique angles, as opposed to the side angles that is generally the gold standard when it comes to gait recognition, because of the dynamic gait characteristics it captures. The overall size of the body comes into play as well for gender recognition.

Finally, gait and gender recognition using thermal images have been explored. It seems like there is a gap between early, conventional approaches extracting static features and using machine learning algorithms, and later approaches exploring complex fusion methods of deep learning algorithms and generating synthetic thermal images to perform gender recognition. Very little work has been carried out using thermal images only and deep learning for general gait-based profile recognition. Thermal images seem to have the advantage of being more robust, as they do not capture the color and texture of people's clothing, allowing algorithms to focus on body shape and body features rather than picking up on clothing peculiarities that could be misleading. However, temperature can affect the images a lot, and make background subtraction difficult. Moreover, people wearing insulating clothes are less visible to thermal cameras, which can be confusing for the learning algorithms, and thermal images capture less texture and details in the image as they are reconstructed from heat, as opposed to near infrared images captured by projecting infrared light on the scene, and standard RGB images that capture even more details. This absence of texture and color also limits the use of face and hairstyle to estimate people's gender, which is why the combination of both RGB and IR images seems so promising. For this thesis however, we focus on the use of thermal images alone.

This review of the field is also meant to provide a solid starting point for future work in this field and future theses on follow-up projects. By taking a wide approach and exploring different learning techniques and promising avenues, more efficient choices can be made for the continuation of this work.

5.2 FIRST GENERATION PROTOTYPE

Going back to the very beginning of the thesis, the initial idea was to extend activity recognition from smart homes to smart cities. However, as described in the introduction, this has quickly led to the realization that some changes needed to be made. Wearables were not considered a promising avenue, as the goal of the thesis was to extend the logic of smart homes which is to make the environment smart, and not to attach sensors to the inhabitant. Besides, wearables can be intrusive, and they reduce the pool of people benefiting from the technology to the ones wearing these devices. This means that costly devices have to be produced and distributed, and that each participant has to be known and intently use the device.

For these reasons, wearable approaches have been quickly discarded. Instead, at this stage of the thesis, the question to answer was: how to perform activity recognition in the city using environmental sensors? The nature of the problem came down to exploring the type of data that could be collected using environmental sensors and to find sensors that could collect data allowing activity recognition in the city.

5.2.1 BOARD AND SENSORS

The choice of the board to attach these sensors is very important as it defines the type of sensors that can be used, and the overall architecture of the system. For the first prototype, the approach has been to use low-power boards, as the end goal of the project is to perform

real-time classification in the city. Based on the available boards in the lab, the Arduino Due seemed to be the best choice, as it was the most powerful low-power Arduino board available on the market (Figure 5.1). Arduino boards have been widely used in IoT applications [207], and the Arduino ecosystem includes a lot of resources and many specialized sensors and modules making development easy.

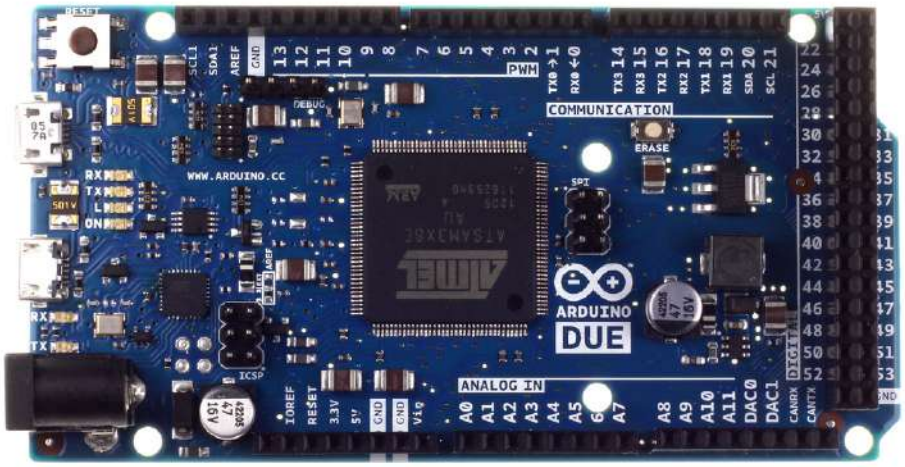

Figure 5.1 : Arduino Due board [208]. Creative Commons Image.

In terms of sensors, the initial goal was to capture as much data as possible from different modalities while still being able to run all the sensors at the same time and store the data without too much loss. The Arduino Due being equipped with a single core CPU means that everything has to be scheduled sequentially, and when the Arduino is storing data, it cannot capture new data at the same time. The best avenue seemed to be the use of image and sound.

However, on such a low power board, the use of a standard camera would have been challenging. Moreover, we keep in mind that the system should be as anonymous as possible in its data collection in order to be widely accepted and adopted.A good proxy for image was

Figure 5.2 : Sensors used in the first prototype: Grid-EYE infrared array (top left), TFMini Micro LiDAR module (bottom left), microphone with OPA334 amplifier (right) [209]. Creative Commons Image.

therefore to use an infrared array sensor, the Grid-EYE sensor. In terms of sound, a simple microphone with an amplifier (OPA334) was used. The possibility of using a piezoelectric sensor was examined in order to convert vibrations to an electrical current, the idea being that vibrations from footsteps could be recorded and patterns could be found. However, the general noise and vibration pollution in the city, as well as the difficulty of including these sensors in a simple prototype that could be invisible to the pedestrians has led this avenue to be discarded. A LiDAR sensor was added in order to measure the distance between the boards and any pedestrian passing by, here, the intuition is that the LiDAR could give an indication of the presence of a pedestrian, as well as some distance information that could be coupled with the Grid-EYE data in order to extract additional knowledge. The sensors used in this prototype are shown in Figure 5.2.

The Grid-EYE sensor captures an 8x8 infrared array, where each cell in the matrix contains a temperature value. This is the equivalent of an extremely low resolution thermal camera, where variations in temperature from cell to cell can help detect the presence, position, and movement of a pedestrian. An illustration of this process is shown in Figure 5.3.

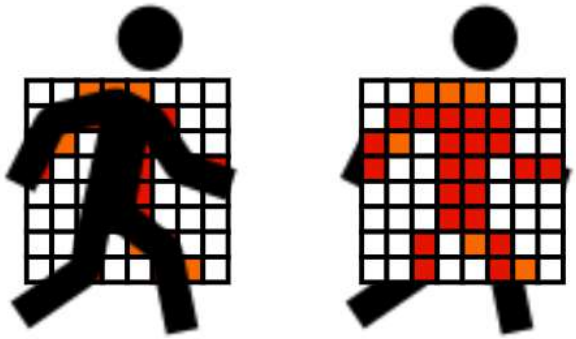

Figure 5.3 : Grid-EYE infrared array activation over a silhouette. © Rani Baghezza, 2022.

5.2.2 PROTOTYPE AND EXPERIMENT

The prototype was put together using a breadboard in order to avoid soldering as much as possible. This allowed more flexibility to replace the sensors and to build the prototype. In order to power the board, a 5V USB power bank with a 3000mAh capacity was used. A rectangular window was cut into a hard plastic box to allow the sensors to see the outside world. A hole was pierced on top in order to access the switch that turns the battery on and powers up the system. Pictures of the prototype can be found in Figure 5.4.

Since the prototype is based on an Arduino board, C was used, and a system of loop was used to collect data from each sensor at a predefined frequency, storing it in the Arduino's memory before regularly storing that data away on the SD card. The optimal polling frequency for the sensors, as well as the optimal buffer size before storing the data was empirically found through testing, and described in more details in the associated publication [53]. The data is stored in text format for all sensors: for the Grid-EYE, 64 numerical values are stored at a frequency of 10Hz, for the LiDAR, a distance value is stored at the same frequency, and for the microphone, an amplitude is stored as a single value, at a frequency of 100Hz. The

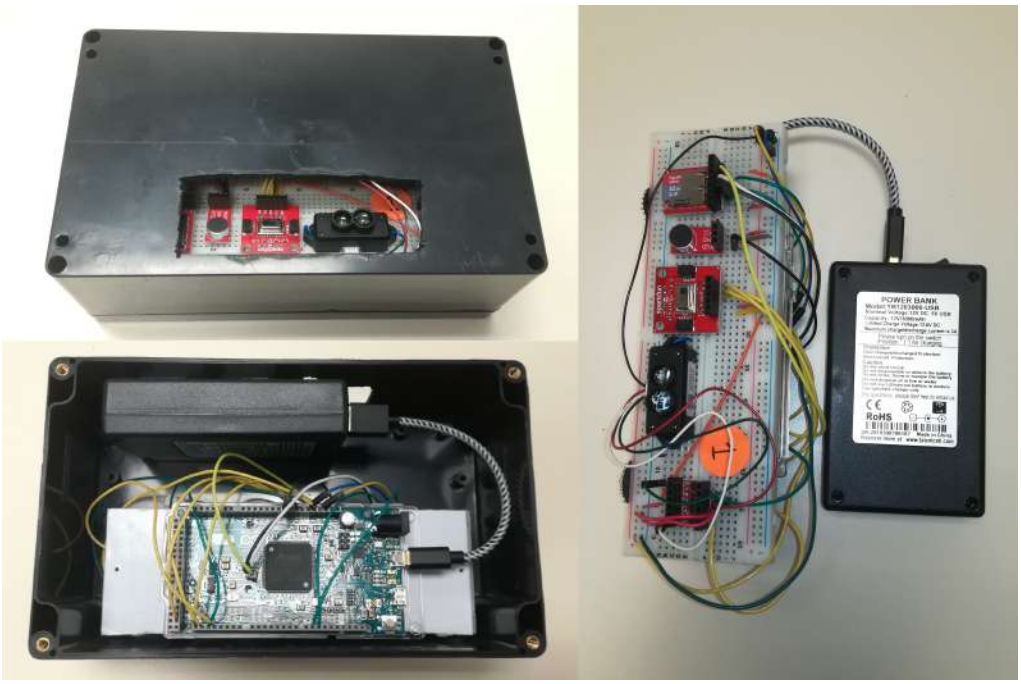

Figure 5.4 : Arduino Due prototype in its handmade case (top left), view from the back of the prototype (bottom), view of the system outside of its case (right). © Rani Baghezza, 2022.

frequencies are limited by the capabilities of the board, as well as the need to store all data while the experiment is going, with minimal data loss.

For the experiment, 4 of these nodes were set on a ledge around elbow/shoulder height in the centre of the city of Saguenay. Several nodes were used as one of the avenues we wanted to explore was the use of distributed machine learning. However, given the number of constraints and the complexity of the problem, adding distributed machine learning on top of the embedded implementation of a profile recognition algorithm using an unconventional type of data (thermal data), the distributed imperative was dropped to focus on getting better results, and solving the initial problem first, which is profile recognition in general, and activity recognition for this first experiment.

The observer sits on a bench where he can see people passing by, and uses an Android application to record the profile of each pedestrian, recording the timestamp of the moment that person walked in front of a specific node. This process is described further in experiments for the second prototype, and the application is presented in the next section. At the end of the experiment, the data of the sensors is matched with the observations of the node by using an initial synchronization step, and creating a 2-second window around each expected observation time in the sensor data. From this window, various features were extracted, such as the lowest, highest and mean value for each sensor.

5.2.3 RESULTS

The goal of this first experiment was to have a starting point from which improvements could be made, and to estimate the ability of a low-power node and low-resolution sensors to perform activity recognition in the city. Because of the experimental setup (pedestrians walking along a short portion of pavement), the number of activities that can be recognized is much lower than inside a smart home. Therefore, the most basic activity recognition scheme that was investigated was the presence or the absence of a pedestrian in front of the sensor. The group size was also investigated using a binary approach (one pedestrian vs. two or more pedestrians), as well as the talking activity (pedestrians talking among themselves or on the phone while walking), in order to evaluate the efficiency of the microphones in an urban environment. In terms of multi-class classification, the activities investigated were: walking, talking, riding a bike and moving in a wheelchair. Age estimation was also performed using a multi-class approach with 3 classes: child, adult and elder.

Random 2-second windows were sampled across the dataset at timestamps where pedestrians were not present in front of the sensors in order to create baseline observations to be used for binary activity classification. Once all of the instances were built, 3 models were

compared for both binary and multi-class classification using Weka: Random Forest (RF), Support Vector Machine (SVM) and Multi-Layer Perceptron (MLP). The training parameters can be found in the associated paper [53]. The results for multi-class classification can be found in Table 5.1 and for binary classification in Table 5.2.

Table 5.1 : Accuracy and kappa values for multi-class classification tasks for RF, SVM and MLP

Target	Metric	Model		
		RF	SVM	MLP
Activity	Accuracy (%)	58.65	64.86	**76.58**
	Kappa	0.21	0.40	**0.60**
Age	Accuracy (%)	63.68	**79.09**	75.45
	Kappa	0.26	**0.59**	0.52

Table 5.2 : Accuracy and kappa values for binary classification tasks for RF, SVM and MLP

Target	Metric	Model		
		RF	SVM	MLP
Activity	Accuracy (%)	86.01	**94.17**	81.12
	Kappa	0.72	**0.88**	0.62
Group Size	Accuracy (%)	**59.25**	54.01	55.66
	Kappa	**0.17**	0.08	0.11
Talking vs. All	Accuracy (%)	54.67	57.66	**66.42**
	Kappa	0.09	0.15	**0.33**

The MLP performed the best for multi-class activity classification with a 76.58% accuracy, whereas the SVM seemed to be able to separate age classes better, with a 79.09% accuracy. However, the Kappa values show that the models perform slightly better than a random classifier (0.60 and 0.59 respectively). For binary classification, the SVM was able to detect the presence of a pedestrian with 94.17% accuracy with a high Kappa of 0.88, which is reassuring, as this is the most basic task that could be asked from the system. Ambient noise pollution and temperature variation, as well as cars in the background could make some of the

empty observation look like pedestrians. The group size estimation was bad overall (59.25% with a Kappa of 0.17), which could be due to the fact that the Grid-EYE sensor does not cover a wide enough range to detect several people at the same time, and the 8x8 resolution could simply be too low. The MLP seems to be able to detect some of the instances of people talking (66.42% with a Kappa of 0.33), however, the low Kappa value seems to show that this could simply be chance. The number of people belonging to the talking class was not significant enough to be able to train a solid model on this task as well.

5.2.4 LESSONS LEARNED

The results obtained using the first prototype have shown that there was a lot of room for improvement. It also showed that the research problem had to be reframed, moving away from activity recognition and towards profile recognition, which was then split into different categories as described in the rest of this thesis. Sound did not seem to play an important role, however, the current microphone was simply recording sound amplitudes at a very low rate of 100Hz, which is far from the 44,100Hz of standard sound recording. An 8x8 infrared sensors can only capture so much information, and the vision of the matrix capturing a silhouette shown by the manufacturers in Figure 5.3 seems very hopeful, and is most certainly applicable at an ideal distance in a controlled environment, or to detect movement in general, but it seems clear that this type of sensor cannot be used for more complex profile recognition. The data from the LiDAR turned out to be unusable, as there was no surface close enough to reflect the signal, leading the LiDAR to have a hard time calibrating fast enough to catch a pedestrian walking by.

The position of the observer and of the nodes were also too obvious during this first experiment, which led a few people to stop and enquire about the nodes, adding noise to the experimental data. It was therefore important to find a place where the nodes could be hidden

from the view of pedestrians, and a location for the observer that would be less obvious to reduce the probability of people stopping and asking questions. Visible nodes can also affect the gait of pedestrians, as the simple curiosity of an unknown object will make them move their head, and possibly walk more slowly. Seeing an observer taking notes on a tablet can also make them feel observed and affect the way they walk (conscious gait instead of unconscious gait).

It terms of hardware, it seemed very clear that an upgrade was needed. Without accurate measure of energy consumption, it was still seen that the batteries were more than enough to run the system for much longer. Five LEDs showing the battery charge are apparent on each power bank, and all five LEDs were still on after a 2-hour experiment, showing that the batteries were still above 80% charge, and probably more, as confirmed in the following experiments.

5.3 BUILDING A SECOND GENERATION PROTOTYPE

From the first experiment, we realized that we needed a more powerful board and higher resolution sensors. In this section, we first describe the new prototype, justify the choice of the board and sensors, and briefly explain the inner workings of the software. More details can be found in the associated publication discussing the first results obtained with the second prototype [54]. The observation application is then presented, as well as the observation-data synchronization before diving into the second set of experiments. Finally, the first leg of data processing is described, and the first results using a naive approach with shallow CNNs are presented, as well as the limitations of these results.

5.3.1 BOARD AND SENSORS

For the second prototype, several boards were considered based on multiple criteria. Since the long-term vision would be to perform real-time profile recognition directly on the boards, sufficient processing power is needed. The first criterion is to use a board equipped with a multi-core CPU in order to carry out several tasks at the same time. Indeed, in a smart city, each node would have to record images and possibly sound, temporarily store the raw data, split it and segment it to form distinct instances, process the data and perform profile recognition, and possibly send out the classification results or even raw data in some cases to a distant server. This is without even considering advanced implementations that would require real-time periodic retraining of the algorithm to deal with concept drift over time [22], and distributed learning, which are much longer term goals.

A powerful board is therefore needed, equipped with a multi-core CPU, good storage capabilities (in terms of RAM as well as in static storage) and communication capabilities (Wi-Fi, Bluetooth Low Energy, and possibly 5G). Reviewing available boards at the time of building this prototype and taking into account the constraints mentioned above, the final candidates were the Raspberry Pi 3, the BeagleBone Blue and the ESP32. The Intel Edison was also considered, but it had been discontinued by the time the second prototype was under consideration. The Raspberry Pi 3 is not as powerful as the Raspberry Pi 4, however, the availability of the Pi 3 at the lab made it an easier choice to consider. A summary of the board's capabilities and a comparison of these boards with the Arduino Due used for the first experiment is shown in Table 5.3.

Even though it has less memory than the BeagleBone Blue, the Raspberry Pi 3 seemed to be the best candidate for the second prototype. Its cheap price, the presence of several USB

Table 5.3 : Comparison of the candidate boards considered for the prototype

	Raspberry Pi 3	**BeagleBone Blue**	**ESP32**	**Arduino Due**
CPU	**1.2 GHz - 64bits** **4 cores**	1GHz - 32bits 1 core	240MHz - 32bits 2 cores	84MHz - 32bits 1 core
Mem.	1GB (RAM)	**512MB (RAM)** **4GB (Flash)**	512KB (SRAM)	96KB (SRAM) 512KB (Flash)
Comm.	Wifi 802.11 n Bluetooth 4.1 BLE	Wifi 802.11 b/g/n Bluetooth 4.1 BLE	**Wifi 802.11 b/g/n** **Bluetooth 4.2** **BLE**	WiFi & BLE (with shields)
I/O	**4xUSB 2.0** **2xI2C**	2xUSB 2.0 1xI2C	2xI2C	2xI2C
Power	Micro USB (5V) AC port	Micro USB (5V) Jack (9-18V)	Micro USB (5V)	Micro USB (5V)
Price (USD)	38.75	82.00	**13.64**	40.30

slots, its communication capabilities and the possibility of easily developing software and machine learning applications using Python made it an ideal choice.

It terms of sensors, the idea of using both video and audio was kept, with the main goal being an increase in resolution of the sensors. In order to explore as many avenues have possible, two types of nodes were designed: a central node and two peripheral nodes. The central node is equipped with the highest resolution embedded thermal camera found on DigiKey, a FLIR Lepton 3.5, with a resolution of 160x120 pixels and a frame rate of 9 fps. The peripheral nodes are equipped with a smaller MLX90640 thermal array (24x32). This has two purposes: reducing the cost of the overall prototypes (shown in Table 5.4), and examining what can be done with an infrared sensor of higher resolution than the Grid-EYE but lower than the FLIR Lepton.

Both types of nodes are also equipped with a USB microphone to pick up environmental sound, and a SD card for storage. The same power banks are used to power the system, and the hard plastic cases from the first prototype were used again as well. Capturing data

Table 5.4 : Price breakdown for both types of prototypes

Item	Price	Central	Periph.
Raspberry Pi 3	38.75	✓	✓
FLIR Lepton 3.5	288.00	✓	
MLX90640	59.95		✓
USB Microphone	9.75	✓	✓
Micro USB-USB cable	10.30	✓	✓
Power Bank 3000mAh	39.99	✓	✓
Waterproof Project Box	19.99	✓	✓
Micro SD Card 32Go	9.96	✓	✓
Total price (USD)		**416.74**	**183.54**

from different types of sensors also allows to test the merging of several modalities to see whether the fusion of heterogeneous data could lead to better results than the use of image or sound alone, as explored by Castro et al. [156]. Because of the time imperative of the thesis and the high number of avenues to explore however, multimodal gait recognition was not explored. Another avenue opened up by the use of two different types of cameras is distributed machine learning using the decision from two different nodes. However, this would be complex to implement, as each node should record data about the same pedestrian, make a classification decision, and communicate with other nodes in order to reach an agreement. This is also difficult to test in real-time in the city, as it is difficult to identify that the same person has walked in front of two different nodes, especially without an observer. Despite these difficulties, it was deemed useful to collect as much data as possible, and to see what could be done in the lab, after the experiments. A picture of both types of nodes is shown in Figure 5.5.

In terms of software, all Raspberry Pi boards run on Raspbian GNU/Linux 10, and Python is used for the code. OpenCV [210] is used to capture images with the FLIR Lepton and the Adafruit MLX90640 library [211] was used for the MLX camera. In order to capture

Figure 5.5 : Arduino Due prototype in its handmade case (top left), view from the back of the prototype (bottom), view of the system outside of its case (right). © Rani Baghezza, 2022.

infrared arrays at a high enough rate, the baud rate of the I2C ports of the Raspberry Pi had to be manually increased from the default 400kbits/s to 1Mbits/s. Using this methods has allowed to boost the frame rate from around 3.5 to 8 fps, almost matching the 9 fps of the FLIR Lepton. Python's sound library was used to record sound from the microphone, and Scipy was used to store the files in 60 second chunks of 2 channels, 44.1kHz sound in wav format.

A python script is launched on each board as soon as the Raspberry Pi is powered on, and two threads are initialized: one for image capture, one for sound capture. Each thread continuously captures data and Python's time library is used to keep track of time on each board, and the capture timestamp of each image is used as the name of the image (eq. the infrared array). An image will be named *1597321577359.png* for example (eq.

1597321577359.txt for the MLX infrared array), and a sound slice of 60 seconds will be named *1597287121971_1597287181971.wav*.

5.3.2 OBSERVATION APPLICATION

In order to obtain the class labels associated with the collected data, which is necessary to perform supervised learning, the observer has to manually record the profiles of the pedestrians as they walk by the node. Not only are the recordings necessary to perform deep learning tasks, but the time of each recording is absolutely crucial in order to match the right data to the right observation. This is where the development of a custom-made android application proved necessary. The android application used in the first experiment was adapted to account for the different labels to collect. A screenshot of the application is shown in Figure 5.6.

During the experiment, the observer activates the radio buttons corresponding to the profile of the pedestrian walking in front of a node, and clicks on the corresponding node button at the right time (3 nodes are deployed, two peripherals and a central node). At the end of the experiment, all of the observations and their timestamp (Android timestamp) are recorded and stored into a text file, after clicking the "Finish Experiment" button on the top left. Just under that button is an observation counter that allows the observer to take notes on a separate smartphone during the experiment when comments have to be added about specific observations. The observations are then synchronized with the data in a process that is explained later in the chapter.

As seen on the screenshot, several aspects of the profile are recorded during the experiment. As a pedestrian walks by, their observed gender, group size, activity, age group and mobility status are recorded. Gender and group size are self-explanatory, and they are recorded based on what is seen by the observer. The activity portion covers different methods

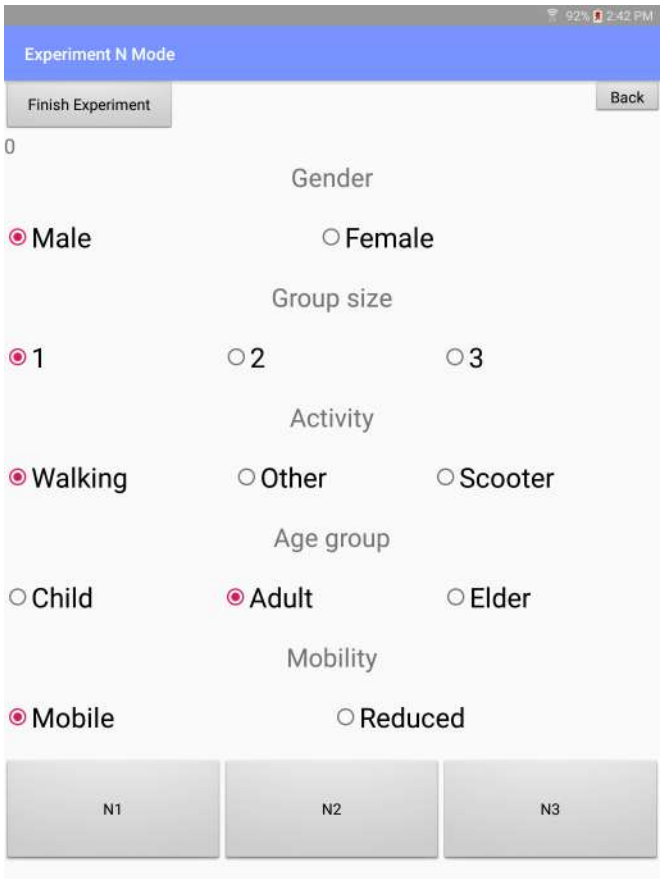

Figure 5.6 : Screenshot of the Android observation application at runtime (Samsung Galaxy Tab S2). © Rani Baghezza, 2022.

of transportation, and it is split between the walking activity, which most people fall into, the scooter activity, specifically made for people (often, elderly) moving in electric wheelchair which are a hybrid between a wheelchair and a vehicle, and the other category for everything that falls outside of this (riding a bike, walking a dog, pushing a stroller, jogging...). In terms of mobility, everyone is considered mobile unless they are in a wheelchair, an electric scooter, using crutches, a cane, or showing a significant limp that makes them walk marginally slower than they otherwise would. The age group largely depends on the estimation of the observer. Generally speaking, anyone looking older than 65 is categorized as an elder, and anyone looking like they are under 15 is categorized as a child.

One of the first observations about these experiments is that it is impossible to completely bypass the observer's biases. For example, some limit cases in terms of age have been difficult to categorize. Someone could be older than 65 but walk with ease, a good posture, and demonstrate strong signs of physical fitness. On the other end of the spectrum, an adult in poor physical shape could have a crooked posture and walk slowly, even though they are less than 40. This also demonstrate the bigger issue that gait alone will never be a foolproof method of estimating someone's age, which is why a lot of approaches have combined information from the face (when it can be captured), as well as hairstyle and accessories.

This issue is also apparent for the observed gender of a pedestrian. In most cases, gender can be determined at first glance, even though there is a difference between observed gender and the gender a pedestrian identifies with, especially considering non-binary variations. For the sake of simplicity, the observer labels pedestrians as being either male or female throughout the experiment. In some cases, however, the observer has to make uncertain assumption on the gender of a pedestrian, especially when unisex clothing is worn, or based on the androgynous look of some of the pedestrians. Since there is no method of obtaining the ground truth for gender, as it would require asking the pedestrian and breaking the no-contact constraint of an observational experiment, we are satisfied with those limitations. A possible improvement on this application would have been to add an *unknown* category for gender in cases where the observer cannot make an assumption with a very high confidence. On the overall scale of the experiments however, such a situation has only occurred a handful of times.

More activities could be added to handle different cases. However, it was important for the application to remain as simple as possible in order to be used efficiently during the experiment, as pedestrian traffic did increase to a point where observations were missed. This is explained further in the next subsection.

5.3.3 SUMMER 2020 EXPERIMENTS

Using the second prototypes, 10 experiments were carried out over the course of a few weeks in the summer of 2020, in Saguenay, Québec. The city has an estimated population of 150.000 inhabitants, 30.500 are elders, making it an ideal city to carry out age recognition experiments [212]. The task of finding an ideal location, the ethics and the unfoldment of the experiment are discussed in this section.

5.3.3.1 FINDING A LOCATION

In order to find a suitable location for the experiment, various locations were scouted first using Google Street View, then by going in the field and seeing whether all of the conditions were met by each location. From the lessons learned after the first experiments, the nodes should be out of sight of the pedestrians. Ideally, they should all be at the same height and be able to capture the entirety of a pedestrian's silhouette, which was an issue in the previous experiment, as the nodes were positioned quite high up on a ledge. The nodes have to be set apart from each other with enough distance for the observer have enough time to click the "N1", "N2" and "N3" buttons at the right time. If the nodes were too close, it would be impossible for the observer to click the buttons quickly enough while ensuring that each button is pressed right as a pedestrian walks in front of the thermal camera, which is necessary for an easy synchronization of the labels and the data after the experiment.

Ideally, we want pedestrians to walk in front of all 3 nodes in an uninterrupted way, which means that a natural obstacle would be ideal, such as a wall. However, performing the experiment in the city means that people can still cross the road and have their profile recorded on one or two nodes only. This is also why the matter of distributed learning and individual pedestrian identification is challenging: there is no guarantee that someone walking in front of

the first node will walk in front of the next two. There is no predetermined walking direction either, and different groups of people can walk in opposite directions at the same time.

A high traffic area is preferable, especially in a small city, in order to collect enough data to train the deep learning models. A very high traffic can become an issue however, as the observer can have trouble picking up all of the observations, and looking at the images from the cameras is often not enough to determine someone's profile after the fact. The final location of the experiments is shown in Figure 5.7. It was deemed the ideal location thanks to the natural barrier provided by the vegetation, which meant that people had to walk passed all three nodes unless they crossed the street. The bench was far enough to be out of sight of most people, and to have a wide peripheral view, allowing to anticipate the profile of people before they entered the perimeter of the system.

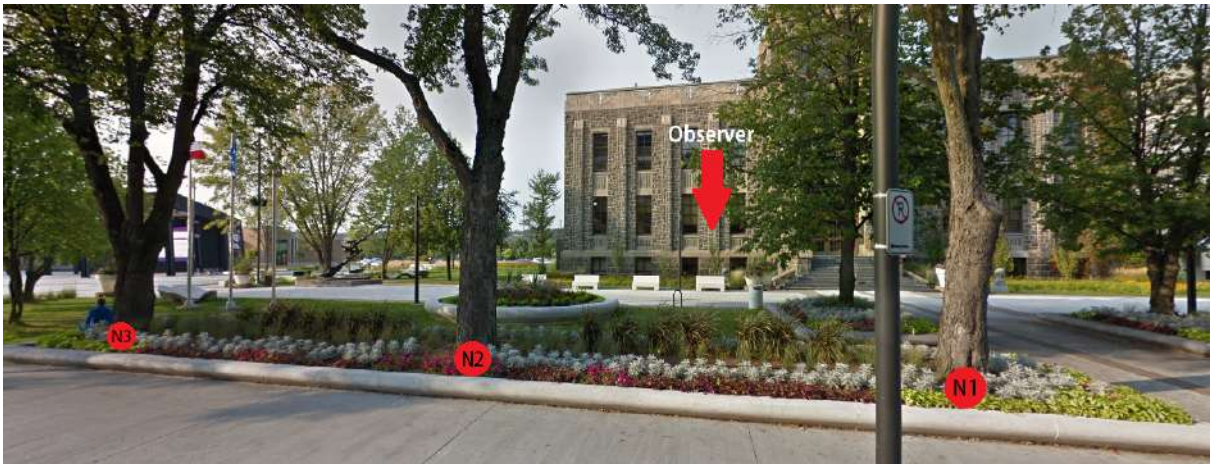

Figure 5.7 : Location of the second set of experiments. The prototype nodes are set in the vegetation, under the cover of trees, and the observer site on a bench to record the profiles of pedestrians passing by. Map Data, © Google, 2022.

5.3.3.2 ETHICS

In terms of ethics, these experiments were performed under Project 2019-227 and approved by UQAC's Research Ethics Committee. The experiment falls under article 2.3 of the policy for research involving human beings and is qualified as an observational study based on the following criteria:

- The experiment does not include any intervention from the researcher or any interaction between the researcher and the subjects.
- The targeted subjects have no specific expectations in terms of privacy, as the experiment is taking place in a public space.
- Shared data does not make it possible to identify particular individuals.

Moreover, the local police station and the city were contacted in order to inform them of the location and nature of the experiments, and to avoid any security incidents. The use of low resolution thermal cameras allows to preserve the privacy of the people involved in the experiment and offers an alternative to a dense deployment of RGB cameras in the city, which would not be welcome by the citizens.

5.3.3.3 EXPERIMENT

At the beginning of the experiment, the nodes are positioned in their respective location, the Android application is launched and the nodes are powered on. In an ideal world, the Raspberry Pi would be connected to the Android tablet using Bluetooth Low Energy (BLE), the devices would be synchronized, and each observation on the tablet would automatically

be matched with the local timestamp of the Raspberry Pi, creating a timestamp mapping that could be used at data processing time to match the data with the labels. Unfortunately, due to time constraints and the necessity to perform the experiments before the winter, a simpler solution was used.

At the beginning of the experiment, a fake observation is recorded on the Android application. For each node, the observer puts their hand to cover the infrared sensors up close and records the fake observation. The idea is to create a distinct outlier at the beginning of the experiment and to record the timestamp of that artificial observation to be used as an anchor to match Android and Raspberry Pi timestamp, in order to locate, isolate and extract subsequent observations. Additionally, we assume that the drift between the Raspberry Pi and Android internal clocks is negligible throughout the experiment. Running both devices for longer periods of time could lead to the additional issue of having to re-synchronize clocks at a regular interval. The experiment then unfolds with pedestrians walking by the nodes, and being recorded on the observation application. The experimental process is described in Figure 5.8.

5.3.3.4 LESSONS LEARNED

This second set of experiments has confirmed some of the lessons learned in the experiment using the first prototype and it has brought additional insights as well. During the first summer 2020 experiment, one of the nodes was too visible, which led people to either glance at it, completely stop, or even pick it up and examine it. The position of the nodes was adapted and the terrain was used to make them less obvious in subsequent experiments. This highlights the fact that the long term vision of a smart city would involve a seamless

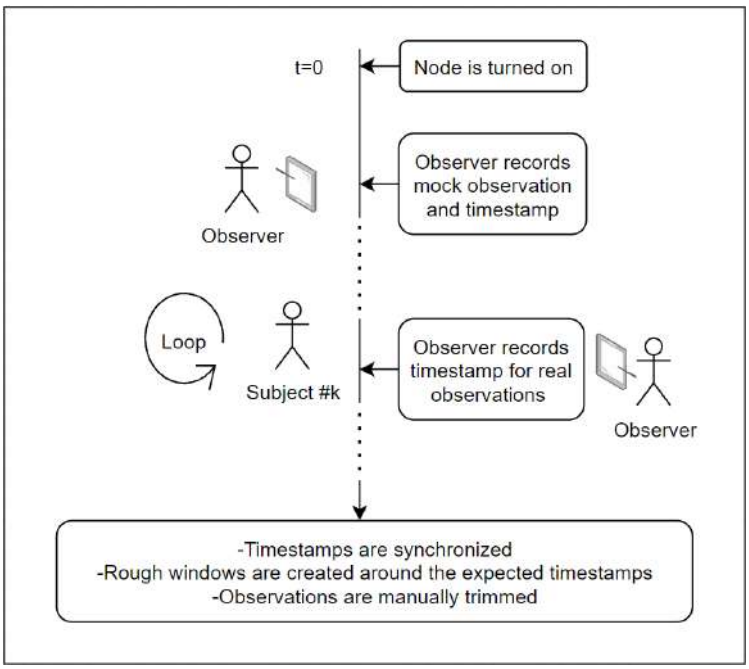

Figure 5.8 : Description of the experimental process, including the capture of the synchronization observation, real observations, and data-label synchronization after the fact. © Rani Baghezza, 2022.

integration of technology into the fabric of the city, and smart solutions will have to be found in order to adapt the positioning of such nodes in existing architecture.

In the real world of urban experiments, pedestrians do not behave in a simple way. In most gait recognition approaches and in most gait recognition databases, each participant walks at a pre-determined angle for a certain distance, and every parameter is controlled. In the city however, pedestrians can walk at different angles and interfere with each other. In some cases, two pedestrians or groups of pedestrians may enter the perimeter of the system at the same time from two different directions (node 1 and node 3), and cross over in the middle before exiting the system after they have walked passed the last node. As an observer, this can be a challenging situation to handle as a choice has to be made of tracking one pedestrian or one group of pedestrians but not the other one. This choice was usually made based on the

class of the pedestrians: since there are less elders than adults in general, if there was a choice between recording an elder or an adult, the elder was chosen for the sake of class balancing. In the same way, persons with reduced mobility had priority. However, the ignored pedestrians do not vanish and they still appear in the raw data, leading to noisy images possibly including several profiles of people that had to be discarded or altered in some cases (removing a few additional frames).

5.3.4 DATA PROCESSING

Once the experiments are over, data has to be processed in order to be usable for training. In order to prioritize the research, the only data used has been the thermal images captured by the FLIR Lepton camera. Using the highest resolution thermal data available is the best avenue to obtain the best possible results. Data from the MLX camera is kept and could be used at a later date. Sound data collected by the microphone is excluded as well for the time being as it requires a lot of pre-processing, has a low probability of being useful by itself, and it would need to be merged with image data.

5.3.4.1 DATA SYNCHRONIZATION

In order to process the data, the anchor point created by the first calibration observation is used. The process is illustrated in Figure 5.9. The timestamp collected by the Android application (t_{A-sync}) is compared to the timestamp of the first frame where the hand of the observer appears to be covering the camera (t_{R-sync}). The respective Android (t_{A-1}) and Raspberry Pi (t_{R-1}) timestamps for the first observation are collected. By taking the difference between t_{R-1} and t_{R-sync}, we know the actual, real-world time difference between the first observation and the synchronization observation. This value is called δt. By adding δt

to t_{A-sync}, we should theoretically get the value of t_{A-1}, corresponding to the moment the observer clicked on the button in the Android application to record the first observation.

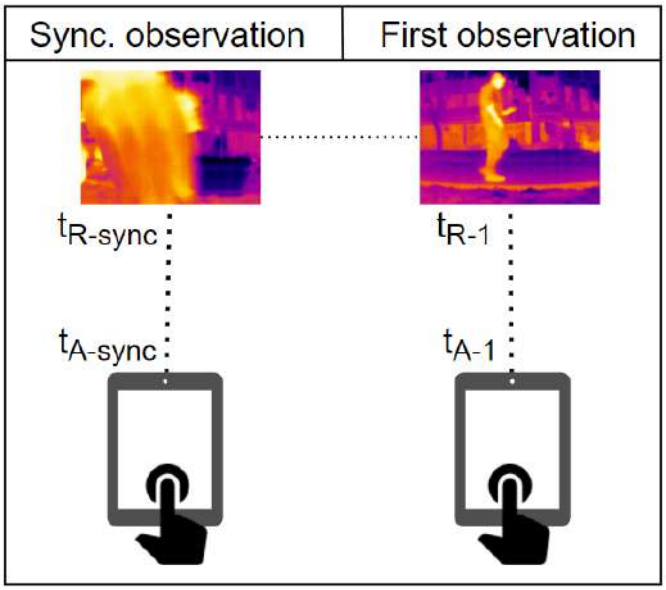

Figure 5.9 : Description of the synchronization process for each experiment. The observer records a first observation while obscuring the sensor with their hand. Both the Android and Raspberry Pi timestamps can be obtained for this artificial observation. Subsequent, real observations are recorded in the same way, and a 10-second window around each expected observation timestamp is created in order to extract the right frames. © Rani Baghezza, 2022.

In the real world however, many factors are at play, leading to an offset between the Android timestamp and the Raspberry Pi timestamp for each observation. First of all, the synchronization observation itself is not that reliable. There is no way of knowing if the observer clicked on the button exactly when they put their hand in front of the camera, as covering the camera is a continuous action, and clicking a button is a discrete step. This is the first source of inaccuracy. After that, each subsequent observation can be recorded with some degree of inaccuracy as well: the observer might click slightly too early or slightly too late compared to the actual moment a pedestrian entered the range of the camera. At this point,

it is impossible to ensure that the button is clicked when the pedestrian is on the first frame, central frame, last frame, any frame in the range, or even before or after they exit the range of the camera. High traffic periods, groups and pedestrians walking in opposite directions at the same time makes this even more challenging.

In order to deal with this uncertainty, a window ε was created around each Android timestamp. All of the frames falling inside that window are then stored in a directory named after the ID of the observation. That ID is nothing more than the date of the experiment concatenated with the observation number (ex: *25_09_20_04* would be the fourth observation of the September 25th 2020 experiment). This window was initially set to 5 seconds, and then extended to 10 seconds to account for the most extreme cases. The end result of running that algorithm on all of the experiments is a set of folders, one for each observation, filled with both empty frames and frames containing the observed pedestrian corresponding to the right observation. The empty frames were then manually removed for each observation, and in certain cases, some observations were removed if they did not seem to match any record, such as in cases with several groups or pedestrians colliding. In some cases, the 10-second window was big enough to include two observations, which is another case that had to be dealt with.

5.3.4.2 DATA SORTING

Once this first step is over, we are left with a master folder which contains one sub-folder for each experiment, themselves containing one sub-folder for each observation, filled with the frames captured for the associated pedestrian. Depending on the walking speed and the distance to the camera, pedestrians can stay shorter or longer in the range of the camera, leading each sequence of frames to have a custom length.

The next step is to organize the data based on the profiles we want to classify. For each experiment, and each observation, we refer to the Android log containing the class information, and organize the observations into 4 different folders: age, gender, mobility and group size. Each one of these classification task is turned into a binary classification problem such that we have (Table 5.5):

Table 5.5 : Summary of the binary classification tasks and classes.

Classification task	Class 1	Class 2
Age	Adult	Elder
Gender	Male	Female
Mobility	Mobile	Reduced mobility
Group size	1	2+

Performing binary classification instead of multi-class classification allows to start with the simplest task, which is important considering the number of challenges that have to be faced. It was decided to exclude children from the dataset altogether as they accounted for less than 5% of the captured observations. Performing experiments later in the afternoon and closer to the end of class would probably have led to a higher proportion of children in the dataset. However, Creating a third age class only containing children with the current data distribution would have led to high class imbalances that would worsen the performance of the system, and grouping up children and adults could make the classes more difficult to separate, as it would increase the intra-class variability within the adult class. In terms of group size, this is not a priority for this thesis, but it was interesting to see whether group size could be estimated, which could be useful for various statistics in smart cities, such as pedestrian traffic density or the type of groups that use the city's infrastructure. One could discover if certain parts of the city are more frequented by single pedestrians or groups of pedestrians, which could then be split between groups of adults, or families made up of adults and children. For a

first approach, groups are simply split between pedestrians walking alone, and groups of two people or more.

Due to the challenging conditions of the experiments in the city, a few outliers have had to be removed. Some examples of these outliers can be found in Figure 5.10. Silhouettes that were too close or very far away from the camera were removed, even though an effort was made not to over-simplify the problem to solve, as the end goal is to perform profile recognition in a real environment. The second factor is the limited amount of data, which makes every instance important in order to maximize the performance of the system. The challenge was therefore to remove clear outliers that would make the problem too complex, at least for a first approach, without shrinking the dataset too much.

Figure 5.10 : Outliers examples from the experiments: overlap between a pedestrian and a cyclist (top left), group of pedestrians blending with a bus in the background (top right), small and still silhouette blending with the background (bottom left), very close silhouette only containing the legs of a pedestrian. © Rani Baghezza, 2022.

Table 5.6 shows the number of instances for each class, in each classification task. Overall, the total amount of instances oscillates between 722 and 726, depending on the

classification task, which is marginally higher than the 144 unique instances collected for the first experiment using the Arduino Due prototype. We can observe a high imbalance in the mobility and group size tasks, which is to be expected, as there are far more mobile people than people with reduced mobility, and that most people walk alone rather than in groups. However, since these tasks are easier to perform than age and gender classification, this is not a very big issue, even though better results could be obtained with more instances in the minority classes. In terms of age and gender, there is still a distinct difference between the majority and minority classes, but it is not as extreme as for mobility and group size.

Table 5.6 : Number of instances for each class in the dataset.

Classification task	Number of instances	
Age	Adult	448
	Elder	277
Gender	Male	501
	Female	225
Mobility	Mobile	679
	Reduced mobility	43
Group size	1	530
	2+	192

5.3.5 FIRST RESULTS

After collecting data, sequencing it, labeling it and pre-processing it, a first approach using simple CNNs was used to perform binary age, gender, mobility, and group size classification. In this section, the process used to get to these results is presented. The limitations of these results, as well as the mistakes made are also presented, as they have allowed to dig deeper into the problem and they have led to the work of the next chapters.

5.3.5.1 DATA ORGANIZATION

In this first approach, a single image is used. Since class labels are recorded for an entire observation which is composed of several images, each image now has to be associated with a class label in order to use CNNs for classification. This is done during the image pre-processing phase before training the models, as illustrated in Figure 5.11.

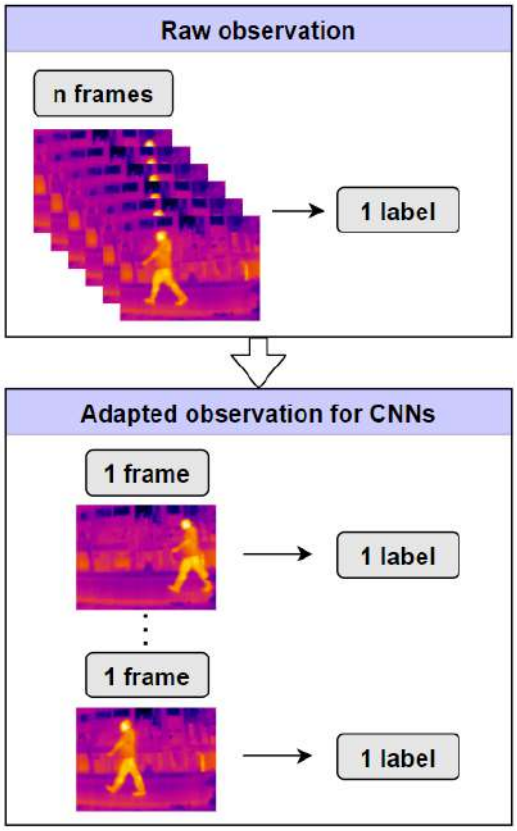

Figure 5.11 : Process of labeling each image inside a sequence with the label of the entire sequence. © Rani Baghezza, 2022.

As observed in the previous section the classes are not balanced, which can cause issues in the classification results. The model could be more biased towards a class, and give results that are more optimistic than truthful, simply based on the ratio of classes in the dataset.

As generating synthetic observations would be too challenging based on the nature and the amount of data available, the simpler choice of removing instances from the majority class was made. Before any learning is performed, the number of observations (in this case, images) in each class is counted for each class, the smallest number n is stored, which represents the number of instances in the minority class. We then randomly sample n instances in the majority class, ending up with a total number of $2n$ instances to be used for training.

5.3.5.2 CLASSIFICATION AND RESULTS

For this thesis, Tensorflow and the Keras API are used to perform all of the deep learning and image augmentation tasks. Since the dataset is limited, image augmentation is very important, as it allows to apply random, minor transformations to the images, artificially generating more data out of the existing data, and to introduce some variation in the dataset, which helps the model generalize better to unseen instances. The ImageDataGenerator class provided in Keras allows to apply transformations such as zoom, width shift (laterally shifting the image), height shift (vertically shifting the image), rotation, shear (rotation and translation), horizontal flip (flipping the image with respect to the horizontal axis), vertical flip (respectively, vertically), brightness shift (random brightness shift within a range). In this case, zoom, width shift and height shift were used, with a parameter equal to 0.2 in all cases.

Various CNN architecture were evaluated by varying the size of convolutional and dense layers from 8 units to 1024 units. The optimization process for deeper CNN architectures is explained further in the next chapter, as it has been carried out much more meticulously, and with a better understanding of the inner workings of CNNs. The architecture that was retained for this step is described in Figure 5.12. Binary classification is performed using binary cross-entropy as a loss function and stochastic gradient descent with a learning rate of

0.01 was used. 5-fold cross-validation is used and the model is trained for 32 epochs. In terms of metrics, the classification accuracy and f-score are presented here. The full results can be found in the associated publication in the IEEE Internet of Things Journal [54].

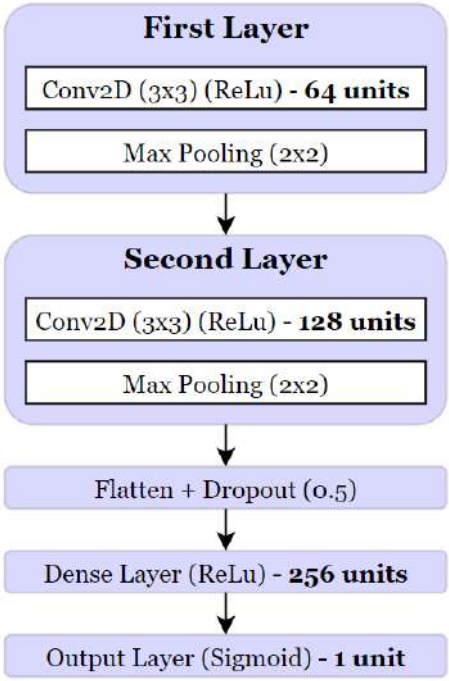

Figure 5.12 : CNN architecture used for the first approach. © Rani Baghezza, 2022.

The results of binary age, gender, mobility and group size classification are presented in Tables 5.7, 5.8, 5.9 and 5.10, respectively. In each case, a number of frames (from 1 to 12) was used for each observation (sequence of image), in order to see how using more data would affect the classification results. One configuration was done using all available frames for each observation, leading to the use of boundary frames, where the pedestrian could have just entered the frame, and only a part of their body is visible. The first results obtained using a shallow CNN are promising, however, they come with limitations which are addressed in the next subsection.

Table 5.7 : Binary age classification accuracy and f-score for different number of frames per observation.

Frames	Accuracy (%)	F1-score	Size
1	67.64	0.68	554
2	70.80	0.70	1108
3	72.41	0.72	1662
4	72.33	0.71	2216
5	72.90	0.71	2765
6	73.50	0.72	3312
7	73.59	0.71	3829
8	73.95	0.72	4352
9	73.07	0.71	4860
10	74.42	0.71	5250
11	74.00	0.71	5687
12	74.16	0.72	6132
All frames	**74.63**	**0.73**	12575

Table 5.8 : Binary gender classification accuracy and f-score.

Frames	Accuracy (%)	F1-score	Size
1	69.78	0.73	452
2	72.00	0.74	904
3	70.79	0.71	1356
4	75.25	0.77	1808
5	75.16	0.77	2255
6	75.19	0.78	2700
7	74.91	0.75	3129
8	76.41	0.79	3568
9	76.73	0.79	3942
10	75.75	0.77	4310
11	76.58	0.79	4653
12	75.38	0.77	5040
All frames	**77.29**	**0.80**	9864

Table 5.9 : Binary mobility classification accuracy and f-score.

Frames	Accuracy (%)	F1-score	Size
1	84.33	0.84	57
2	84.55	0.86	114
3	88.80	0.88	171
4	88.83	0.89	228
5	90.71	0.91	285
6	87.35	0.87	342
7	89.94	0.89	399
8	91.09	0.90	464
9	91.00	0.90	504
10	92.73	0.92	550
11	92.59	0.92	583
12	**93.98**	**0.94**	636
All frames	91.32	**0.92**	1291

Table 5.10 : Binary group size classification accuracy and f-score.

Frames	Accuracy (%)	F1-score	Size
1	79.49	0.78	392
2	80.51	0.79	784
3	82.25	0.81	1176
4	83.40	0.83	1568
5	83.32	0.83	1960
6	84.46	0.85	2346
7	84.67	0.85	2723
8	84.23	0.84	3104
9	84.58	0.85	3438
10	84.27	0.85	3770
11	**85.77**	0.86	4070
12	84.25	0.86	4404
All frames	84.51	**0.86**	9161

5.3.5.3 ANALYSIS

At first glance, it seems clear that mobility classification is the easiest task. Indeed, the highest accuracy reached 93.98%, which is understandable given how distinct the thermal signature of a wheelchair is compared to a pedestrian. This is illustrated in Figure 5.13. This is followed by group size estimation, with an accuracy of 85.77%, gender classification with 77.29% and finally, age classification, which seems to be the most challenging task to perform, with 74.63% accuracy. Examples of dataset samples for age and gender can be found in Figure 5.14. Overall, the accuracy seems to be improved when using more frames per observation, up to a certain point and under certain conditions.

Figure 5.13 : Dataset samples showing instances of each class for the group size and mobility classification tasks: 2-pedestrian group (top left), single pedestrian (top right), person with reduced mobility (bottom left), mobile pedestrian (bottom right). © Rani Baghezza, 2022.

Both age and gender benefit from the use of all frames for each observation, whereas both group size and mobility seem to achieve peak accuracy when 11 and 12 frames are used, respectively. This could be due to the fact that, at this stage of the thesis, observations

Figure 5.14 : Dataset samples showing instances of each class for the age and gender classification tasks: elder (top left), adult (top right), female (bottom left), male (bottom right). © Rani Baghezza, 2022.

were cropped in order to keep as many frames as possible, leading to frames containing less than half of the silhouette to be used in the *"all frames"* configuration. This is why a fixed number of frames was used as well, as it allows to estimate the influence of using a smaller portion of silhouette to perform classification. Even for gender and age recognition, the gap between using 12 frames, and all frames (which could go up to more than 40 frames in some cases) is quite small: 1.91% and 0.47% respectively, for an increase of 4,824 and 6,443 frames respectively, which is an increase of 95.7% and 105.1% in dataset size. In other words, doubling the size of the dataset by using frames capturing portions of silhouettes only leads to a marginal increase in accuracy using a shallow CNN. This is an argument for prioritizing high quality frames containing complete silhouettes, at least in the dimension that can be controlled (width), as nothing can be done if the silhouette is not vertically complete because of the positioning of the camera, as we will see in the next chapter.

5.3.5.4 LIMITATIONS

These results are limited in a few different ways, the first of which is the model used. A shallow CNN was used, which means that more features could have been extracted with the use of a deeper CNN. The choice of the architecture came from the combination of several factors, the first of which was a lack of experience with Tensorflow and CNNs in general, which has led to sub-optimal choices in several different areas, namely: the number of convolution layers, the depth of the overall network, the optimizer used for gradient descent, and the image augmentation parameters. However, it was also justified because of the embedded implementation end goal. Even though deeper CNNs could have been used straight away, the logic behind the use of the shallow CNNs was to see what results could be achieved with a shallower architecture that could give a better guarantee of working in real-time in a limited resource environment. However, as explained in the following chapter, the lack of understanding of the CNN architecture has led to inefficient design choices, which have been corrected in the rest of the thesis.

The dataset itself is a limiting factor that cannot be overlooked. First of all, the number of images is limited, especially when boundary frames are removed. A few thousand images can be enough to train a deep learning network for a simple binary classification task, however, this one is quite complex due to several factors. The environment is not controlled, which means that a lot of noise is present in the dataset in terms of a changing background, cars, buses and motorcycle driving in the back. We have no control over the view angle or the distance of the pedestrians to the camera, and no control over the portion of the silhouette that is captured either. If all cropped silhouettes had to be removed, very little data would be left and the solved problem would essentially be an unrealistic version of the problem that this thesis aims to provide solutions for. In real life, regardless of how optimal the position of the

camera is, we are bound to observe portions of silhouettes if people walk very close to the camera, or very small silhouettes if they walk far away.

Additionally, clothing variation as well as a high intra-class variability makes the problems of age and gender classification difficult to deal with. Indeed, a 20 and a 55 years old pedestrian are both considered adults, and male and female can belong to the same class. Additionally, a physically fit 70 years old could walk in a more youthful and dynamic way than an out of shape 20 years old adult. Combining this with camera positioning variation between the experiments, as well as temperature variation throughout the same experiment, and between different days, it is clear that the dataset and the problem to solve are very challenging. In terms of gender, unisex clothing like jeans, a t-shirt or a coat can greatly confuse the model.

Another factor was discovered later in the thesis when it comes to data organization. Because of the limited data, a 5-fold cross validation approach was used, where the model is trained on 80% of the data, tested on the last 20%, and the average of the test results are presented. However, since CNNs only use a single image, this has led the test dataset to contain unseen frames of known people, making the task easier for the CNN. It means that in some cases, the CNN could see a silhouette that was quite close to a known silhouette from the training stage, as it was the same person at a different stage of the gait cycle. At this point, the CNN could classify these images efficiently due to their overall similarity, but it is unclear how much this has helped the model in the classification task. This was counter-balanced by the fact that all data was used in this experiment, meaning that age and gender classification was also performed on persons walking in groups, as well as people using a wheelchair, which is further explained in the next chapter.

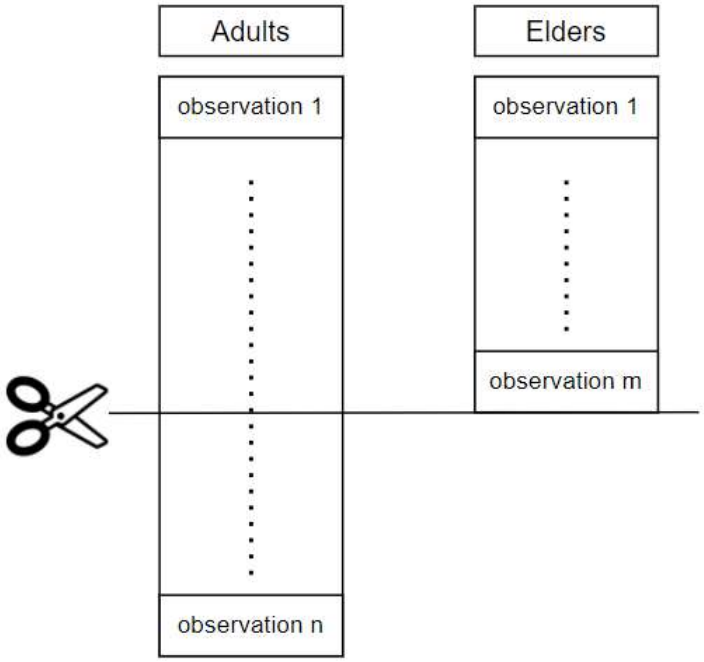

Figure 5.15 : Illustration of the old data balancing technique. Since the observations were not shuffled ahead of time and organized in an alphabetical order, the instances of the last experiments for the majority class (in this case the adult class), never appear in the dataset used for training and testing. This has been corrected in the subsequent approach. © Rani Baghezza, 2022.

Finally, the way class balancing was carried out in this first approach has led some instances to be consistently ignored when building the dataset. As instances are organized alphabetically in the folder, and as a random shuffling was only performed *after* the class balancing, some of the later experiments were ignored for the majority class. This is illustrated in Figure 5.15. Since the temperature and the camera position were unique to each experiment, this led to significant changes in the overall image as well as the background and its color, as seen in Figures 5.13 and 5.14. This had led the CNN to understand that the specific general appearance of an image was never found in one of the two classes. An example can make this clearer. Assume the images of the experiment performed on the last day (September

25th 2020) have a specific shade of purple that is deeper than in the other experiments. Since no shuffling is performed before storing the frames in a vector, and since the vector of the majority class (say, adult), is cut at the length of the minority class (elder), it means that no adult from the last experiment would ever appear in the dataset. If the CNN learned that, it would mean that the simple shade of the image could lead it to understand that this specific observation has to be an elder, because it has never encountered a member of the other class and this shade at the same time. It is difficult to estimate to what extent this has impacted the results of this experiment, as the CNN was quite shallow, there were 10 experiments with temperatures varying all throughout the day, and that the experiments took place around the same time. This has been corrected in the next approaches by randomly shuffling the instances before and after class balancing, in order to ensure an equal probability of appearance of each class, for each experiment.

5.4 CHAPTER CONCLUSION

In this chapter, we have covered the process of building the two versions of the prototypes used for the experiments of this thesis, and presented the lessons learned in each instance. This chapter has covered the span of what happened in the first 30 months of the thesis. Indeed, coming from the context of smart homes and activity recognition using environmental sensors, a lot of path had to be covered in order to shift the perspective enough to be able to not only isolate and formulate the problem in a way more suited to smart cities, but also to build the prototypes themselves and learn the practical and theoretical skills necessary to carry out these experiments.

Since the goal of this first approach was to have a working deep learning model, and to obtain a first set of results to estimate the feasibility of performing profile recognition on thermal images in the city, the model has not been thoroughly optimized. Many more details

of the architecture could have been tuned in order to improve the results, but these have been left for the approaches explored in the following chapters.

Since one of the first avenues of this thesis was to perform real-time distributed activity recognition in the environment, a survey paper was written about the evolution from offline to real-time distributed activity recognition in smart homes [53]. Although this knowledge has not been directly applicable at this point in the thesis, it has been useful to learn how to efficiently cover the literature in a new field, draw parallels, see trends, and foresee future developments. These skills have been very useful when moving from activity recognition to gait recognition. The problem of the thesis has evolved from the very beginning, to the first prototype, to the second prototype and beyond. All of these intermediary steps were necessary in order to thoroughly cover the search space and find an adapted solution.

CHAPTER VI

PROFILE RECOGNITION ON LOW RESOLUTION THERMAL IMAGES USING A SINGLE IMAGE AND A DEEP CNN

The previous chapter has shown some of the limitations of using shallow CNNs to perform profile recognition on thermal images in the city. In this chapter, the process of exploring and optimizing a deep CNN to improve the results is described. Before this is done however, more general lessons about the previous results have to be understood, and the dataset itself has to be altered, which is explained in the first section. The following sections explore the iterative optimization process carried out in order to find the appropriate CNN depth, image augmentation parameters, and learning parameters perform age, gender and mobility recognition using a single image. Group size estimation is excluded as it is a less interesting problem in the scope of this thesis, and further work can be performed at a later date. In order to simplify the problem, the models are optimized on gender recognition, before making slight adjustments to perform age recognition, and mobility is treated as an interesting additional task as it can be applied for accessibility purposes in the city and its nature makes it easy to solve.

6.1 FURTHER LIMITATIONS: REORGANIZING THE DATASET

In the previous approach, all of the available data was used for each classification task. For example, it means that gender and age recognition were performed on groups of 2 or more people, as well as on people in wheelchair. The justification behind that approach was to address the problem in the most straightforward way possible, without pre-processing or pre-sorting, in order to see what could be done in a simple setting where a deep learning algorithm was applied on the raw data. However, this presents a few issues: images of people

in an electric scooter are much blurrier than the images of people walking, which makes it challenging to identify gender with cues such as hairstyle or accessories. In terms of posture, someone sitting in a scooter is not walking, which means that gait recognition simply cannot be performed: there is no gait to recognize. The CNN is also bound to discover the association that elders tend to use scooter more than adults, which gives it a hint for classification, but raises the probability of having false elder positive if an adult is sitting on a scooter as well. We remind that the term scooter is used for electric, motorized wheelchairs, not motorcycles.

In the case of groups, the problem is difficult as well. A lot of groups are made up of both males and females, as well as adults and elders. The algorithm simply cannot classify a mixed group without being wrong. In order to perform age and gender classification on a group, silhouettes would have to be extracted and examined one by one, and an individual classification would have to be performed. This entails addition data sequencing and pre-processing, which would be another problem to solve in itself before even getting to the classification part. Cyclists are also an issue as the images containing them are blurry, and similarly to people in scooters, they are not walking, so there is no gait recognition.

6.1.1 SORTING AND BALANCING

The dataset was therefore adapted to remove groups of 2 pedestrians or more as well as people with reduced mobility or people using any other means of transportation than walking, as illustrated in Figure 6.1. The mobility dataset was not altered, as we ideally want to be able to recognize someone with reduced mobility among any individual in the city.

After that step, the next issue to address is the inefficient class balancing. The simple fix was to shuffle the datasets before and after subset selection based on the number of instances in the minority class. The process is illustrated in Figure 6.2. The number of instances in each

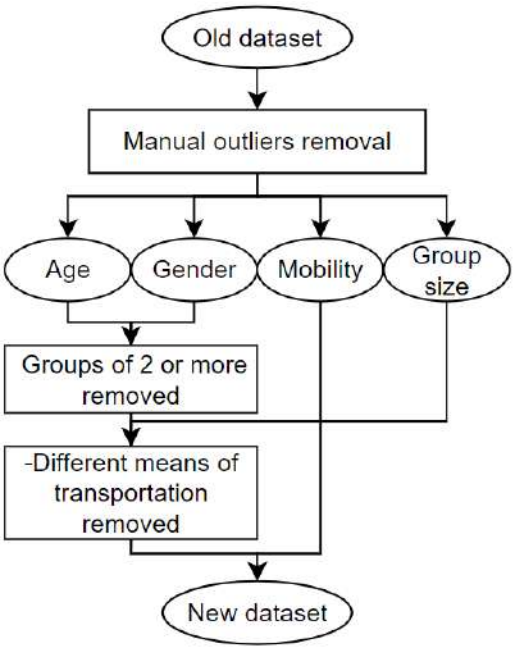

Figure 6.1 : Data sorting process used to remove groups and other means of transportation than walking for age and gender, and removing different means of transportation for group size. The mobility dataset remains unaltered. © Rani Baghezza, 2022.

class after the sorting step are shown in Table 6.1. Since we balance the classes before each run, the most important factor to consider is the reduction of the number of instances in the minority class, as it will be the limiting factor for training. A total of 103 and 67 instances were removed from the minority classes for the age and gender classification tasks, respectively (elder, female), whereas 3 instances were added in the reduced mobility class, as they had been overlooked in the first pass over the dataset. In terms of percentage, this is a reduction of 37.18% and 29.78% in the usable age and gender datasets, respectively. This was also the opportunity to remove a few more outliers, such as very close silhouettes, or observations that flew under the radar. Such observations had missing frames because of some internal bug during image collection, either coming from the FLIR Lepton, or from the Raspberry Pi itself.

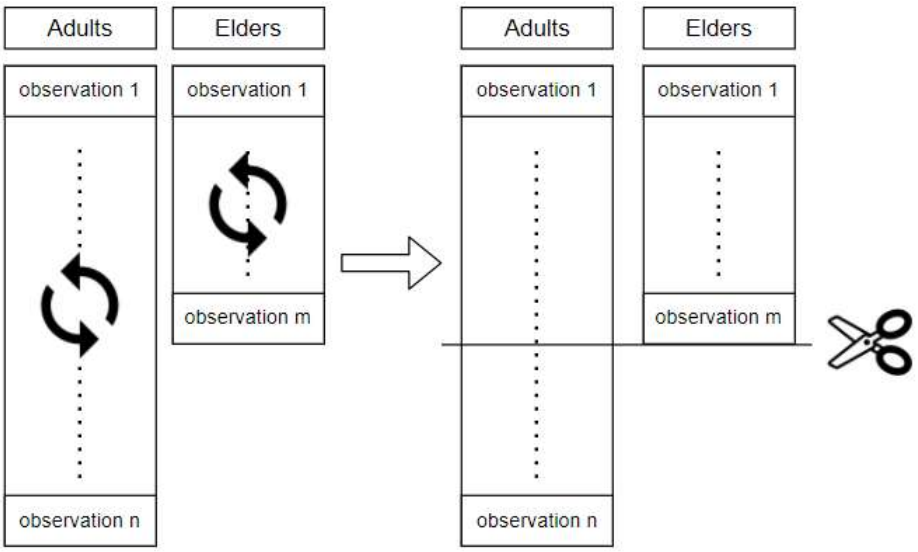

Figure 6.2 : New data balancing including a pre-shuffling of the vectors of both classes. The vectors are shuffled after the balancing step as well. © Rani Baghezza, 2022.

6.1.2 VALIDATION DATASET AND PROBLEM SPACE

In the previous approach, 5-fold cross-validation was used, where the model is trained on 80% of the data, tested on the last 20% and the average of the test results on the 5 folds are presented as being the performance of the model. As illustrated in the previous chapter however, the nature of the dataset has led the test set to contain images of known people at different stages of their gait. In order to avoid that scenario, 5-fold cross-validation was still used, but a completely separate dataset was created by sampling observations from all experiments. This is carried out in a very simple way: for age recognition, for each class, one out of every five observations is moved to a separate folder to create the validation dataset. In order to have a representative sample, observations from each separate experiment appear both in the training and validation dataset. A few observations were kept in a separate test folder, however, the validation results are mostly presented.

Table 6.1 : Comparison of the number of instances between the old and the new dataset.

Classification task		Number of instances	
	Class	Old	New
Age	Adult	448	**287**
	Elder	277	**174**
Gender	Male	501	**306**
	Female	225	**158**
Mobility	Mobile	673	**658**
	Reduced mobility	43	**46**
Group size	1	530	**453**
	2+	192	**179**

This approach could be debatable, as the textbook way of performing supervised learning, especially when optimization is involved, is to train the model on the training dataset, test it on the validation dataset, tune the parameters until the desired validation accuracy does not improve anymore, and to finally test the model on a separate dataset, never seen before by the human or the model. This method ensures that the performance of the model is truly representative of what the model is capable of doing, and that overly optimistic validation results coming from an over tuned model are not presented as being the true performance of the model. However, in this case, we are dealing with a very small dataset and a very complex problem to solve.

This is illustrated in Figure 6.3. Examples from the dataset were taken, and organized in 3D space. The frame of reference illustrates the infinite space of the problem of *profile recognition using thermal cameras in the city*. Since we are dealing with a finite dataset, just like for any machine learning problem, we are essentially looking for a model that can generalize in the best possible way to the overall problem. The issue in this context is that the term *profile* itself is vague and encompasses a lot of different aspects of a person. From this, we sub-divide the problem into age and gender recognition as the two main problems to solve. From that point, the data is still subject to a very high variability as we are dealing

with thermal silhouettes in an uncontrolled environment. It can be seen in the figure with the example of very bright, almost white silhouettes on an orange background of very warm days, as well as orange silhouettes on a purple background, all the way to purple silhouettes on a purple background, making them very difficult to tell apart. Additionally buses and cars appear in the background, some people are wearing backpacks, some people are walking with a cane, others are looking at their phone. We are now very far from the well-known gait datasets presented in the Related Work part of the thesis, which are captured against a green screen with optimal lighting and high resolution cameras, and in fact, in a realistic deployment setting. In these datasets, the silhouettes are then bounded, extracted, normalized, turned into binary format, and the GEI is calculated before even getting to the deep learning part.

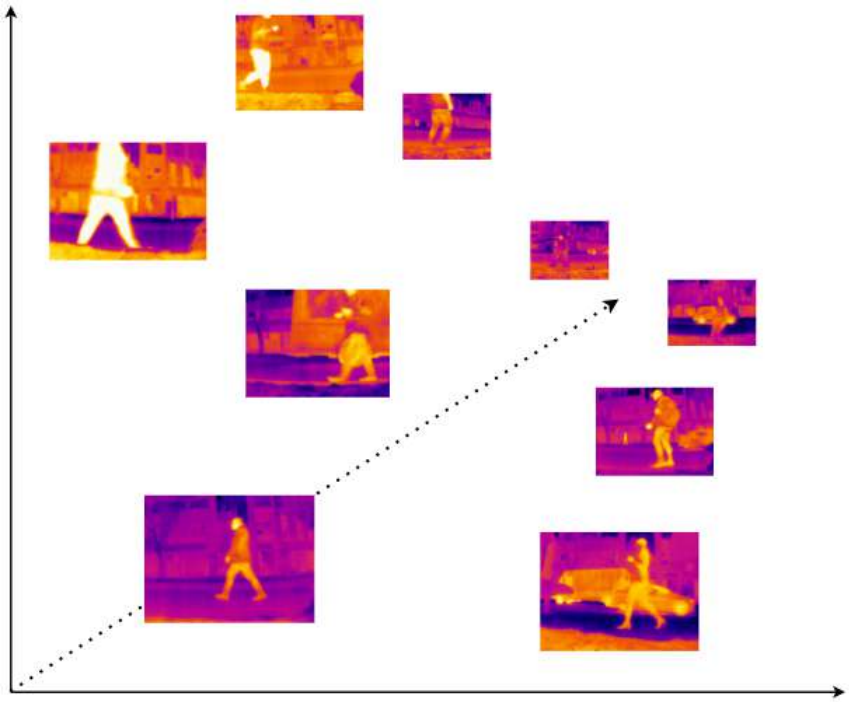

Figure 6.3 : Illustration of the problem space with examples from the dataset. © Rani Baghezza, 2022.

Despite this huge variability, we are working with less than a couple hundred instances at most for each class. This makes it very challenging to split the dataset into a train/validation/test trio of subsets that would capture a similar data distribution: if there are only two instances of a very bright silhouette on a very bright background (two observations), we either train and validate with this data point included, or we train and test on it, but we don't have it in the validation dataset, or the instance is never seen in the training dataset and used for validation and testing. Since the number of gait covariates and variable parameters in the dataset is very high, and the dataset is small, a train/validation/test split is difficult to justify. Besides, each instance that goes towards the test set cannot be used for training and validation, reducing the possible upper bound performance of the model.

That being said, a few instances have been kept apart from both the training and validation dataset in order to have some way of evaluating the model on unseen instances if necessary and for future applications and models, but the results presented are obtained on the validation dataset. Ultimately, this thesis aims to explore profile recognition using thermal images in the city. The dataset then sets the perimeter of what is possible to do and to explore, and smart choices have to be made in order to present the most objective results possible, while still exploiting the dataset as intelligently as possible. The number of images in the training and validation dataset for all classification tasks is shown in Table 6.2 below.

Table 6.2 : Number of training and validation instances in the CNN dataset

Dataset	Task	Nb. Instances
Training	Gender	2524
	Age	2265
	Mobility	204
Validation	Gender	646
	Age	648
	Mobility	106

6.1.3 SEQUENCE TRIMMING AND PARTIAL SILHOUETTES

In the previous approach, the concept of *boundary frames* was mentioned, which is used to describe frames containing incomplete portions of silhouettes. All of the observations have been trimmed in order to solely include full silhouettes, at the exception of the very first and last frame which could contain a silhouette with a very small portion missing, such as the end of an arm swinging, or a part of the foot or the leg. This allowed to preserve as much data as possible while maximizing the surface area of the silhouettes.

However, this close investigation of the dataset has brought up an interesting observation regarding the silhouettes. The majority silhouettes captured by the FLIR camera are cropped vertically to a certain extent due to the experimental setup. Whenever a pedestrian walked too close to the camera, a portion of their head or their shoulders did not appear in the frame. The majority of the body is still apparent, as very far outliers only containing the legs have been removed. However, this adds yet another challenge to this problem, where age and gender recognition have to be performed with a majority of incomplete silhouettes. The repartition between full silhouettes, silhouettes cut at the head and silhouettes cut below the shoulders is shown for age and gender in Tables 6.3 and 6.4. The total values are shown in bold. For the training dataset, 45.81% and 46.41% of the silhouettes are full for age and gender respectively, and 44.07% in both validation datasets. The rest of the dataset is made up of incomplete silhouettes, samples of which can be seen in Figure 6.4.

Removing partial silhouettes would lead the dataset twice as small, which would be less than optimal, especially considering it has already been shrunk when removing groups and people using different modes of transportation. Including these silhouettes also leads to a much more realistic dataset, as it could be expected that partial silhouettes would appear in the dataset, one way or another, depending on the position of the nodes. Most datasets

Table 6.3 : Ratio of complete and partial silhouettes in the age recognition dataset. Full denotes complete silhouettes, head denotes silhouettes that are cropped at the level of the head (forehead, middle of the face, bottom of the face), and shoulders denotes silhouettes that are cropped around or below the shoulders area.

Dataset	Portion	Adult	Elder	Total
Training	Full	27.5%	18.06%	**45.81%**
	Head	14.54%	11.01%	**25.55%**
	Shoulders	17.62%	11.01%	**28.63%**
Validation	Full	28.81%	15.25%	**44.07%**
	Head	15.25%	8.47%	**23.73%**
	Shoulders	22.03%	10.17%	**32.20%**

Table 6.4 : Ratio of complete and partial silhouettes in the gender recognition dataset.

Dataset	Portion	Male	Female	Total
Training	Full	27.00%	19.41%	**46.41%**
	Head	15.61%	9.28%	**24.89%**
	Shoulders	22.36%	6.33%	**28.69%**
Validation	Full	22.03%	22.03%	**44.07%**
	Head	11.86%	11.86%	**23.73%**
	Shoulders	22.03%	10.17%	**32.20%**

presented in the gait recognition literature are simply not realistic for real-world used, as presented in the Related Work part of this thesis. However, a few more recent approaches, such as Nguyen's [204] and Baek's [91] have started investigating more realistic, real-life datasets, which confirms that the trend in gait-based profile recognition is moving towards more realistic applications rather than optimizations of algorithms on lab-like datasets.

6.2 DESIGNING A CNN FROM SCRATCH

As covered in the Related Work section, CNNs have been used to perform age recognition, such as DenseNets [171] and ResNets [175]. In terms of gender recognition, an approach using a CNN and SVM has been presented by Liu et al. [190], and CNN-based approaches

Figure 6.4 : Sample of full and partial silhouettes in the age dataset. From left to right: full, head and shoulders. © Rani Baghezza, 2022.

have been presented by Zhang et al. [175] and Kitchat et al. [191]. The DenseNets and ResNets architecture involve the use of skip connections, which are used to facilitate the backpropagation of weight corrections in the early layers throughout the gradient descent process. In very deep networks, chain multiplications to correct the weights starting from the last layer all the way back to early layers can run into the gradient vanishing issue, where the corrections are not backpropagated far enough, and the changes in weight in the early layers are negligible despite a long training. This leads the early layers of the network to not be trained efficiently, which worsens the performance of the entire network.

However, since we are dealing with low resolution images, and since the end goal is to perform profile recognition in a limited resource environment, it was deemed preferable to explore more straightforward models that did not implement skip connections and to gradually build up from there. This also allows to reduce the search space of the possible CNN architectures to explore in order to be more efficient.

6.2.1 GENERAL ARCHITECTURE

The first chapter of this thesis has covered the basics of Convolutional Neural Networks, and a general overview of a standard CNN architecture is illustrated in Figure 6.5 as a reminder. Overall, the CNN takes an image as an input, and successively extract features maps from

this image by using a convolution operation with a filter, and pooling is used to reduce the dimension of the feature maps as we get deeper into the network. Since feature maps are matrices, a flatten or a global average pooling layer is used to convert these features into a one-dimensional vector that is then fed to a single or a set of dense layers to learn further relationships from the features. In the case of binary classification, a final output layer with a single neuron is used, and the output is either 0 or 1, giving the class prediction. This is just a general illustration, as many different parameters can vary, as well as the organization of the layers.

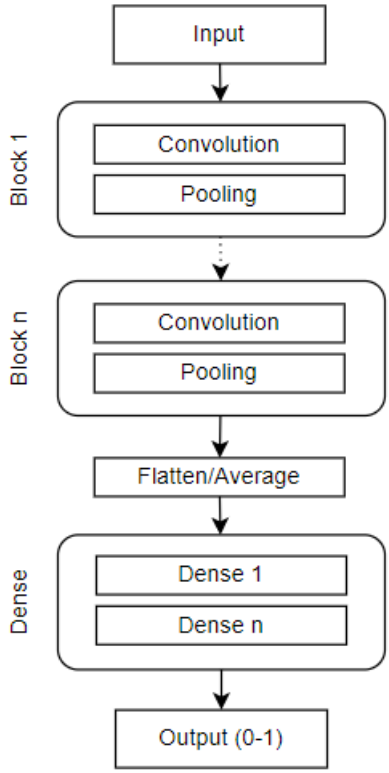

Figure 6.5 : Illustration of a general CNN architecture for binary classification of images. © Rani Baghezza, 2022.

6.2.2 OPTIMIZATION STRATEGY

In order to find the CNN architecture that maximizes classification accuracy on the validation dataset, different parameters have to be taken into account. Since most parameters are inter-dependent, it is very difficult to approach CNN optimization in a linear way: the discovery of a new parameter or a new setting that improves the results may lead to changes in other parameters. A simple example of that is the addition of extra layers, or the increase in number of weights in the model in general. This leads to a bigger model, which generally needs to be trained for more epochs and possibly needs a higher regularization.

A common strategy used to optimize models is to perform a grid search. Just like the name suggests, the idea is to increase or decrease the value of a parameter, run the model (training and validation), log the results, change the parameter again, and compare the results. This gives an idea of where to search next, and which parameter to increase or decrease. However, because of the size of the search space, and the huge number of variables that come into play, a pure grid search is generally not a good approach, especially when limited computing resource is available. Random methods have proven to perform better than grid search as well [213]. The fact that this has also been a learning experience meant that as CNNs and CNN optimization was better understood, the way the search was carried, and the importance of various parameters changed.

The approach used in this thesis is a mix of local grid search when small variations have to be tested on a specific parameter, and bigger, more intuitive configurations based on the understanding of the problem and the growing experience with CNNs as the research has carried on. The first step has been to use a more efficient optimizer to carry out the gradient descent, as well as a more resilient activation function, which is explained in the next subsection. After that, the number and size of the convolutional and dense layers had to be

optimized. A few adaptations were brought to the model, such as replacing the Flatten layer with a Global Average Pooling layer, adding regularization in the dense layers, trying to use momentum in the gradient descent process, replacing the Max Pooling operation with a stride parameter in the convolution. A detour was taken to test a model architecture similar to the model used by Baek et al. [91], as that paper was the closest paper to the research carried out in this thesis. However, it did not seem to yield to better results on this dataset. Throughout the entire optimization process, image augmentation parameters were routinely adjusted, which allowed to gain a general understanding of their effect under different circumstances.

6.2.3 GRADIENT DESCENT

In the previous model, standard Stochastic Gradient Descent (SGD) had been used to minimize the loss function and train the network. However, one of the limitations of SGD is that the step size used for the descent is the same at each step and for each dimension of the gradient. This can lead to cases where a local minima of the loss function is difficult or impossible to reach as the gradient swings back and forth around it. Optimizations such as the Adaptive Gradient Algorithm (AdaGrad) have allowed the step to adapt for each dimension over the course of training by taking into account the history of gradient values at previous time steps. For the rest of this thesis Root Mean Squared Propagataion (RMSProp) is used to perform the gradient descent [214]. Where AdaGrad takes into account all past partial derivatives to adapt the step size as training goes on, RMSProp uses a decaying average over the past partial derivatives, reducing the impact of earlier steps as training goes on. This avoids running into the issue of training slowing down too much that can be encountered with AdaGrad. The equations of RMSProp are presented below:

$$v_t = \rho v_{t-1} + (1-\rho) g_t^2 \tag{6.1}$$

In equation (6.1) v_t is the exponential average of the square of the gradients at time t. It is obtained by multiplying the result at the previous step by a parameter ρ and adding the square of the gradients at the current step g_t^2 multiplied by a factor of $(1-\rho)$. This is done separately for each parameter to update [215].

$$\Delta\omega_t = -\frac{\eta}{\sqrt{v_t+\varepsilon}} g_t \tag{6.2}$$

The step size is then computed in equation (6.2), using the initial learning rate η, the exponential average v_t previously computed, a non-zero parameter ε, and the gradient g_t.

$$\omega_{t+1} = \omega_t + \Delta\omega_t \tag{6.3}$$

Finally, each parameter is adjusted for the following time step as shown in equation (6.3). For the entire process, we use $\rho = 0.9$ and a learning rate of 0.0001. The learning rate has varied in a few instances, but since no better performance could be observed, 0.0001 kept. Besides, adding more epochs with a lower learning rate has been found to be more stable than using less epochs and a higher learning rate, especially considering the small size and the high variability of the dataset.

In the previous approach, a standard Rectified Linear Unit (ReLU) activation function was used in each layer [216]. It is defined as $y = max(0,x)$, and it has been shown to allow faster training and reduce gradient vanishing when compared to sigmoid and tanh. However, since all negative values are mapped to 0, this leads to the dying ReLU problem. Neurons can end up stuck on the negative side and always output 0, effectively rendering a part of the network useless, as it does not propagate any information [217]. One of the easy fixes is to

lower the learning rate. Another fix is to use a Leaky ReLU [218], which works the same way ReLU does for positive values, but instead of mapping negative values to 0, they are multiplied by a constant α. This is illustrated in Figure 6.6. This allows to avoid the dying ReLU problem while speeding up training. The empirical value of $\alpha = 0.3$ was used based on some other applications as well as a small grid search. However, fully optimizing this value considering all of the limitations of the dataset would have been a waste of time, the main point simply being the use of a more efficient activation function.

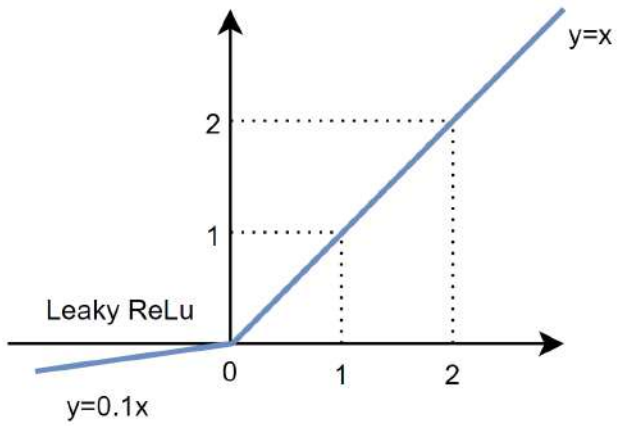

Figure 6.6 : Illustration of the leaky ReLU function. © Rani Baghezza, 2022.

6.2.4 CONVOLUTIONAL AND DENSE LAYERS

The first step has been to optimize the number and the size of the convolutional and dense layers. To keep things simple, a conventional structure is kept where a single convolutional layer is used (Conv2D), followed by a max pooling layer (MaxPooling2D). This structure is repeated until the Flatten layer is reached, and the dense layers are added.

When it comes to the size of the convolutional layers, a standard, linear approach was used where each subsequent layers contains twice the amount of units of the previous layer,

starting with 32 filters in the first layer. The architecture of the CNNs therefore looks like: 32-64-128-256-512-1024, where layers are added and removed throughout the optimization process. Since MaxPooling is used between after each convolutional layer, the dimension of the images is divided by 2 at each step. Starting from a 160x120 resolution, after each pooling operation, the dimension of the feature maps is reduced to 80x60, 40x30, 20x15, 10x7, 5x3 and 2x1. As explained in the first chapter, the convolution step involves sliding a filter over the image, same padding is used to add pixels to the sides of the image in order to maintain the dimension during the convolution part. The pooling operation is the only operation reducing the size of the image at this point.

In terms of filter size, a 3x3 filter was used in the majority of cases, even though some exploration has been done with larger filters without any significant or reliable and reproducible improvements. Using larger filters simply allows to capture the interaction between more pixels into a single value through the convolution operation, but it comes at the cost of a sharp increase in the number of parameters in the CNN, as a 3x3 filter holds 9 weights, and a 5x5 filter holds 25. Besides, through the stacked convolution layers, the CNN discovers relationships between non-neighboring pixels in the image in an iterative manner. Thus, 3x3 filters have largely become the norm, as they strike a good balance between size and efficiency.

CNNs of different depths starting from a single layers to 6 layers have been evaluated with 2 or 4 dense layers. As mentioned before, the optimization process is carried out on the gender recognition task. The training and test protocol is the following: the training dataset contains 2524 images, and the validation dataset contains 646 images. The training dataset is split into 5 parts, 4 of which are used for training, the last being used for evaluation. After each fold, the model is tested on the validation dataset as well. There are a few reasons for this organization. As seen in the previous chapter, images of the same person at a different step of the gait cycle give very optimistic results. Images of unknown people are much more difficult

to classify, especially when accounting for the absence of pre-processing and the variability in silhouette sizes. Knowing the accuracy at each fold for close instances as well as completely different instances (validation dataset) allows to get a better understanding of how the model is working, if it is not generalizing enough, if it is under-trained or over-trained and so on. This also allows to use a random subset of data to train the model each time, and to be able to average the training and validation results over 5 folds, giving a more reliable result, which is necessary when taking into account the variability of the dataset.

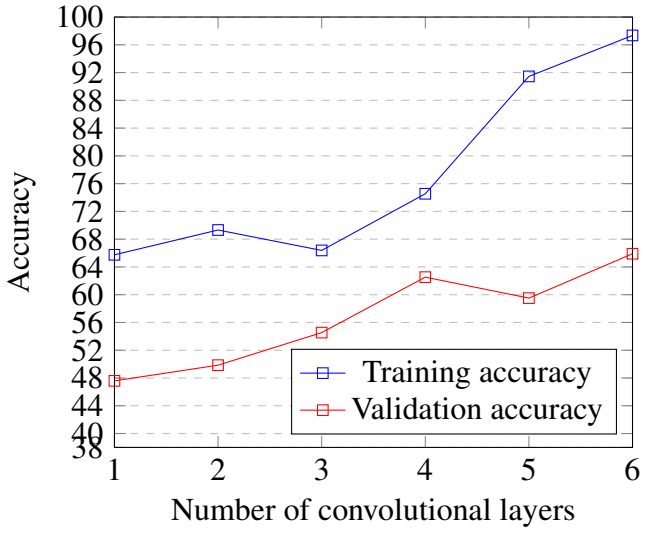

Figure 6.7 : Training and validation accuracy for gender recognition vs. CNN Depth using 2 dense layers (256-128).

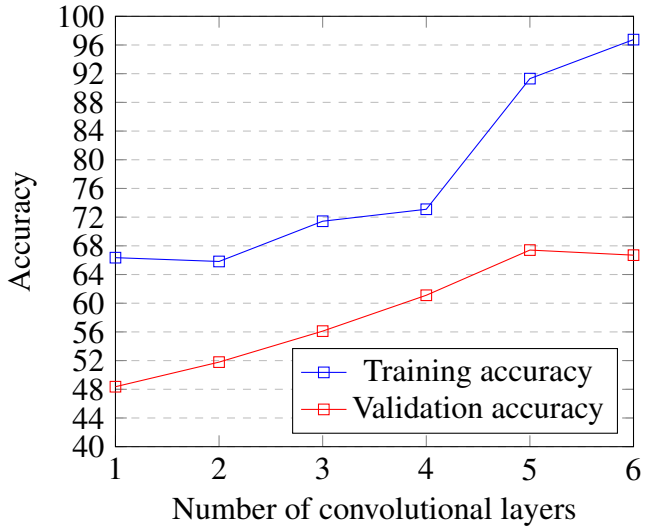

Figure 6.8 : Training and validation accuracy for gender recognition with 4 dense layers (512-256-128-64).

As observed in Figures 6.7 and 6.8, the deeper the network in terms of convolutional layers, the higher the training accuracy. In terms of validation accuracy, the network with 4 dense layers seemed to give a higher performance than the network with 2 dense layers (67.41% vs. 65.9%), however, the peak validation accuracy was reached for 5 convolutional layers and not 6.

Changing the size of the dense layers from 256-128 to 512-256, 512-128, 1024-512 and 1024-256 has always yielded a worse performance than using a 256-128 configuration. Various sizes and depths were experimented with, and ultimately, the best results were obtained with the use of a single dense layer containing 512 units. This seems to confirm the trend that was illustrated in the Related Work part of the thesis, were a 1024 dense layer performed better than a 4096 one, and Smith's conjecture about higher dimensionality dense layers leading to worse performance for gender recognition when studying transfer learning [192]. It was difficult to know whether this would apply to this problem, as the dataset is quite different, and the background has to be taken into account, but the assumption seems to hold.

Based on these initial results, two main CNN architectures were tested: a 5-block deep CNN and a 6-block deep CNN. The logic remains the same: a block is a unit made up of one or several convolutional layers of the same size. Dimension reduction is either performed using Max Pooling between each block or using a convolution with a stride of 2 pixels horizontally and vertically. Both configurations were tested, and the convolution with stride was used in the CNNs, as it has the advantage of being an operation that can be learned through training and it combines two operations in one.

The two architectures tested were variations of stacked convolutional layers of the shape 1x32-1x64-1x128-1x256-1x512 for the 5-block deep CNN, and 1x32-1x64-1x128-1x256-1x512-1x1024 for the 6-block deep CNN. The number of layers was increased in order to see the impact of the depth and number of parameters on the results. Such a linear architecture has proven more efficient than other architectures and it has the upside of being easily scalable for embedded implementations, as the 1 layer per block approach can be used for a compact model, and this number can be brought up to 5 or 6 layers if necessary for implementation on a desktop. The results presented here are the ones of runs performed with optimal dropout, batch normalization, regularization and image augmentation parameters, which are explained in the next subsection. A summary of the results for these architectures is described in Table 6.5.

From the results obtained using grid search with a number of epochs fixed to 32, focusing on 5-block deep and 6-block deep CNNs, it seems like the best overall performing model on the validation dataset is a 5-block deep model with each block containing 3 convolutional layers of increasing size. The general architecture is: 3x32-3x64-3x128-3x256-3x512, with an extra input layer with 32 filters, and a dense layer with 512 units. For the sake of simplicity, we call this model the CNN-5-3, where 5 denotes the number of blocks and 3 denotes the

Table 6.5 : Comparison of different CNN architectures for gender recognition. A linear structure is used where each layer has twice the number of units of the previous one, starting with 32. Each line uses twice as many stacked convolutional layers as the previous one. The models are trained for 32 epochs.

CNN Depth	Nb. Params	L2 Reg.	Train Acc. (%)	Valid Acc. (%)	Best Fold (%)
5	1,853,377	0.0002	92.27	63.75	67.34
	4,999,009	0.0002	96.47	66.22	67.03
	8,144,641	0.0006	95.48	**70.15**	73.22
	11,290,273	0	96.63	69.32	**76.63**
	14,435,905	0.0006	96.39	68.92	73.22
6	6,837,185	0.0002	95.92	68.11	72.60
	19,423,073	0.0006	**96.79**	68.05	70.90

number of stacked convolutional layers per block. The other models are referred to using this same notation in the rest of the thesis.

Because of the high variability and the small size of the dataset, it seems like the effect of regularization is not always linear or predictable, as illustrated with CNN-5-4 performing better without any regularization than CNN-5-5 with a L2 coefficient of 0.0006. It generally seems like more complex models work better with a higher regularization, which theoretically makes sense, with the exception of CNN-5-4 mentioned before. Using a 6th block did improve the results compared to smaller 5-block deep networks, but as soon as more layers were stacked, leading to an explosion in the number of parameters, the performance stopped improving. The relationship between network size and performance on the training and validation dataset is illustrated in Figure 6.9. The cut-off point where additional weights stop improving the performance since to be around 8.000.000 parameters, after which the training accuracy improves, but the validation accuracy drops. However, adding more epochs and more regularization as well as tuning other parameters could lead to an improvement in results for bigger networks. Out of curiosity, a few runs using 40 epochs instead of 32 were done on bigger networks, but they did not yield a higher performance.

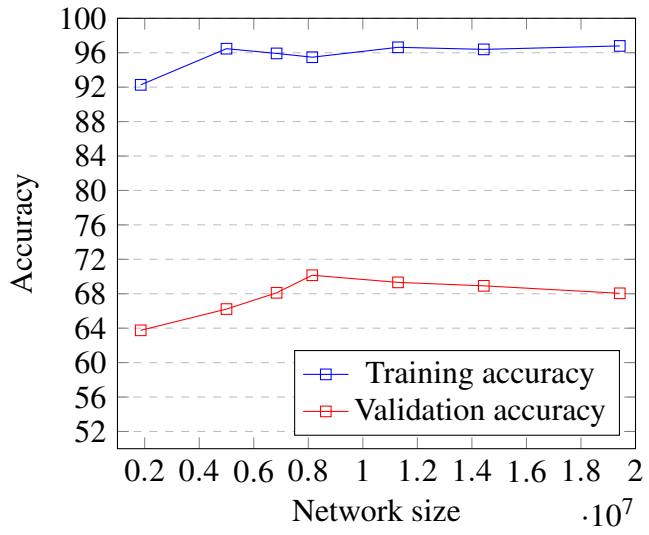

Figure 6.9 : Number of weights in the network vs. validation accuracy. Both 5 and 6-block deep networks are included

Previous experience with CNNs classification on Raspberry Pi had shown that models smaller than 5.000.000 parameters were able to run in real-time. However, the architecture was different, and proper runs with the new architecture are presented at the end of this chapter. Based on the graph above, it would seem like 5.000.000 would be a good complexity vs. cut-off point for the model to run in real-time on the board. Clearly, an accuracy below 70% is less than ideal for real-life implementation, but the magnitude of the project means that small incremental steps have to be taken in order to gradually improve the performance. The architecture of the best performing model is presented in Figure 6.10.

6.2.5 DROPOUT, BATCH NORMALIZATION, REGULARIZATION AND IMAGE AUGMENTATION

Considering the small size and high variability dataset used, it is difficult to estimate the impact of some parameters. Whereas increasing the depth of the CNN will lead to a significant improvement in performance up to a certain point, some parameters can give contradictory

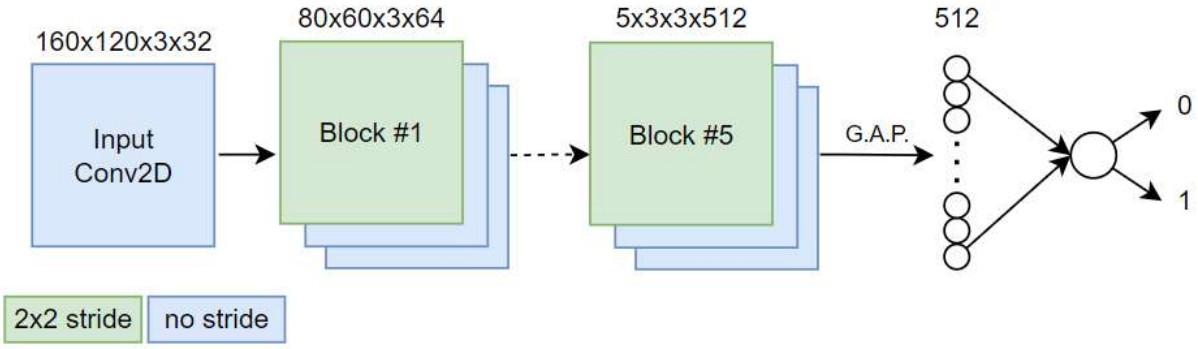

Figure 6.10 : CNN-5-3 architecture. The dimensions indicated are the ones of the output of each block. G.A.P. Stands for Global Average Pooling, and the green layers implement a 2x2 stride at the beginning of each block, which is the way dimensionality reduction is performed in the network. © Rani Baghezza, 2022.

results on successive runs. The objective was to establish a trend and have a general idea of the parameters that seemed to improve the performance of the model, without looking for an optimal configuration, as it would simply be impossible with this dataset.

When training a deep learning model, one of the most basic method to avoid overfitting is to use Dropout, especially in the dense layers of the CNN. Since overfitting is caused by the model learning the noise in the training dataset as well as the signal, the complexity of the model has to be reduced in order to avoid that scenario. A dropout value between 0 and 1 (excluded) is applied to a layer, and at each step, that value represents the probability of each unit being completely dropped out of the network. This allows to create sub-views of a simpler network at each step, and to force more or less responsibility on each node, as well as breaking co-adaptations between units that can lead the model to generalize less to unseen instances [219]. A dropout of 0.5 has been used for most of the runs, as a lower or higher rate did not seem to consistently improve the results. Adding dropout to the convolutional layers did not seem to an improve in performance as well, it was therefore only used in dense layers.

Batch Normalization has been a widely adopted technique in deep learning models to improve training. It is the process of normalizing the data within a mini-batch by setting the mean to 0 and the variance to one. Santurkar et al. [220] have found that using BatchNorm helped making the optimization landscape significantly smoother, which makes gradient descent easier and much more predictable. A BatchNorm layer has been added after every convolutional layer, which has led to better and more consistent results.

In terms of regularization, L2 regularization was used in the dense layers. L2 regularization adds a penalty term to the loss function based on the squared of the weights of the matrix at the current layer, which is essentially noise that the network has to deal with during training [221]. A coefficient α is used to control how strong the penalty is at each layer. After testing various configurations, it seemed like the general ideal range for L2 regularization was between 0.0001 and 0.001, depending on the complexity of the model. The larger the model, the higher the coefficient necessary to maintain a good generalization. As defined by Goodfellow, the best model is generally a large model with optimal regularization [19]. Below is a graph of a small grid search to find the optimal regularization parameter (Figure 6.11).

As observed below, changing the L2 regularization coefficient can lead to an increase in validation accuracy, in this case the highest accuracy reached is 70.15% with a coefficient of 0.0006. Increasing this value further seems to push the validation accuracy back down. It is possible that increasing the number of epochs could push this value back up, and that some local maxima exist at different values, however, performing a full grid search for each configuration is not realistic. Therefore, the default regularization coefficient used is 0.0002, with 0.0002 increments or decrements depending on the performance of the model. An extra run was performed using 0.01 regularization, which did not yield better results (67.34% validation accuracy).

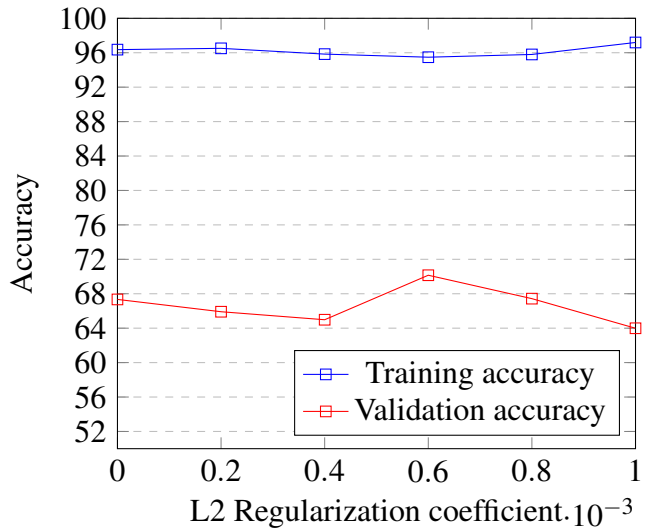

Figure 6.11 : Example of a small grid search to find the optimal L2 regularization value for a 5-block deep CNN with 8,144,641 parameters for 32 epochs of training.

In terms of image augmentation, it was difficult to pinpoint which transformations and which parameter values would lead to the best results. One sure conclusion is that image augmentation is necessary, as it allows to artificially inflate the size of the dataset, and to introduce even more variability, which helps the model generalize its performance better to unseen instances. Another aspect of these transformations is that they are applied randomly. For example, a parameter between 0 and 1 for the zoom range is provided. At each step, a random zoom is applied within the range of that parameter, either zooming in or out of the image. For a value of 0.25, which has empirically found to be ideal, each image is zoomed in or out by a factor anywhere from 0 to 0.25. This factor was chosen as it was a good middle ground between zooming in too much and losing a big portion of the silhouette, or zooming out too much, especially on already small silhouettes present in the dataset. It also helps the model deal with all sizes of silhouettes.

For the other transformations, the height shift was dropped since the last approach, as there is no reason to further crop the silhouettes vertically, considering that most of them are

already partial, and the zoom could make them even more cropped. Width shift was set to 0.1, leading to a slight possible translation of the silhouettes, which could prevent the model from relying too much on the position of the silhouette in the frame, but without go so far as laterally cropping boundary frames too much. The horizontal flip was set to *True*, meaning that silhouettes facing a direction could randomly be flipped so that they face the other direction. This allows the model to generalize better and not rely on the direction of the silhouette as much, which could have been a hidden correlation in a small dataset where unseen factors could be at play.

Finally, a random brightness transformation is applied, with a factor comprised within the range [0.9, 1.1] which slightly adjusts the brightness of the image, either up or down, depending on if the value is above or below 1. Since thermal images are sensitive to the temperature which affects the color and brightness of the image, this allows to add slightly more variability in the data in order to generalize better to unseen instances. Here again, using a very wide range has not led to better results, and in extreme cases, very bright images could be made so bright that they become useless in training, which makes a moderate range more justified.

6.3 RESULTS SUMMARY AND ANALYSIS

Once the bulk of the exploration and optimization has been performed on the gender recognition task, the model has been trained and slightly adjusted to perform age and mobility recognition. There are no guarantees that a completely different model could not be marginally better for age and mobility recognition, however, due to the similarity between age and gender recognition, we expect the best performing models to be of a somewhat similar complexity. The only parameters that have been adjusted for age and mobility recognition are the number of epochs, the L2 regularization coefficient, and the batch size.

Whereas a batch size of 48 has led to good results for age and gender recognition, it was necessary to bring this number down to 10 in order to train the network to perform mobility recognition, otherwise an accuracy of 50% was obtained regardless of the number of epochs used. This behavior could be explained by two factors: the low number of total instances in the training dataset, and the high inter-class variability, leading the Batch Normalization layers to normalize widely different instances, which could be the cause of this abnormal behavior. A summary of the CNN results for gender, age and mobility recognition can be found in Table 6.6 below.

Table 6.6 : Performance of a 5-blocks, 15-layers deep CNN for gender, age and mobility recognition.

Params.	Task	Epochs	L2	5th Fold	Valid	Best
8,144,641	Gender	32	0.0006	95.48	**70.15**	73.22
	Age	32	0.0001	95.48	**65.89**	70.99
	Mobility	64	0.0003	99.22	**94.77**	100

As illustrated by the results, the mobility recognition task seems much easier to perform, with a 94.77% accuracy and the best fold reaching a 100% accuracy. The CNN seems to perform better on gender recognition (70.15%) than age recognition (65.89%), even though it could simply be due to the fact that a different CNN architecture would be more suited to age recognition. However, a possible hypothesis would be that the shape of a silhouette and its size makes it easier to tell a male apart from a female, whereas the silhouette alone might not contain enough information to perform as well on age recognition. This initial analysis has to be understood in the context of the problem, which relies on the use of a small, high variability, thermal dataset without any additional pre-processing performed on the images. Existing literature in the field of age recognition has shown that CNNs could perform really well on big datasets containing pre-processed, bounded and normalized silhouettes [128].

If we consider that the no pre-processing constraint is fixed and necessary for a real-time system in the city, The main limiting factor is therefore the size of the dataset. Based on the number of gait covariates such as clothing, angle, silhouette size and crop, outdoors temperature, combined with the fuzzy boundary between classes, especially in the case of age classification, it seems clear that a much larger dataset is needed in order to push the performance of the CNN further.

6.4 EMBEDDED IMPLEMENTATION ON A RASPBERRY PI 3

Since the end goal of the system is to have deep learning algorithms perform real-time profile recognition directly on the nodes in the city, we explore the feasibility of performing real-time classification on the same Raspberry Pi 3 that was used to collect the data in the first place. In order to do so, a set of compact models has been built and optimized using local grid search to explore a small range of epochs and L2 regularization. The number of convolutional layers per block in the CNN has been reduced from 3 to 2 and 1 in order to build two compact models with a smaller number of parameters. A total of six compact models have been optimized, namely CNN-5-1 and CNN-5-2 for age, gender and mobility recognition.

A number of epochs ranging from 24 to 40 in increments of 4 and a L2 regularization coefficient ranging from 0.0001 to 0.0005 was explored for age and gender recognition. For mobility recognition, the same L2 coefficient was used, but epochs ranging from 56 to 68 have been explored, with the use of a batch size of 10 instead of 48. The best performing models after optimization are presented in Table 6.7 below.

As expected, smaller models lead to a decrease in performance for age and gender recognition. The reduction in model complexity prevent it from capturing more intricate features that could help discerning inter-class boundaries, and a drop of 2.26% and 1.48% in

Table 6.7 : Performance of the best compact CNN models.

Params	Task	Ep.	L2	5th Fold	Valid.	Best
4999009	Gender	40	0.0005	97.42	**69.75**	71.52
	Age	32	0.0004	95.30	**66.05**	69.75
	Mobility	68	0.0001	98.83	**98.49**	100.00
1853377	Gender	24	0.0004	89.42	**67.89**	71.21
	Age	28	0.0004	90.11	**64.91**	69.29
	Mobility	60	0.0004	100.00	**98.87**	100.00

validation accuracy is observed between the smallest model (CNN-1) and the biggest model (CNN-3) for gender and age respectively, compared to the 70.15% and 66.39% accuracy presented in the CNN section. However, it seems that mobility recognition actually benefits from the use of a smaller model, with an increase of 3.72% and 4.1% in validation accuracy for CNN-2 and CNN-1 respectively. The smallest model leads to the best performance for mobility recognition, further reinforcing the hypothesis that mobility recognition is a much easier problem to solve than age and gender recognition. The evolution of the performance of the model compared to their size is presented in the graph below (Figure 6.12).

However, these models are still running on a desktop equipped with a NVidia GeForce GTX 1080 using tensorflow-gpu. Once the best configuration has been found, each model has been trained again and saved on a hard drive. The best fold has then been transferred to the Raspberry Pi, as well as the validation data necessary to run the tests. Each model was then used to classify the same instances on both a desktop and the Raspberry Pi, using the same version of tensorflow (2.1.0), with the only difference being the hardware, and the use of tensorflow-cpu on the Raspberry Pi. The results obtained in terms of performance as well as classification speed are presented in the table below (Table 6.8).

The most noticeable aspect of these results is the gap in performance between the Raspberry Pi 3 and the desktop implementation. In the case of age and gender recognition,

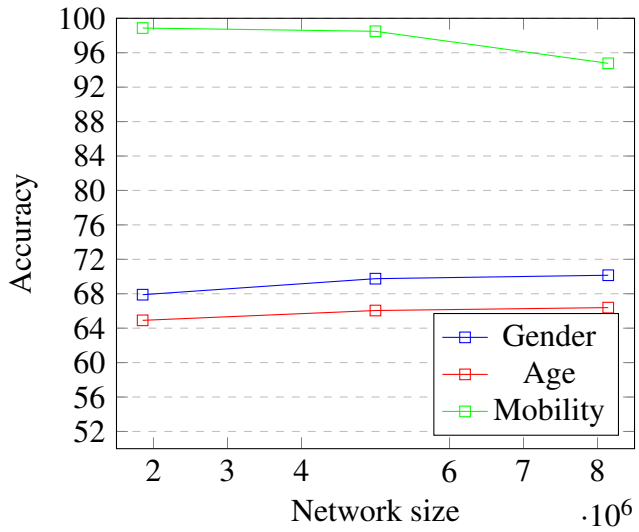

Figure 6.12 : Relationship between validation accuracy and size of the model for gender, age and mobility recognition.

a difference of 13.74% and 13.94% is observed in the worst case. The gap is even more apparent in the case of mobility recognition, where a drop of 41.57% is observed in the worst configuration. These results have been obtained running the exact same code in the same tensorflow environment and using the same dataset, which points to the hardware being the main issue, as well as the use of CPU instead of GPU to perform the computation. It has been observed that using GPU leads to much better performance than using CPU when it comes to tensorflow and deep learning in [222].

It seems that the issue could come from the snowballing effect of lower floating point precision at each layer of the network. A slight drop in performance when using a CNN for image classification a Raspberry Pi 4 compared to a Jetson Nano and a Jetson TX2 was observed in [223]. In that case, the accuracy dropped by 2.2% on a dataset containing 10.000 images. While this is far from what has been observed in our work, the CNN used in the paper mentioned above only contains two convolutional layers containing 32 and 64 units, which is a much shallower network than the ones we have used, containing between 5 and 10 layers,

Table 6.8 : Summary of the performance of the compact model on both the Raspberry Pi and PC. BS denotes the batch size used at classification time, time is expressed in milliseconds per instance.

Params	Task	BS	Raspi Acc.	PC Acc.	Raspi Time	PC Time
4999009	Gender	1	60.68	69.20	304.62	6.01
		8	**63.00**	69.20	**222.93**	2.66
	Age	1	55.40	69.14	345.08	5.94
		8	**56.48**	65.43	**247.54**	1.90
	Mobility	1	55.06	91.57	332.19	7.86
		8	**55.06**	90.45	**201.90**	5.21
1853377	Gender	1	**55.73**	66.41	151.87	4.59
		8	55.26	69.20	**119.05**	1.52
	Age	1	55.86	68.21	153.63	4.43
		8	**55.86**	62.96	**126.23**	1.50
	Mobility	1	**57.87**	94.38	170.95	5.97
		8	55.62	97.19	**130.26**	3.15

with a number of units ranging from 32 to 512. The network also worked with 32x32 images as opposed to the 120x160 images used in our work. A small imprecision could compound into a big error by the time the information propagates to the end of the network, leading to a higher misclassification rate, as the activation of the last neuron would be affected.

In terms of classification time, using a larger batch size allows for a faster classification, leading to a classification time per instance getting close to the 111.11ms mark, which is the threshold below which the system could classify images as quickly as they are captured by the FLIR Lepton (9Hz). Our conclusion is therefore that the system could run in near real-time with limited performance using the current hardware, but the most likely avenue for the future of this project would simply be the use of a more powerful board, such as the Jetson TX2, which is equipped for a GPU and is designed for deep learning [91].

Using a more powerful board equipped with a GPU would allow the use of tensorflow-gpu, which would most likely reduce the performance gap in terms of time and accuracy.

However, the energy consumption of the system will have to be measured, as the end goal of this project is to power the nodes using sustainable energy sources, such as solar panels [54]. The difference in performance between mobility, age and gender recognition could also point to the fact that different models could be used to solve different tasks in order to use the most appropriate model in each case. Complex architectures could also be explored with several stages of classification and feature extraction in order to optimize performance.

6.5 CHAPTER CONCLUSION

In this chapter, we have covered the use and detailed optimization of deeper CNNs after a more realistic sorting and organization of the data to perform age, gender and mobility recognition. The main conclusion of this chapter is that deep CNNs can be used to perform profile recognition using thermal images, but that the performance of the models are limited due to the challenging nature of the dataset, as well as the absence of pre-processing. A more advanced pipeline could be built with the use of a first model to perform silhouette detection and extraction before performing soft biometrics recognition in order to reduce the noise in the data, which could lead to a better performance. The main obstacle to a better performance seems to be the size of the dataset, which is insufficient to properly train a deep learning algorithm considering the complexity of the problem and the number of gait covariates.

CHAPTER VII

PROFILE RECOGNITION ON LOW RESOLUTION THERMAL IMAGES USING A SEQUENCE OF IMAGES

Using a single image was a mandatory step in order to see whether profile recognition could be performed with a still frame collected by a thermal camera. However, since the system is designed to be implemented in smart cities, it seems realistic to expect that people will walk and move in a linear manner, leading to the guarantee of being able to collect several frames per person. In this chapter, the use of a CNN combined with a RNN is explored in order to perform profile recognition using several frames, with the main motivation being the extraction of temporal relationships between successive images in a sequence, in order to learn more about each pedestrian's gait.

7.1 FROM A SINGLE IMAGE TO A SEQUENCE OF IMAGES

The main limitation when it comes to profile recognition with a single image in this project has been the variability in the dataset. No pre-processing has been used for several reasons mentioned throughout the thesis: because of the moving background and the high variability of the data in terms of silhouette size as well as silhouette crop, it is challenging to find a simple method of extracting silhouettes. A threshold-based approach would not be efficient as warm objects in the background, including cars and buildings, could throw off the algorithm. The lens used to view this project is always a real-time embedded implementation with no human intervention in the pipeline, which is important to remember. The images could be manually processed, and combining threshold-based approaches and a manual overview, followed by a normalization of the size of the silhouettes could lead to a uniform dataset similar to CASIA-like datasets in the field. However, a real-time implementation would virtually be

impossible without more complex algorithms performing this extraction and normalization on the fly. The time spent to build an algorithm that would perform a proper silhouette extraction and normalization would have prevented the approach of the actual problem of profile recognition. Besides, there would be no guarantee that unseen data could be properly pre-processed using that same algorithm, even if it is deep learning-based itself.

This has therefore led the project in the direction of applying deep learning on raw images. Using a single image becomes a difficult challenge, as the CNN has to learn through training how to tell the silhouette apart from the background and to classify it afterwards. There are no guarantees that the model did not pick up on other patterns present in the image, which would be difficult to spot in hindsight. The dataset is also too small for the CNN to be trained efficiently, especially considering gait covariates.

7.1.1 USING A SEQUENCE OF IMAGES

Using a sequence of images allows to leverage the temporal aspect of the gait. On a still image, the shape of the silhouette and the posture of the pedestrian are used to determine their profile. However, when considering gait, we are generally interested in the complete gait cycle of an individual. A lot of the approaches mentioned in the Related Work part of this thesis compute the GEI on an entire gait cycle, or on all of the available frames for the subject. A normalized gait cycle is shown in Figure 7.1.

Even though a single image has been used in a lot of cases to perform individual or profile recognition in the literature, hybrid methods combining models such as a CNN and a RNN have generally shown better results [111]. In terms of individual recognition, this conclusion makes sense as capturing an entire gait cycle gives a much better representation of the actual gait of the person, in terms of sampling frequency. Gait is a continuous movement

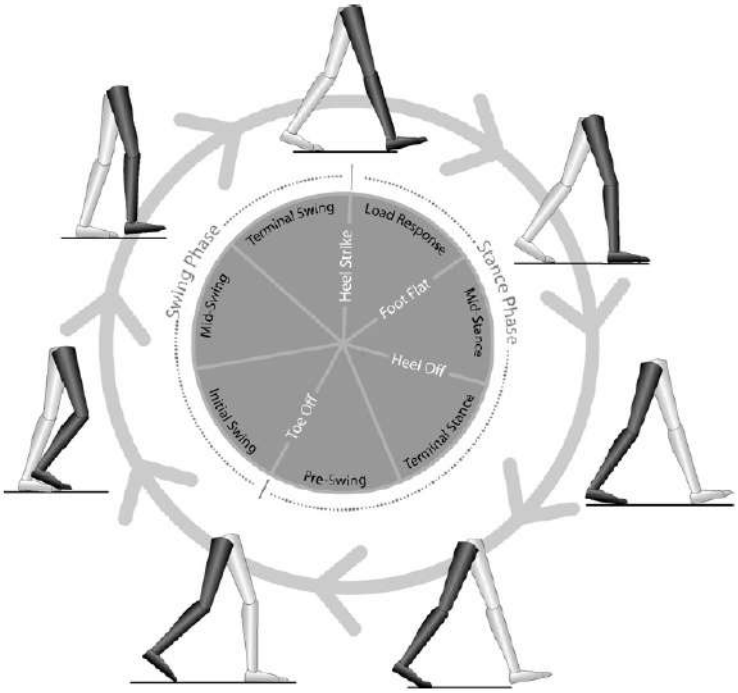

Figure 7.1 : A normalized representation of the different gait phases during the gait cycle by Rueterbories et al. [224]. © 2010 Elsevier.

that is captured discretely using cameras. The more sample points are collected, the closer we get to the actual gait of the person, which should logically lead to an easier discrimination between different individuals.

Profile recognition is a more challenging problem to solve than individual gait recognition for a few reasons. Even though multi-class classification is generally a more challenging problem than binary classification, the fact that gait is unique and different from individual to individual makes its recognition easy enough in the right conditions. When the data is not too noisy and when the feature space is large enough, it becomes easy to discriminate between classes. This has been achieved by removing the background from the silhouettes, normalizing the data and using deep learning methods in most of the literature. However, profile recognition, even if it is binary, already assumes that there is a correlation between

gait and profile. As mentioned in the last chapter of the literature review of this thesis, this correlation does exist to some extent. Where gait covariates are already challenging to deal with when it comes to the case of individual gait recognition, which is a problem that has relatively easily separable classes, it becomes even more complex for profile recognition. Adding the lack of pre-processing of the images and the small dataset makes this problem even more challenging.

Taking the example of age recognition using a sequence of unprocessed images, we are essentially capturing the noisy gait of a pedestrian captured under variable conditions and trying to determine whether they are an adult or an elder, when our reference is a similarly noisy gait dataset with a high variability. Moreover, the variations in individual fitness and health and the observer's bias are really capping the accuracy that can be obtained to less than 100%. Since age and gait are not entirely correlated, and since the age label is dependent on the observer's interpretation, there is a lot of room for errors and inaccuracies. This was already the case for a single image, but it becomes even more prevalent when using a sequence of images, as the dataset is much smaller.

On the other hand, using several images gives us much more information about a person's gait. This works both ways, where the model is trained on the actual gait of the person rather than a static silhouette and a posture, and any unknown instance contains much more gait information than a still frame of a silhouette.

7.1.2 EXISTING APPROACHES

In terms of profile recognition using thermal images specifically, no other approach has combined the use of a sequence of images, to the best of our knowledge. Such approaches have been used for profile recognition, as presented earlier. Batchuluun et al. [142] have combined

the best of both worlds when it comes to CNN and LSTM. They have highlighted that CNN lack temporal information, and LSTM tend to have a higher loss of spatial information, since the emphasis is put on the temporal relationships. Therefore, they have combined a deep CNN that works on a single image with a shallow CNN on top of which a LSTM is stacked, working on a sequence of images to perform individual recognition on thermal images. The pre-processing of the images as well as the feature extraction step is illustrated in Figure 7.2.

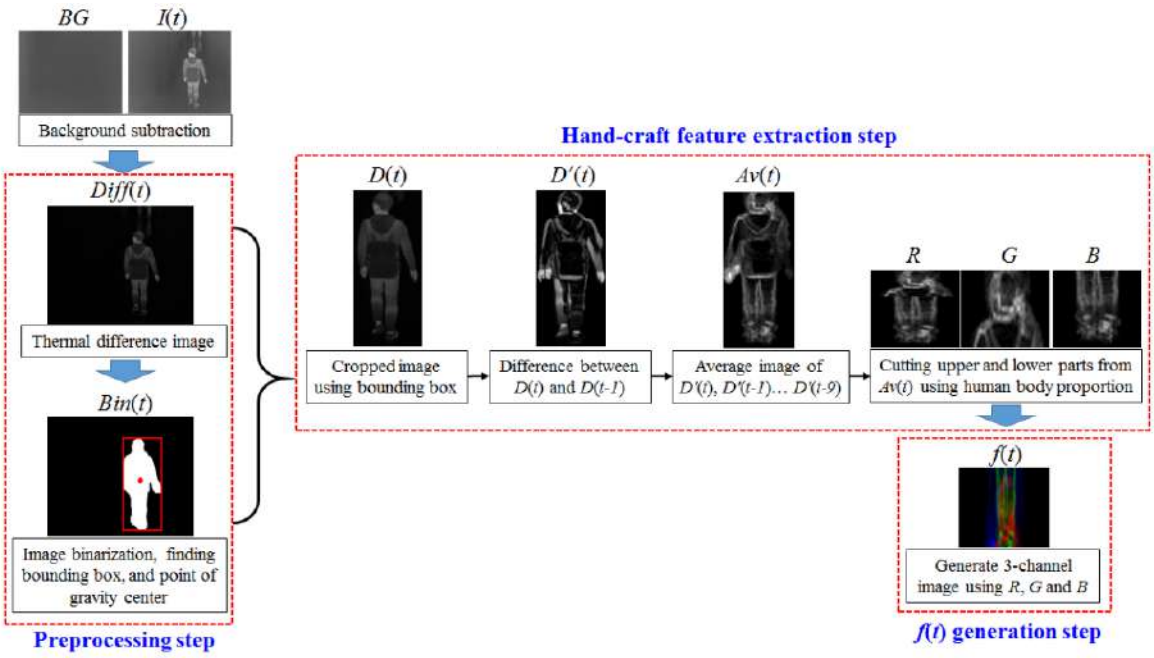

Figure 7.2 : Pre-processing and feature extraction step by Batchuluun et al. [142]. © 2018 IEEE.

Using a shallow CNN in the CNN-LSTM structure allows for a more efficient training of the network than stacking a LSTM after a very deep CNN. Additionally, combining the fusion of both models has allowed the authors to achieve a 99.71% CCR for individual profile recognition using thermal images. The architecture used for the CNN-LSTM is shown in Figure 7.3.

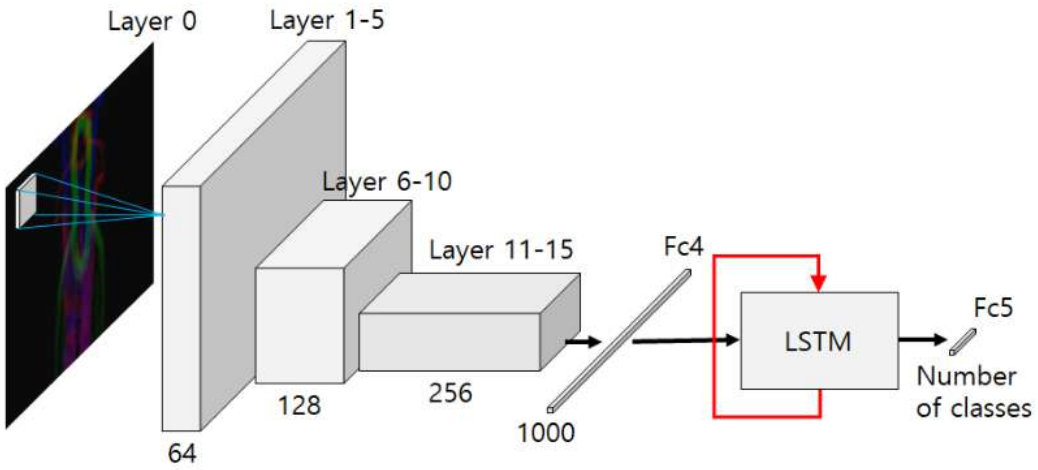

Figure 7.3 : Shallow CNN and LSTM architecture by Batchuluun et al. [142]. © 2018 IEEE.

Sepas et al. [128] have used a combination of RNNs, more specifically Bidirectional GRUs and CapsNet in order to perform profile recognition. This approach has already been presented in the beginning of the thesis, however, it is important to point out that the GRUs are used in a different way. Instead of using a frame as a time step, the GEI is first computed, capturing the temporal information of a gait cycle before splitting the silhouette into horizontal bands of varying size, learning the relationships between those using a BGRU before feeding the recurrently learned features to a CapsNet to learn part-whole relationships. The architecture used by the authors is shown in Figure 7.4.

The authors have achieved a CCR of 95.7% on average on the CASIA-B dataset, combining the results from the 11 different view angles in the dataset. Combining spatial and temporal feature extractions seems to yield very promising results. However, these approaches have still been limited to gait recognition, and even when thermal images are used, a lot of pre-processing is carried out on the data before learning takes place. It is therefore not realistic to expect to achieve such results when performing profile recognition on noisy thermal images, but it does give an argument for using and exploring these approaches.

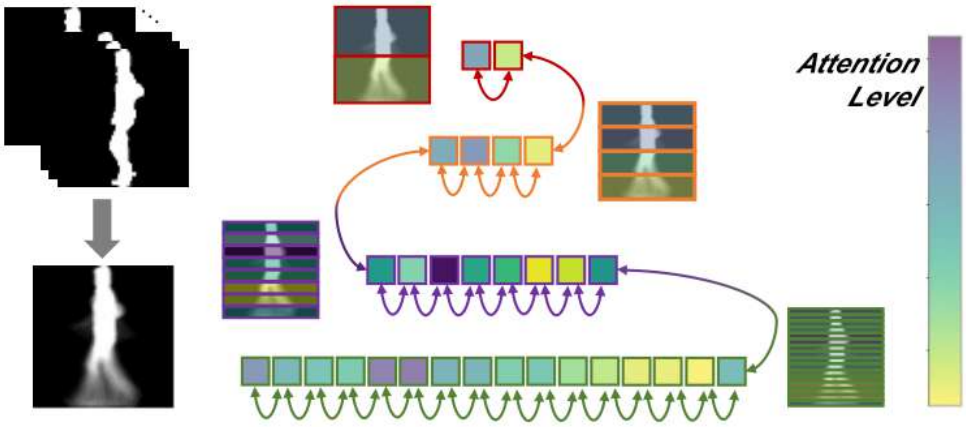

Figure 7.4 : BGRU architecture used by Sepas et al. [128]. © 2020 IEEE.

7.2 BUILDING AND OPTIMIZING A CNN-RNN

Following a similar approach to the one used in the previous chapter, the models are optimized on the problem of gender recognition, and results for age and mobility recognition are given using a similar model with minor adjustments when necessary. The best performing CNN from the previous chapter, namely the CNN-5-3 architecture, is used as the backbone of the model before adding the RNN part of the model, which takes the output of the CNN for each temporal step, processes it, and predicts the class of an observation after taking into account all of the temporal steps.

7.2.1 LSTM, GRU AND BGRU

When considering the use of a sequence of images and the combination of a CNN and some type of RNN, the first order of things is to determine which recurrent model to use. In the first chapter of the thesis, both the GRU and LSTM were presented as they are among the most common recurrent models used in the literature. While the LSTM is more complex

because of the extra gates, its main downside is that training is longer compared to GRU because of the additional gates present in the network.

In the context of this research, pedestrians walk on the sidewalk in front of a thermal camera capturing frames at a frequency of 9Hz. Depending on the means of locomotion used (walking, electric wheelchair, bicycle), more or less frames could be captured, leading to sequences of variable length. In order to simplify the research problem, and taking into account the fact that sequences of similar length have to be used in the same mini-batch during training, it was decided to use sequences of fixed length and to train several models in order to evaluate the performance of using more or less frames. If we only considered sequences containing 20 frames, it would be impossible to include observations containing less than 20 frames in the training and validation dataset. It was therefore decided to test the model for 2, 4 and 6 frames in order to keep the sequences short and include as many observations as possible. Training time also greatly increases when more frames are added, and longer sequences may amplify the exploding or vanishing gradient problem, despite counter-measures offered by the GRU and LSTM.

Since small sequences are considered, it seemed more suited to lean towards a GRU rather than a LSTM. Both models were tested using a few different configurations, and it seemed like using a LSTM did not give a significant improvement in performance and came at the cost of a significantly longer training. It was also challenging to rigorously measure the performance of using one model over another one because of the dataset being even smaller than the one used for the CNN. Therefore, couple of runs were performed using an unoptimized GRU and LSTM on sequences of 10 frames for 100 epochs, leading to the GRU reaching a 67.31% accuracy as opposed to 68.08% for the LSTM on the validation dataset. However, the validation f-score of the GRU was higher (0.708 as opposed to 0.694 for the LSTM). Since training the LSTM was about 20 to 30% longer than training a GRU, it was

easy to foresee that less combinations could be tested in the same amount of time, which, combined with high variance results, made the choice of GRU more justified.

In order to explore recurrent networks a bit further, a Bidirectional GRU (BGRU) was also evaluated. It is essentially the combination of two GRUs, where the first one propagates information in the forward direction, and the second one in the backwards direction. Where standard GRUs only learn from the past elements in a sequence, BGRUs learn from both past and future elements in the sequence, which could allow to extract additional temporal relationships from the dataset. The general BGRU architecture is presented in 7.5.

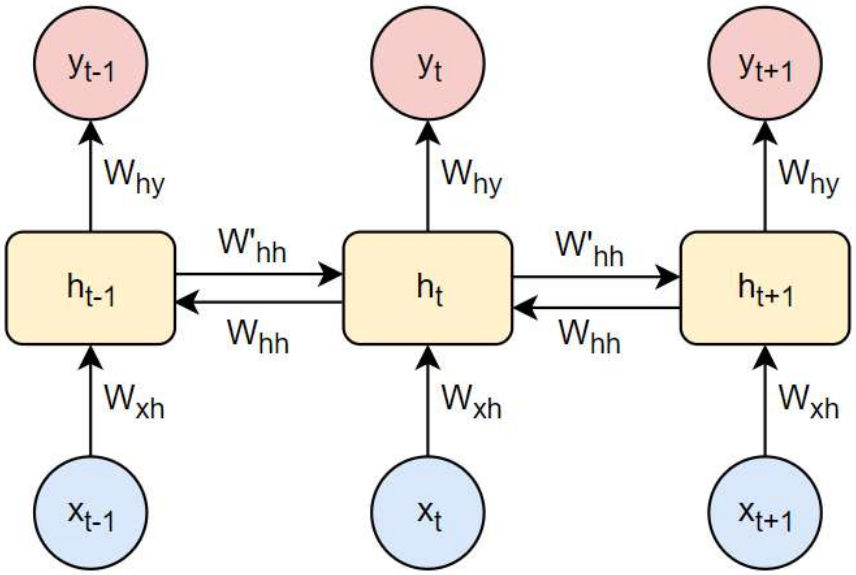

Figure 7.5 : General BGRU architecture. © Rani Baghezza, 2022.

7.2.2 DATASET

Since we are dealing with Recurrent Neural Networks, each instance is a sequence of images instead of a single image leading to a drastic reduction in the size of the dataset. The Table 3.6 below summarizes the number of instances. Where at least a thousand images were

present for each class using a CNN, this number drops to below a hundred, and only a couple dozens for validation, which greatly caps the results that can be obtained. In the table below, the actual size of the dataset used for both training and validation appears in the last column, as classes are balanced beforehand.

Table 7.1 : Training and validation dataset size and repartition for CNN-GRU. Class A is the majority class (male, adult, mobile) and Class B is the minority class (female, elder, reduced mobility).

Dataset	Profile	Class A	Class B	Total	Balanced
Training	Gender	154	82	236	164
	Age	136	91	227	182
	Mobility	221	15	236	30
Validation	Gender	33	26	59	52
	Age	39	20	40	59
	Mobility	57	8	63	16

Despite these limitations, the amount of information contained in a sequence of image is far superior to a single image, especially for gait recognition. This is illustrated in Figure 7.6 below. By learning the temporal and spatial relationships between the images in a sequence, the model can learn a representation of how a pedestrian walks, which, compared to other pedestrians with a different profile, can help the model empirically discover boundaries between classes.

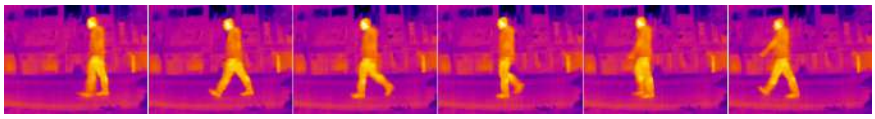

Figure 7.6 : Adult male observation example including 6 frames. © Rani Baghezza, 2022.

When working with a single image with a CNN, it was imperative to use some type of image augmentation in order to help the model generalize better. This also helped artificially inflating the size of the dataset which is always useful to train a deep learning model. When

it comes to using a sequence of image however, the same image augmentation should not be used. If random transformations are applied to each image in the sequence, all sense of cohesion is lost within the time series, as it no longer captures the unaltered gait of a pedestrian. An argument could be made for the use of transformations applied to the entire sequence, such as adjusting the brightness using the same factor on all images, or slightly zooming in or out, however, for this first approach, no image augmentation was used in order to see what results could be achieved on the dataset as it is.

7.2.3 ARCHITECTURE AND OPTIMIZATION

In terms of architecture, the best performing CNN (CNN-5-3) from the previous chapter was stacked on top of a GRU. Different configurations were tested using shallower and deeper CNNs, but they did not seem to perform really well. Interestingly enough, this CNN is 15 layers deep, which is the same number of layers as the CNN used by Batchuluun et al. [142], as presented at the beginning of this chapter. The authors have also worked with thermal images, even though they have pre-processed them and extracted features beforehand. Therefore, even taking into account the fact that the data used is not the same, the issue would probably be that deeper CNNs are more difficult to train when stacked on top of RNNs, as the gradient has to propagate from the very end of the network to the beginning, and as several timesteps use the same CNN. The general architecture used for the CNN-GRU is shown in Figure 7.7.

There are a few points to highlight about this architecture. Instead of using a 2x2 stride to perform dimensionality reduction at the beginning of each convolution block, a MaxPooling2D layer is added between each block. Using pooling instead of a convolution with stride seemed to lead to a more stable and better performance overall. Once again, performing a detailed analysis and a full grid search to optimize each parameter was not considered time efficient,

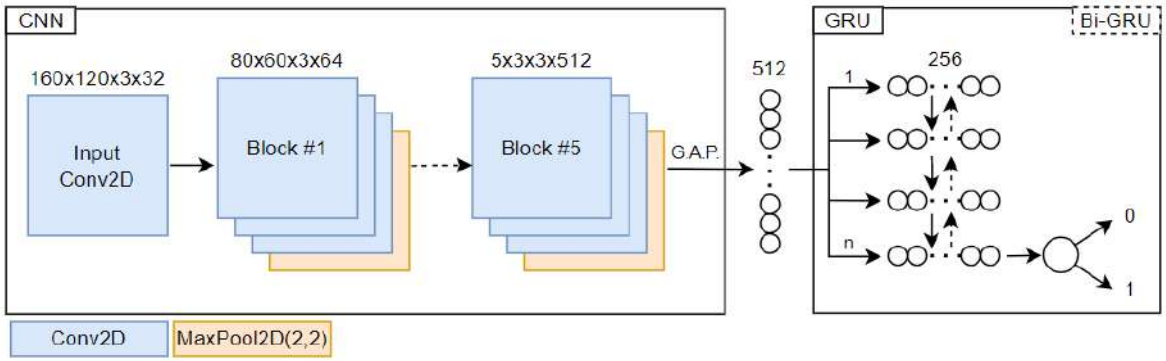

Figure 7.7 : CNN-GRU-5-3 architecture used to perform profile recognition on a sequence of images. © Rani Baghezza, 2022.

and because of the small dataset and the variance induced in the results, it was difficult to come to clear conclusions. However, it may be the case that since using a 2x2 stride in a convolution is equivalent to learning the dimensionality reduction operation, and since adding a RNN layer complicates training, the network might not be able to learn dimensionality reduction as efficiently. It could also be the case that max pooling extracts more salient features which translates into a better extraction of temporal information.

Throughout training and optimization, it was discovered that varying the size of the dense layer and the recurrent layer could help for longer sequences. Initially, it was found that using a dense layer containing 256 units (instead of 512 in the previous chapter), and a single recurrent layer containing 128 units led to better results than using other configurations. Using several dense or recurrent layers has led to consistently worse performance. However, after adding more epochs and using a dense layer containing 512 units and a recurrent layer containing 256 units, better results were achieved, especially on longer sequences. A 512-256 duo of dense and recurrent layer was therefore used from this point on.

The better performance of a larger recurrent layer seems to come from the fact that at 256-units recurrent layer allows the network to learn more complex temporal gait relationships, whereas a 128-units layer may force the network to learn a more coarse representation of these features. It could be the case that larger recurrent layers could lead to better results, however, the limiting factor would probably be the size of the dataset at this point.

The parameters used were the same in terms of learning rate (0.0001), but the number of epochs used was higher for CNN-GRU/BGRU than for CNN, and the regularization factor used was on average ten times higher than the one used for the CNN. However, reducing the complexity of the model and the regularization factor did not lead to a better performance. A small grid search to find the optimal regularization for a BGRU using sequences of 6 frames is illustrated in Figure 7.8.

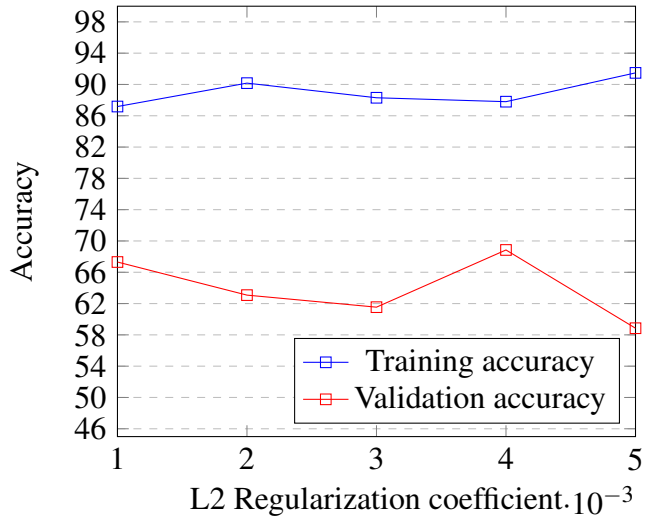

Figure 7.8 : Example of a small grid search to find the optimal L2 regularization value for the CNN-BGRU using 6 frames and 90 epochs

It seems like a local maxima appears around a coefficient of 0.004 using 90 epochs on a BGRU, and that a higher regularization does not lead to better results. It has to be pointed out that the old 256-128 configuration was used in this run, and it did not achieve sufficient

results compared to the use of the larger 512-256 dense/recurrent configuration. The optimal regularization for the bigger configuration turned out to be 0.003, so this grid search still allowed to get a general sense of what the regularization coefficient should be. Moreover, these results are all subject to variance, and this graph is here to illustrate an example of local grid search during the optimization process, without necessarily representing the best performance or the optimal architecture.

The number of epochs to maximize the validation accuracy has been significantly higher than for the CNN, due to the smaller size of the dataset and the additional temporal learning compared to the use of a CNN only. The optimal range of epochs has been found to be between 70 and 120, as opposed to 32 and 64 for the CNN. The process of optimization has been more straightforward than the one used for the CNN, as most parameters have been fixed. A standard grid search has been carried out in 10 epochs and 0.001 L2 regularization increments, starting at 70 epochs and 0.001 for most architectures, and ending at 120 epochs and 0.004 regularization. A batch size of 10 has been used for both the GRU and BGRU architectures. Indeed, each observation containing 6 frames has led to higher memory requirements, thus leading to a lower batch size in order to be able to run the computations without exceeding the memory of the GPU.

7.3 RESULTS AND ANALYSIS

The optimization process has been shorter for the GRUs and BGRUs as most of the work has been done when optimizing the CNN, which is the backbone of the CNN-RNN architectures. The results obtained for standard and compact models are presented in this section.

7.3.1 STANDARD GRU AND BGRU MODELS

Tables 7.2 and 7.3 below give a summary of the results obtained for gender and age recognition using CNN-GRU and CNN-BGRU architectures for sequences containing 2, 4 and 6 frames. A similar technique has been used for training, where 80% of the training data is used to train the model, and it is tested on leftover data, as well as on the validation dataset. After 5 folds, both results are averaged and presented in the table. However, since we are now dealing with entirely independent sequences of images, the results obtained on the 5th fold are actually completely independent from the training data. The gap between the 5th fold and the validation results has to be attributed to a possible bias in the construction of the dataset, leading the 5th fold to be closer to the training data than the validation dataset. However, the complex problem space and the limited size of the dataset make it difficult to pinpoint how the two datasets are different. Suffice to say that good performance has been achieved on both datasets, and the results illustrated in the last column, especially for age recognition using a BGRU speak for themselves, with the best validation fold reaching a 92.50% age recognition accuracy.

Table 7.2 : GRU and BGRU accuracies for gender classification. Fr indicates the number of frames in the sequence.

Model	Params.	Fr.	Epochs	L2	5th Fold	Valid	Best
GRU	8,743,361	2	90	0.001	83.60	**75.77**	80.77
		4	100	0.004	86.00	73.46	**82.69**
		6	120	0.002	**92.05**	73.08	80.77
BGRU	9,335,489	2	110	0.004	89.66	71.15	80.77
		4	100	0.003	**92.67**	70.77	84.62
		6	90	0.003	88.41	**78.46**	**86.54**

These results demonstrate the feasibility of performing age and gender recognition using GRU, and it would seem like BGRU has a higher potential due to its use of both past and future information in the sequence at each temporal step. This is illustrated by a better best fold using

Table 7.3 : GRU and BGRU accuracy for age classification.

Model	Params.	Fr.	Epochs	L2	5th Fold	Valid	Best
GRU	8,743,361	2	90	0.003	84.82	73.00	80.00
		4	80	0.004	84.00	**79.50**	**82.50**
		6	80	0.002	**89.14**	72.50	85.00
BGRU	9,335,489	2	80	0.004	84.83	**77.00**	**92.50**
		4	80	0.001	**89.14**	75.00	87.50
		6	100	0.004	87.49	74.00	90.00

the BGRU in both gender and age recognition. However, the best age recognition and overall results on the validation dataset were obtained with 4 frames and a GRU. Since variance has played an important role in these results, it is difficult to draw clear conclusions on the best model, but a combination of results analysis and intuition would make us lean towards the use of a BGRU with more data as a promising avenue to improve the performance of the system, especially if an entire sequence can be collected before performing classification.

There is no direct linear correlation between the number of frames used in the sequence and the classification accuracy, however, this could simply be due to the fact that the more frames are used, the bigger the dataset should be in order to train the model efficiently. Using longer sequences means that throughout training, the model will learn more complex gait behaviors and temporal relationships summarized in a 256-units recurrent layer. Finding non-linear boundaries between classes might become more challenging when more intricate temporal relationships are extracted from each gait sequence, and more instances might be needed for the model to be able to generalize better.

Another important observation that can be made from looking at these results is the fact that leveraging the temporal relationships of gait data seems to give better results for age recognition than for gender recognition. Whereas better results for gender recognition have been achieved with the use of a single frame with the CNN, using a sequence of frames

seems to be more promising for age recognition. It could be the case that spatial and static information is more efficient to perform gender recognition than age recognition, such as the shape and size of the silhouette, which would confirm existing findings in the literature. Baek et al. [91] have analyzed the learned features of their 2-CNN architecture, where one CNN is used on visible images and the other on thermal images to perform gender recognition. They have found that the visible light image-based CNN gave more importance to hairstyle and face, and the IR-based CNN gave more importance to body shape and silhouette features. Static silhouettes may contain less age-related information than gender-related information, especially using unprocessed thermal images, and when considering a high proportion of partial silhouettes, such as the dataset used in this paper.

7.3.2 COMPACT BGRU MODELS

Following along with the previous chapter, compact CNN-BGRU models, namely CNN-BGRU-5-2 and CNN-BGRU-5-1 were optimized in order to be tested on the Raspberry Pi 3 to perform real-time classification. However, it turned out that the models simply did not run on the board, due to a lack of memory. The issue comes from the fact that Keras' TimeDistributed layers have to be used when working with CNNs and RNNs, where each image in a sequence is processed through the same CNN before being fed to the RNN. This leads to a higher memory requirement as several images are processed at the same time, which is combined with the additional weights of the BGRU, leading to a model that is too complex to run on a board that is not designed for deep learning operations, and that uses tensorflow-cpu instead of tensorflow-gpu. The performance of the compact optimized BGRU models running on desktop are shown in Tables 7.4 and 7.5.

Comparing these results with the results of the larger models leads to the conclusion that there is a correlation between the number of parameters of the model and the age and gender

Table 7.4 : Compact BGRU accuracies for gender classification.

Size	Params.	Fr.	Epochs	L2	5th Fold	Valid	Best
1	3,040,257	2	110	0.003	84.17	68.46	73.08
		4	110	0.002	92.86	**72.31**	78.85
2	6,187,873	2	100	0.004	90.81	**78.08**	86.54
		4	90	0.001	87.16	74.23	86.54

Table 7.5 : Compact BGRU accuracies for age classification.

Size	Params.	Fr.	Epochs	L2	5th Fold	Valid	Best
1	3,040,257	2	80	0.002	86.40	70.00	80.00
		4	90	0.002	83.59	**73.00**	85.00
2	6,187,873	2	110	0.003	84.77	**75.00**	87.50
		4	90	0.002	88.45	74.50	82.50

recognition performance. However, it would seem that a model with 6 million parameters performs almost as well as a model with more than 9 million parameters, as illustrated by the results. For example, the best validation accuracy for gender classification was 78.46% was achieved with a 9,335,489-parameters model using 6 frames, whereas a 6,187,873-parameters model achieved a 78.08% accuracy using 2 frames. The accuracy of the best fold was identical in both cases (86.54%). In the case of age recognition however, the best model achieved a 79.50% accuracy using 4 frames as opposed to a 75.00% accuracy for the second biggest model. The accuracies of the best fold are 92.50% and 87.50% respectively. The relationship between the size of the network and its accuracy for age and gender recognition is illustrated in Figure 7.9 below. The same relationship is illustrated for the accuracy of the single best fold in Figure 7.10.

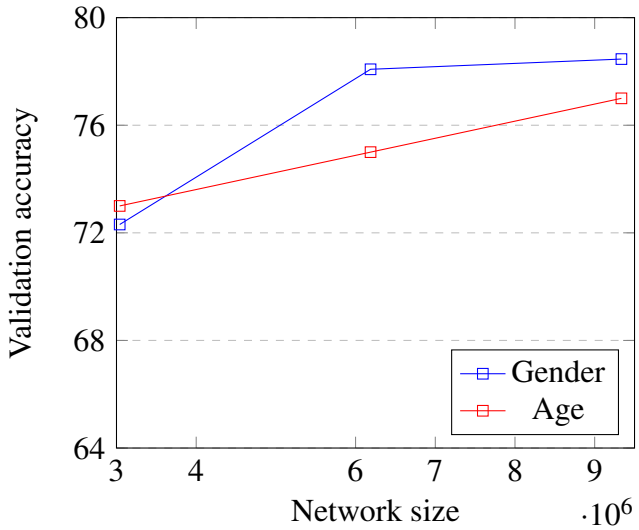

Figure 7.9 : Relationship between validation accuracy and size of the model for gender and age recognition using BGRUs

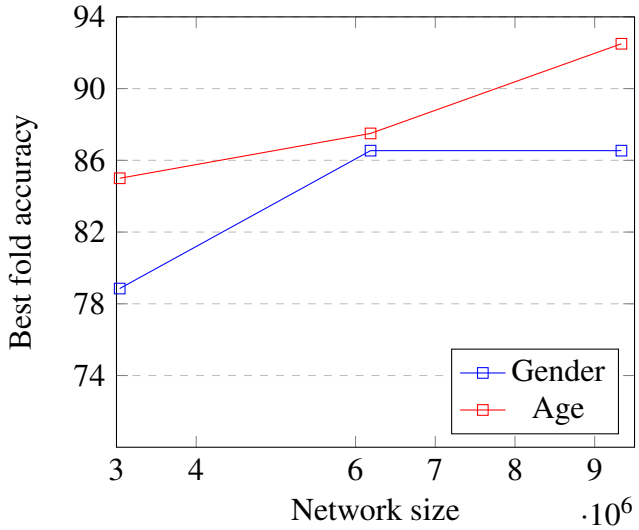

Figure 7.10 : Relationship between the best fold accuracy and size of the model for gender and age recognition using BGRUs

The smallest models have achieved an accuracy of 72.31% and 73.00% for gender and age recognition, respectively, which gives an idea of the complexity of the problem. It would seem like a model with 3 million parameters is simply not large enough to solve the problem of profile recognition, considering the variability in the dataset and the gait covariates. It seems like there is a linear correlation between model size and age classification accuracy when comparing BGRUs only. This is seen in both the validation accuracy and best fold accuracy graphs above. However, three data points do not provide enough information in order to draw more general conclusions. Ideally, larger networks should be used to perform age recognition in order to see whether the trend is confirmed, or whether it is just a coincidence. However, the fact that the gender curve flattens past 6 million parameters for both the average validation accuracy and the best fold accuracy might point towards the fact that bigger models would not help for gender recognition, as opposed to age recognition. Once again, a bigger dataset would be necessary in order to draw more general conclusions, but these results give a solid foundation on which the system could be extended in the future.

7.4 LIMITATIONS AND FUTURE AVENUES

It seems clear that using several frames leads to a much better profile recognition performance, even with a challenging dataset and without any image pre-processing. It is likely that the process of using several frames and having a RNN learn the temporal relationships between the frames almost acts as a pre-processing operation. The model has to learn how the frames relate to each other, which leads it to learn about the moving parts of the frame. In most cases, the background is static, which naturally leads the model to learn that the interesting part of the image where temporal relationships can be learn is located around the silhouette. The presence of cars in the background as well as other pedestrians on the

opposite sidewalk can be an issue in some cases, but it is safe to assume that a sufficiently large dataset would mitigate the contribution of these factors.

Additionally, there are many more avenues to explore in order to improve the performance of the model. The first step could be the collection of more data, which could be useful for the CNN as well as for the CNN-RNN. In terms of architecture, the use of an attention module could help the model pay attention to the important parts of the images [225], and replacing the CNN part of the architecture with a CapsNet could help the model learn more part-whole relationships, and lead to a better implicit understanding of what constitutes the background and what constitutes the silhouette [140]. Ultimately, a Single-Shot Multi Box Detector [226] could be used in order to detect the silhouettes and extract them before normalizing them and feeding them to the new architecture. The combination of extracting the silhouettes beforehand, using more advanced models and gathering more data could lead to a sharp increase in performance of the model.

In terms of embedded implementation, deep learning optimized boards such as the Jetson TX2 [91] could be used in order to achieve better results. It seems clear that the Raspberry Pi is not the optimal candidate for this task. However, the use of a more powerful board also implies that the initial calculations that were carried out in order to see whether the system could be powered with a solar panel will have to be re-evaluated, and that real-life tests will have to be carried out in order to see whether it could be possible to run the boards solely using sustainable energy. The illustration of a possible embedded system can be found in Figure 7.11 below.

Since privacy is an important concern, and even though using thermal silhouettes instead of RGB silhouette reduces the probability of being able to identify a pedestrian, images could be periodically deleted from the device after they have been used for retraining. Each

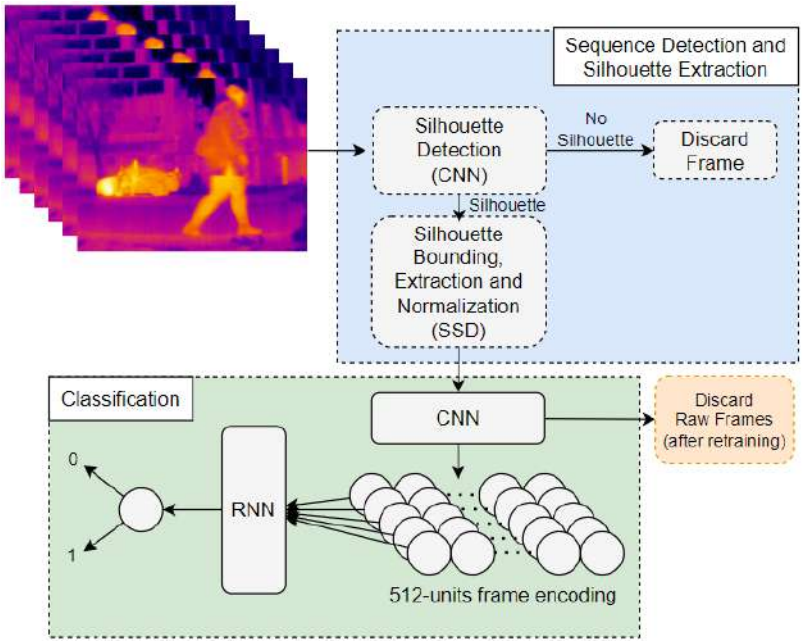

Figure 7.11 : Example of a possible embedded architecture for real-time profile recognition using thermal images. © Rani Baghezza, 2022.

frame would be analyzed by a first CNN that would have a simple task: estimate whether or not the frame contains a silhouette. This task could already be challenging because of the cars, motorcycles and different types of silhouettes that could be observed by the system. If the frame is empty, it is immediately discarded, as it is of no use to the system. If the frame contains a usable silhouette, and here, the boundary of what is usable will have to be determined, it is then fed to another model that would detect the position of the silhouette, create a bounding box around it, and extract it from the rest of the image. The following CNN would serve as a way to encode the image into a compact representation that could then be fed to a RNN, and each time step would be used to perform a prediction regarding the age, mobility or gender of the observed pedestrian. A variable input size RNN could be used at this step, and each additional frame could help it improve its prediction.

Each frame could be represented in a compact 512-units representation, and deep learning models such as Deep Auto Encoders (DAE) could be trained to compress and decompress an image if enough real-world data is available. Several dense layers could also be used for different degrees of compression and encoding. In terms of size, the thermal images we have captured are generally around 35.000 octets. Compressing this information using a 512-units layer would lead to the use of 512 float32 variables, each occupying 4 octets. This would lead to a 2.048-octet representation of the image, reducing the storage need for each image by a factor of 17. It could also be the case that bounding silhouettes and extracting them could reduce the required size of the dense layer, as the noise coming from the background would not be present in the image anymore.

This is just a possible example of the architecture that could be used, and we can see how it could be generalized to many more applications. The silhouette detection step could be generalized to perform object detection before subsequent classification steps take on the task of recognizing these objects. In this way, the system could be extended to many tasks in the city, such as drivers emotion recognition [227], pedestrian, bicycle and cars counting [46] as well as suspicious activity detection [228]. Moreover, the implementation of such a system would allow the exploration of real-time embedded thermal vision, re-training and concept drift handling [22], as well as semi-supervised learning [229]. Moving towards pervasive intelligence in the environment also requires the exploration of optimizations for embedded deep learning algorithms. The emergence of new classes and situations would make this system an ideal testbed to explore flexible and more generalizable algorithms that could be extended to many vision tasks.

Part V

Conclusion

CHAPTER VIII
GENERAL CONCLUSION

This thesis has tackled the challenge of performing gait-based soft biometrics recognition using low resolution thermal images in the city. A lot of ground has been covered since the beginning of the thesis, as the core problematic was to extend ambient assisted living from smart homes to smart cities. From this vague idea, and through the exploration of how activity recognition worked in smart homes, as well as the exploration of wearable-based outdoors approaches, a first prototype was designed and built in order to perform activity recognition using environmental sensors on a portion of pavement in the city, as presented in Chapter 5. From the results and the conclusions drawn from that first experiment, the core problem of the thesis has transitioned from the recognition of activities to the recognition of pedestrian profiles, more precisely, soft biometrics using gait recognition.

Most existing gait-based soft biometrics recognition work has relied on RGB or active IR cameras, leading to no privacy guarantees for the people involved in the experiments. Moreover, most existing work in the field has focused on pre-processed, extracted and bounded silhouettes in order to achieve the best possible gait recognition results. The main issue with these approaches, despite the very good results they have achieved, is their lack of applicability in a real-life context. In the city, the background fluctuates, and the images collected by the thermal cameras include a lot of noise. There is no control over the distance between the pedestrians and the cameras, and various issues, such as dealing with partial silhouettes, missing frames, as well as cars and other pedestrians in the background have to be dealt with.

The complexity of the task of performing soft biometrics recognition on a custom-made dataset using a technology that has not been thoroughly exploited in the scientific literature

has led to a large tree of possible avenues to explore which had to be pruned and simplified in order to explore the more interesting and pressing challenges.

8.1 ASSESSMENT OF THE CONTRIBUTIONS

After an initial survey on distributed, real-time activity recognition published in the Sensors journal [22], the first main contribution of this thesis has been the review of the existing literature in an organized and cohesive format in Chapters 3 and 4 in the fields of gait-based hard and soft biometrics recognition, namely identity, age and gender. A trend towards the use of deep learning algorithms has been identified in both fields, and the main challenges have been highlighted, such as dealing with various view angles, gait covariates and accessories. The link between age and gender has been highlighted, and it could be an avenue to explore in future work. Existing work using infrared and thermal images has also been reviewed in order to close the gap between standard, RGB-based profile recognition and this thesis. This step has allowed to slowly funnel the content of the thesis from individual gait recognition to profile recognition while gathering useful bits of information regarding the standard architecture used, the depth of the networks, the size of the dense layers, as well as the different strategies used to mitigate the effect of gait covariates. This part will be useful for future work in the field, as time constraints have made it impossible to explore interesting approaches such as DAEs or GANs, as well as CapsNet that could lead to very interesting results.

The second contribution has been the design and fabrication of the first two prototypes presented in Chapter 5 with associated publications in the ANT conference [53] and the IEEE Internet of Things Journal [54]. The first prototype has been a way of getting more familiar with the hardware, as well as the field of activity recognition in general, and it has allowed to redirect the thesis towards profile recognition. The second prototype has been a good

improvement over the first one, and it has allowed the collection of higher resolution data in order to build a custom gait dataset containing thermal images. It has allowed to design a better experimental setup by choosing a more appropriate location and a smarter positioning of the node so that they would not be as obvious to the pedestrians. This phase also includes the first results obtained on the full dataset.

The third contribution has been the optimization of a deeper CNN after sorting the data in order to keep single pedestrians only in the dataset. Various parameters have been explored, starting with the depth of the CNN and the size and number of dense layers. After that, the optimization algorithm to perform the gradient descent, the activation function, the dropout, batch normalization, regularization and image augmentation have been investigated and tuned in order to find the best performing model on the new dataset. This part also includes the optimization of the CNN-GRU and CNN-BGRU models that has mostly been about tuning the size of the dense and recurrent layer, as well as the regularization and the number of epochs.

Finally, the fourth contribution has been the implementation of the CNN directly on the Raspberry Pi in order to evaluate the accuracy of the models as well as the feasibility of performing real-time classification on the board. The CNN-BGRU could not be implemented on the board due to its limited memory, but compact models have been optimized and conclusions have been drawn between the size of the network and the performance of the profile recognition task.

8.2 LIMITATIONS OF THE WORK

Despite the encouraging results obtained and reported in this thesis, especially using the CNN-GRU-based approach, there were a few limitations that are worth pointing out in this section.

8.2.1 FIRST PROTOTYPE

The limitations of the approach using the first prototype mainly come from the low resolution of the sensors coupled with a very small dataset. On a sample containing 140 observations, and with the use of approximate sliding windows around the observations to compute basic features used for conventional machine learning, the results obtained are bound to be limited. It has also been identified that the position of the nodes brought on a bias for age and activity recognition: the height of a silhouette was enough to distinguish between some of the classes, thus, positively affecting the accuracy for these tasks.

Moreover, node positioning was far from optimal during the 2019 experiments. The nodes were clearly visible to a lot of the pedestrians, leading them to affect their gait while walking, or sometimes stopping to look at the nodes. Because of the experimental setup, it was not always possible to take a note quickly enough to identify the affected observations, leading to noise and imprecise observations being present in the dataset. However, since the goal of this first prototype was to see whether the idea of activity recognition was even realistic in the city, the lessons learned from this first approach were invaluable.

8.2.2 SECOND PROTOTYPE: SINGLE IMAGE

Using the Raspberry Pi 3 to collect thermal images has led to a significant improvement in the raw quality of the data. However, the first approach using a shallows CNN on the entire dataset without any sorting or pre-processing has proved to be flawed in a few different places.

First of all, all tasks (age, gender, group size and mobility recognition) were performed on all instances in the dataset. This means that age and gender recognition were performed on groups, according to the label that was defined by the age/gender majority. In groups containing one person of each class, the results cannot be anything else than random. Additionally, age

and gender recognition were performed on people riding bicycles, skateboards or wheelchairs. In the first two cases, the blur induced by the movement in the image would make the task of recognizing someone's age or gender much more difficult. Besides, the task is no longer gait-related at this point, since there is no formal gait, but simply someone riding a bike or a skateboard. Finally, the fact that most people in wheelchairs were elders allows the algorithm to find hidden correlations and give results that are too optimistic.

Secondly, the data used included a lot of what was referred to as "boundary silhouettes", which are silhouettes cut along the vertical axis, symptomatic of someone entering or leaving the camera's frame. The task of recognizing someone age or gender based on a single low-resolution image is hard enough, but the addition of images only containing two limbs makes it much more difficult. Additionally, it reduces the part of the image containing useful information for classification, as most of the image captures the background. The additional image augmentation could also make the silhouette completely disappear in some instances (boundary silhouette to the left of the frame shifted even more to the left with the width shift transformation, for example).

Lastly, data shuffling was performed after class balancing. In the case of age recognition, it is easy to identify how this led to a clear advantage in terms of accuracy. The elders always constituted the minority class. In each directory (adult, elder), the observations are organized in alphanumerical order. In each case, the adult observations were truncated so that their number would match that of the elder class. This led the observations from latest experiments (September 24th and September 25th) to be systematically excluded in the adult directory. Since the temperature and the camera setup were slightly different on each experiment, this could lead the model to learn that images with a specific shade of colors corresponding to these experiments were always capturing adults. However, the application of image augmentation could have mitigated the effect of this issue, but it is important to specify its existence.

8.2.3 SECOND PROTOTYPE: SEQUENCE OF IMAGES

The main limitations of the work come from the limited size of the dataset coupled with the complexity of the profile recognition tasks. The results achieved, especially using a CNN-BGRU architecture have been very promising, and the main question of the thesis has been positively answered: it is possible to perform profile recognition on unprocessed low-resolution thermal images in the city. When it comes to the real-time part, the conclusion seems to be that it would not be possible to perform real-time profile recognition with a Raspberry Pi 3, as the performance of the embedded algorithm is too low in terms of accuracy, and just above the real-time threshold in terms of classification time. However, this has led to the conclusion that the tasks should be feasibly on a deep learning-oriented board such as the Jetson Nano or the Jetson TX2. Proper research should be carried out in order to draw factual conclusions on that matter.

The results obtained by the CNN-BGRU for both age and gender recognition are very encouraging considering that less than two hundred instances have been used, and it seems that a bigger dataset could lead to even better results. Moreover, performing experiments in different locations in the city as well as different times, such as early in the morning or during the night could lead to the collection of a more complete dataset that would allow to address a larger problem. However, the more variation are added in terms of time of the day, location of the camera, the higher the number of different profiles and gait covariate, the larger the dataset needs to be in order to properly generalize the performance of deep learning algorithms to unseen instances.

The low resolution of the camera is also a limiting factor when it comes to profile recognition accuracy. In most work mentioned in the Related Work part of the thesis, the cameras used capture much higher resolution data, from 640x480 for thermal cameras all the

way up to 1280x720 for RGB cameras, this leading to a lot more information being captured. However, this limitation is justified by the fact that a lower resolution is an additional privacy guarantee and that the end goal of the system is to run in real-time on a limited resource board.

8.3 FUTURE AVENUES

There are many future avenues for this project, as illustrated in Figure 8.1 on the next page.

8.3.1 DATASET AND DEEP LEARNING

Since the main limitation of this work comes from the size of the dataset, it seems natural to run more experiments in order to collect more data. This step could also be coupled with a full revamp of the prototypes in the form of a transition towards a deep learning board. The case of the prototypes could themselves be optimized in order to be smaller and more discreet, and various engineering solutions could be explored in order to attach the cases directly to the facade of a building, allowing much more flexibility in terms of locations for the experiments.

As stated in the previous section, experiments could be performed at night in order to assess the efficiency of thermal cameras in a dark environment. This would also lead to interesting research problems in terms of transfer learning as well as concept drift: what is the performance of a system trained during the day on data collected at night (and vice versa). Should different models be used depending on the time of the day? What parameters could be adjusted, and could we find a model that performs well in any situation? The exploration of artificial data in order to pre-train the algorithms could also be a way of avoiding the repetition of long and tedious experiments in the city [155].

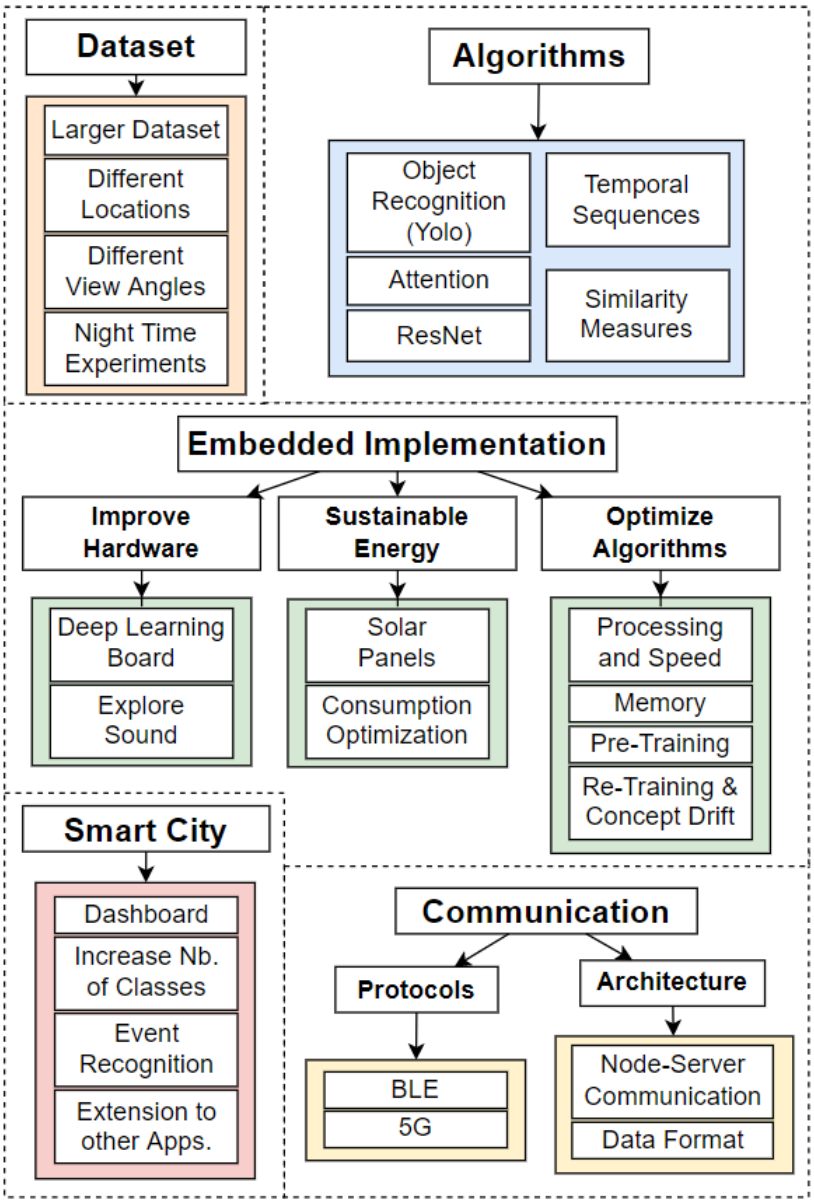

Figure 8.1 : Illustration of some of the possible future avenues for this project. © Rani Baghezza, 2022.

In terms of deep learning, there are many more interesting perspectives and challenges to address. Various models could be explored, such as Siamese Networks [230] for their ability to perform well on a smaller dataset through the use of one shot learning, ResNets [176] for

their ability to allow the training of deeper networks, which would allow the full exploration of deep CNNs, Attention modules [225], that would allow the model to learn where the important parts of the images are, in this case, the silhouettes. An argument could also be made for the exploration of CapsNets [140] instead of CNNs, as they are designed to capture part-whole relationships, which could be useful to tell silhouettes apart from the background. Thanks to the experience accumulated throughout this thesis, as well as the comparison with results in the field of gender recognition [204], it seems clear that extracting the silhouettes should lead to better results. The next step will therefore be to use models such as Yolo or Multi-Box SSD [231] before using a vision model to perform soft-biometrics recognition on pedestrians.

In hindsight, the problem has shifted from a smart home approach, where low resolution sensors were used to copy and paste the smart home model to smart cities, before moving to a version of the prototype with better sensors, and a shallow CNN-based approach, before eventually adding gait techniques relying on temporal information in the latest approach. However, all of these approaches have shown their limits, the main of which has been dealing with other objects and entities present in the images. In most gait datasets, the environment is controlled and subjects are walking from one side to the other without any other object or person being present in the background. In a real-life system, the activity of the city makes these cases rare and hardly representative of the real problem to solve. This evolution is illustrated in Figure 8.2 below.

It now seems clear that extending ambient assistance from smart homes to smart cities requires solving a vision problem first and foremost, before solving a gait-based profile recognition problem, before solving the actual ambient assistance problem. The best course of action therefore seems to be to transition towards a two-stage recognition problem: first we recognize and extract objects in the image, such as cars, bikes, wheelchairs, or pedestrians,

then we recognize the profile of pedestrians. The first stage of recognition can already be useful for smart cities applications, as described in the last subsection, a few pages below.

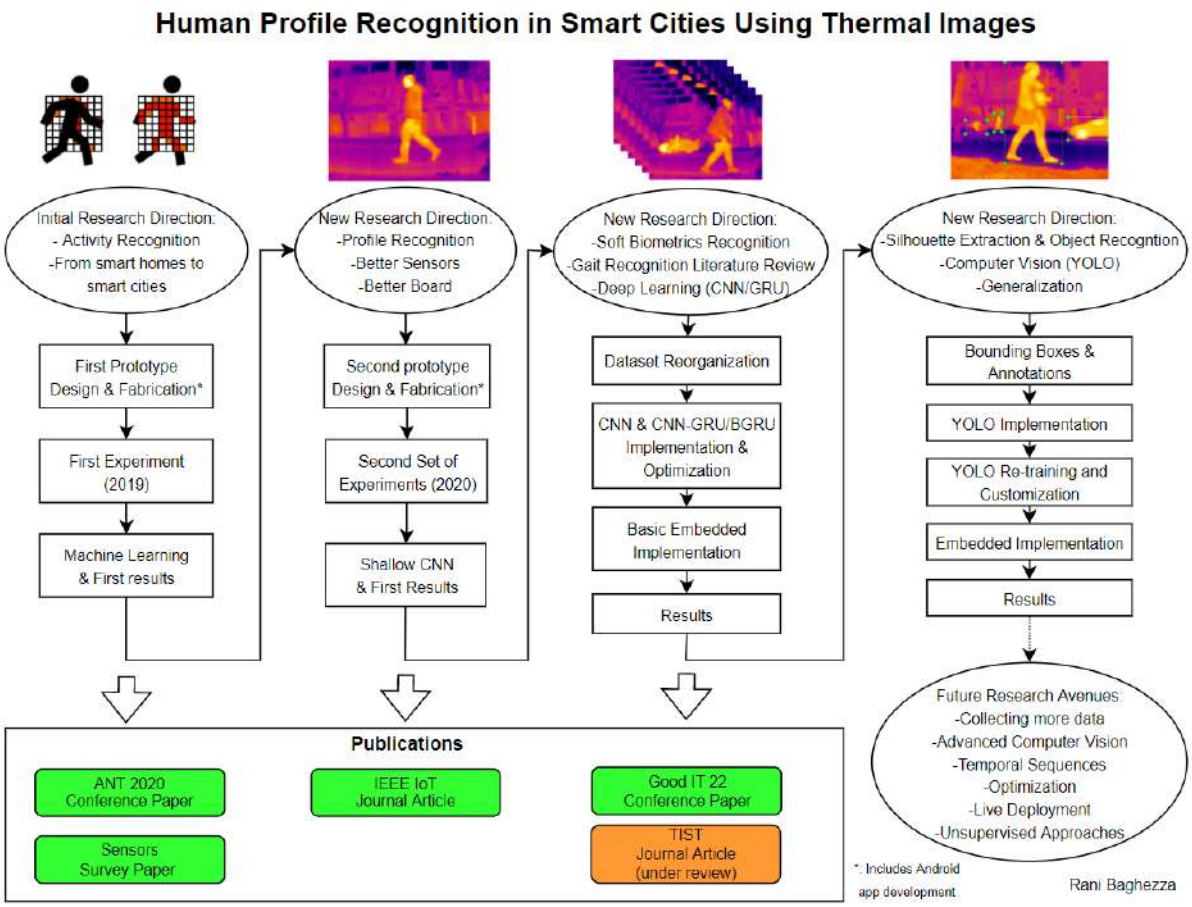

Figure 8.2 : General overview of the different steps of this project © Rani Baghezza, 2022.

The main obstacle to the second stage of the process is the availability of enough high quality annotations collected during experiments: this is where the limits of the current experimental setup start to appear. During the experiments, ground truth for age and gender was only valid for single pedestrians walking across the frame: it is not possible to know the age or gender of an entire group without collecting an individual label for each person in the

group. Therefore, a more advanced experimental setup should be developed, with the ability to either see the thermal feed in real time, synchronize it with a tablet, and collect labels in real-time, or with the help of ground truth coming under the form of CCTV camera data shared by the city, with the appropriate privacy and ethics concerns. The collection of soft biometrics ground truth is therefore the main limiting factor of this entire project, as it is an information that cannot always be collected directly from the thermal images by a human observer.

When it comes to the first stage of labeling, semi-supervised approaches could be used. Since the system will require a lot of data, the use of an initial dataset that would be manually labeled could be leveraged to train an object detection model. This model could then perform object recognition on unknown frame, output the coordinates of the bounding boxes with the predicted label, and a human annotator could simply correct the predictions when they are wrong. As more and more data is fed to the algorithm, it could perform better and better, automatically annotating the majority of the data with very high precision. This process could be repeated to some extent for age and gender recognition, but it would most likely be much more inaccurate and much slower, which is the entire challenge of the project, as we are trying to solve soft biometrics recognition using data that is not easily interpretable by humans for privacy reasons. A summary of the natural progression of this project is illustrated on Figure 8.3 below.

8.3.2 HARDWARE AND COMMUNICATION

As mentioned before, the Raspberry Pi 3 has shown its limitations in this project. Deep learning boards such as the Jetson Nano or the Jetson TX2 could be used in future iterations in order to take advantage of the presence of a GPU and of their higher computation capabilities. The issue of powering the boards using solar panels should be explored as well in order to move towards a sustainable implementation of the boards in the city. At each step of the

process, embedded optimizations could be found in order to reduce computation time and memory requirements, which would make it easier to power the system with a smaller solar panel, and make it more energy efficient overall.

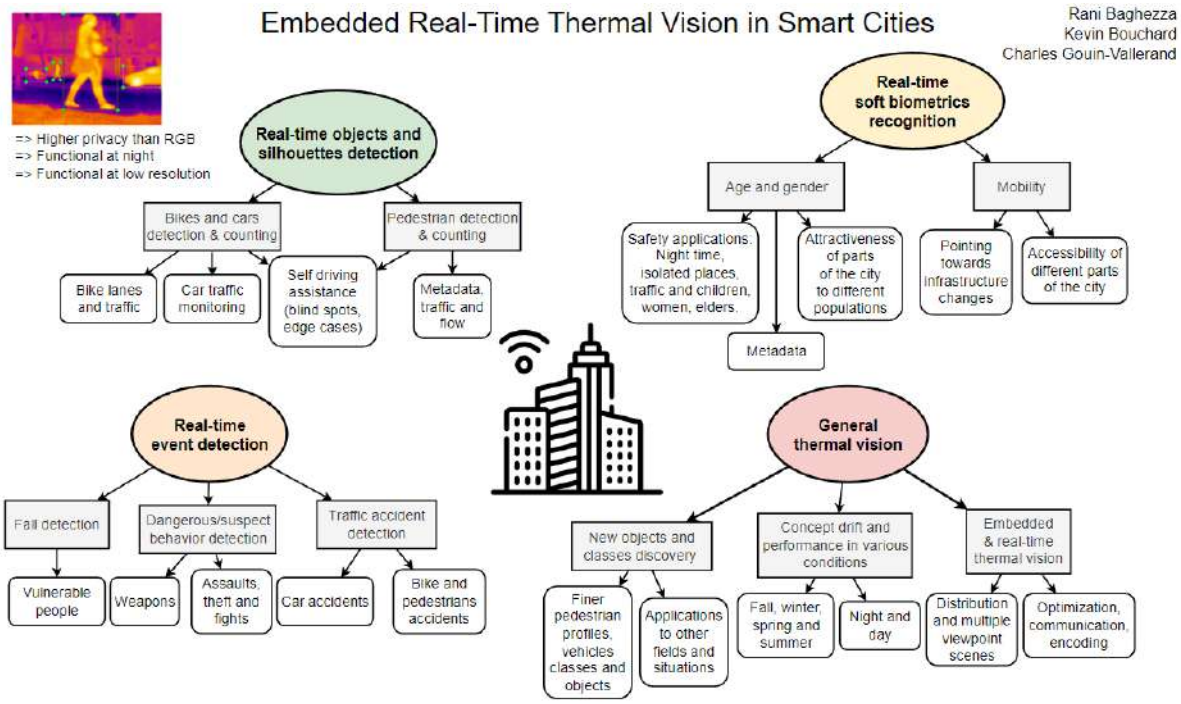

Figure 8.3 : Overview of the next steps of the project, following the thesis. © Rani Baghezza, 2022.

The sound portion of the data has been put aside for this thesis as working with images was deemed to be a more promising avenue. However, the exploration of the sound data collected by the boards for possible multi-modal profile recognition applications could be explored. The main issue coming with the use of sound data comes from the privacy concern of recording possible conversations in the streets as pedestrians walk by the node. This is why the use of sound data should be done in a privacy-preserving way, by extracting the interesting

and relevant features, and discarding the raw sound data as soon as it has been processed. The main challenge with the use of sound data comes from the ambient noise pollution in the urban environment, which could make it difficult to exploit. Moreover, the position of the node could present challenges from board to board, where the same feature extraction algorithm, the same model, or the same fusion strategy could give different results. For example a node in a big mall would record sounds of a different nature than a node positioned next to a road with heavy traffic or in an alleyway with very little traffic.

As more advanced algorithms are developed and optimized, as the prototypes are improved and as more data is collected, the communication between nodes as well as the node-server communication could be addressed. Should learning be performed on individual board, or could distributed learning be leveraged in some cases? An example of that could be the use of two nodes on either side of an area where pedestrians walk, such as a pavement or a bridge. The combination of data captured from opposite angles could lead to a better performance than the use of data from a single node. What data should be kept on the nodes, and what data should be sent to a central server for archiving and analytics purposes? How could we perform real-time re-training, and should each node be individually re-trained? What should the weight freezing strategy be, and what protocols should be followed in order to monitor the status of all of the nodes as well as their performance? How do we guarantee the performance of the nodes in a real-time environment, where no human is present to confirm the nature of the profile of a pedestrian to provide the system with the ground truth? What are the different types of concept drifts that will be encountered, and how should they be handled?

This project opens up many avenues, and a lot of questions are left to be answered. The exploration of all of these avenues would most likely allow the development of very advanced real-time thermal vision models that could be used for many applications within our cities and outside of them as well. A good example of such an application would be autonomous driving.

Recently, the use of thermal images has been explored as a way to see further and to be able to see at night, allowing the detection of pedestrians and various objects from a higher distance than what RGB cameras allow when they rely on headlights at night.

8.3.3 SMART CITIES

In smart cities, the system could be extended to many different applications beyond pedestrian profile recognition. Previous work has been done using thermal images to perform emotion recognition for drivers [227], pedestrians, bicycle and cars counting [46] as well as suspicious activity detection [228]. The problem of recognizing and extracting silhouette could be extended to a more general object recognition problem, which has been addressed for thermal images [232], [233], [234]. The city would be an ideal testbed to collect data and to explore the emergence of new classes, which could allow the exploration of semi-supervised learning approaches [229].

The collected data and the classification results could allow to gain a real-time understanding of the use of the city's infrastructure. The accessibility of different portions of the city could be measured, as well as the pedestrian traffic in certain areas. The use of thermal images could allow the system to run at night and give the opportunity to make isolated parts of the city safer. The detection of dangerous behaviors such as fights or aggression or the detection of falls in elderly could allow a quicker intervention of the authorities and medical services when needed. Beyond that, detailed statistics about the city could be collected in order to provide more appropriate services to the citizens.

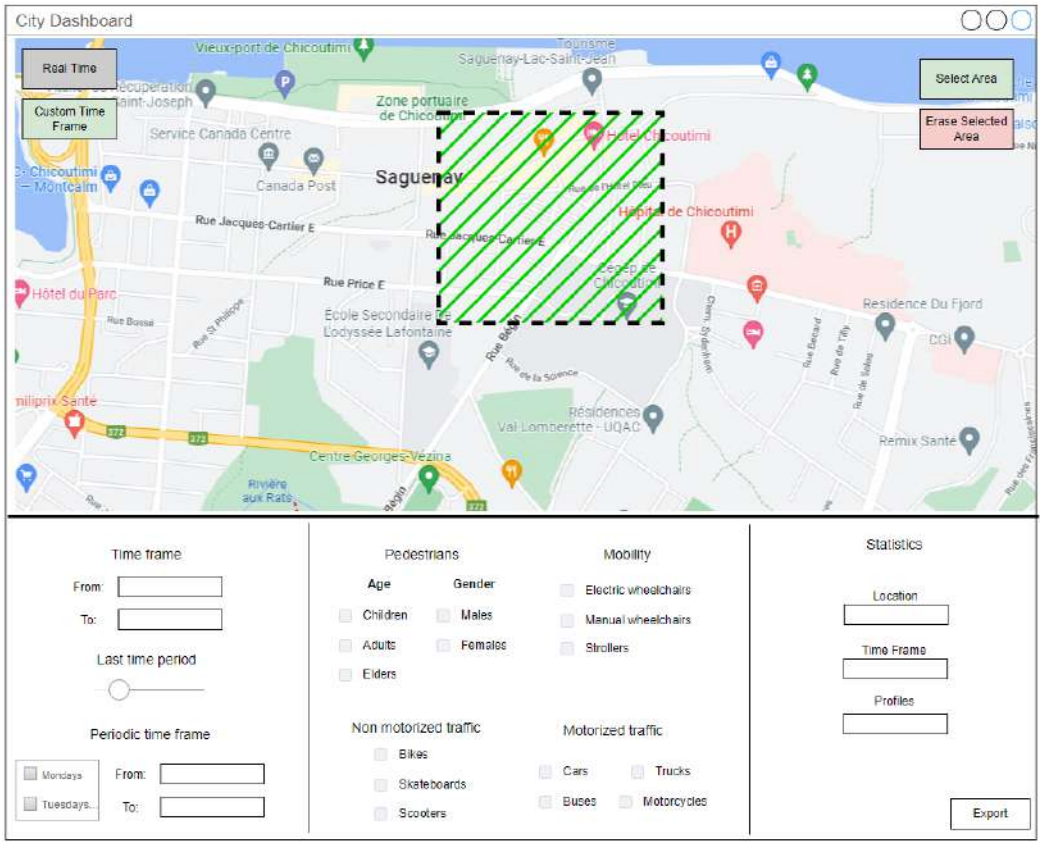

Figure 8.4 : Example of an interactive map that could be used in smart cities in order to extract knowledge from the data collected by the thermal nodes. © Rani Baghezza, 2022.

A dashboard could be created and fed with real-time data coming from a lot of nodes operating in the city, allowing to measure different metrics such as pedestrian or car traffic, possible accidents or events that require immediate intervention. The use of thermal images would guarantee a high degree of privacy for the citizens, as their facial features and details about their appearance would not be captured. Besides, since the classification results only would be transferred to the dashboard, the focus would be on the inner workings of the city and the events and distributions of profiles in real-time, rather than the individuals. Such a macroscopic, anonymous perspective on the city relying on pervasive intelligence could be a

step towards the future of smart cities. An example of a possible interactive map is shown in Figure 8.4.

As the system gets more advanced and polished, and as communication between the nodes and between the city servers intensified, the opportunity to move towards a hyper-connected city would become more and more realistic. Clusters of nodes could be combined to carry out more complex tasks by combining several point of views, tasks that would be impossible for a single node to perform. An organization of the nodes in different groups and sub-groups could lead to a system with a hierarchy, where different connected objects could be added to the network, each with their own usage in the city, such as temperature and humidity sensors, as well as sensors collecting data about the traffic, free parking spaces, opening hours of various stores and services in the city. Ultimately, the end vision could be to use these nodes, as well as other nodes performing different tasks, as the backbone of the architecture of smart cities of the future, which would allow the digitalization of modern cities, leading to more flexibility and reactivity as well as a net added value for the citizens. Interconnections between the nodes and our smartphones through city-specific apps could also be explored, moving from smart homes, to smart buildings and smart cities, to an entirely smart ecosystem interconnecting the city and its citizens, as well as smart cars and future AI-driven systems. Such a city would lead us closer to the opening quote of this thesis attributed to Nikola Tesla nearly a century ago, as well as Weiser's vision [9] of an interconnected society brought about by the use of distributed pervasive intelligence.

8.4 PERSONAL CONCLUSION

From a personal point of view, this thesis has been a very interesting, unexpected and inspiring journey. It has allowed me to destroy existing beliefs I had about research and what it should look like. Whereas my initial focus was set on getting good results to solve a

problem, I realized throughout the process of getting to know more about research that the process itself was as, if not more important than the results. Ultimately, each researcher gets to know their field, learns about the state of the art and the existing approaches, and brings their own contribution in the form of a paper. I realized that opinions could diverge within the scientific community, and that the human element was a very important factor that I had not even considered before I started submitting papers and contributing to science.

I saw the evolution of the thesis from a naive extension of activity recognition from smart homes to smart cities, to a deeper understanding and a better grasp on the problem that actually had to be solved. In order to find solutions, I learned to look at the problem at hand in a very objective way, and to let the solutions emerge and the dots connect by themselves. With enough digging in the literature, pondering, trial and error, as well as frustration, the solutions (and even more questions) started appearing by themselves.

Throughout the thesis, I realized that creativity could take on many different forms, and that the right amount of freedom and exploration combined with sufficient discipline could lead to very interesting projects being carried out and brought to completion. The time spent reviewing activity recognition approaches that initially felt like it was wasted when the direction of the thesis switched from activity recognition to gait recognition, was in fact a great introduction to the process of reviewing the existing literature in the field. I realized that we learn as much, if not more, from failure than from success, and that failure is an opportunity to course correct and achieve even better results.

I initially wished I could have done more in the field of deep learning, as the first two years of the thesis have been spent on the first and second prototype and on the collection of a suitable dataset, but I realized that I learned very valuable skills throughout this process, such as electrical engineering skills, designing a proper experimental protocol, working with

real-world data and facing real-world challenges. I used to think that theoretical research was much more important than applied research, however, this thesis has taught me that both sides of the spectrum are necessary in the world: whereas theoretical research sets the foundations, applied research makes sure that knowledge can be used and applied in the real-world for the benefit of humanity.

I would also like to thank my supervisors for their patience and their understanding of my way of working: they gave me enough room to feel free and be autonomous and they trusted my intuition and my insights, but they were also able to keep me on track and on schedule by helping me focus and prioritize on specific tasks. Their insight has been extremely valuable, and they have allowed me to quickly course correct at different times throughout the thesis. Their role has been instrumental in my choice to persevere in the world of research, and I am looking forward to the next step.

REFERENCES

[1] R. R. Schaller, "Moore's Law: Past, Present, and Future," *IEEE Spectrum*, vol. 34, no. 6, pp. 52–55, 57, 1997.

[2] G. Q. Zhang, G. Q. Zhang, Q. F. Yang, S. Q. Cheng, and T. Zhou, "Evolution of the Internet and its Cores," *New Journal of Physics*, vol. 10, 2008.

[3] K. G. Coffman and A. M. Odlyzko, "Internet Growth: Is There a "Moore's Law" for Data Traffic?" pp. 47–93, 2002.

[4] J. Chase, "Introduction The Evolution of the Internet of Things," *Texas Instruments*, vol. 1, pp. 1–7, 2020. [Online]. Available: http://www.tij.co.jp/jp/lit/ml/swrb028/swrb028.pdf

[5] L. Atzori, A. Iera, and G. Morabito, "Understanding the Internet of Things: Definition, Potentials, and Societal Role of a Fast Evolving Paradigm," *Ad Hoc Networks*, vol. 56, pp. 122–140, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.adhoc.2016.12.004

[6] T. Bojan, U. Kumar, and V. Bojan, "An Internet of Things Based Intelligent Transportation System," *2014 IEEE International Conference on Vehicular Electronics and Safety, ICVES 2014*, pp. 174–179, 2014.

[7] Y. YIN, Y. Zeng, X. Chen, and Y. Fan, "The Internet of Things in Healthcare: An Overview," *Journal of Industrial Information Integration*, vol. 1, pp. 3–13, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.jii.2016.03.004

[8] J. Wollschlaeger, M; Sauter, T; Jasperneite, "The Future of Industrial Communication: Automation Networks in the Era of the Internet of Things and Industry 4.0," *IEEE Industrial Electronics magazine*, vol. 1, no. 1, pp. 17–27, 2017.

[9] M. Weiser, "The Computer for the 21st Century," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 3, no. 3, pp. 3–11, 1999.

[10] F. John Dian, R. Vahidnia, and A. Rahmati, "Wearables and the Internet of Things (IoT), Applications, Opportunities, and Challenges: A Survey," *IEEE Access*, vol. 8, pp. 69 200–69 211, 2020.

[11] J. McCarthy, "What is Artificial Intelligence?" Tech. Rep., 2004.

[12] A. Bahrammirzaee, "A Comparative Survey of Artificial Intelligence Applications in Finance: Artificial Neural Networks, Expert System and Hybrid Intelligent Systems," *Neural Computing and Applications*, vol. 19, no. 8, pp. 1165–1195, 2010.

[13] K. H. Yu, A. L. Beam, and I. S. Kohane, "Artificial Intelligence in Healthcare," *Nature Biomedical Engineering*, vol. 2, no. 10, pp. 719–731, 2018. [Online]. Available: http://dx.doi.org/10.1038/s41551-018-0305-z

[14] J. hua Li, "Cyber Security Meets Artificial Intelligence: a Survey," *Frontiers of Information Technology and Electronic Engineering*, vol. 19, no. 12, pp. 1462–1474, 2018.

[15] L. Sijing and W. Lan, "Artificial Intelligence Education Ethical Problems and Solutions," *13th International Conference on Computer Science and Education, ICCSE 2018*, no. Iccse, pp. 155–158, 2018.

[16] P. Svenmarck, L. Luotsinen, M. Nilsson, and J. Schubert, "Possibilities and Challenges for Artificial Intelligence in Military Applications," in *Proceedings of the 2018 NATO Big Data and Artificial Intelligence for Military Decision Making Specialists*, 2018.

[17] J. Kietzmann, J. Paschen, and E. Treen, "Artificial Intelligence in Advertising: How Marketers can Leverage Artificial Intelligence Along the Consumer Journey," *Journal of Advertising Research*, vol. 58, no. 3, pp. 263–267, 2018.

[18] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, "Data Mining: Practical Machine Learning Tools and Techniques," *Data Mining: Practical Machine Learning Tools and Techniques*, no. October 1999, pp. 1–621, 2016.

[19] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[20] F. C. Delicato, A. Al-Anbuky, and K. I. Wang, "Editorial: Smart Cyber–Physical Systems: Toward Pervasive Intelligence Systems," *Future Generation Computer Systems*, vol. 107, pp. 1134–1139, 2020. [Online]. Available: https://doi.org/10.1016/j.future.2019.06.031

[21] H. Serra, R. Francisco, A. Arsénio, F. Nabais, J. Andrade, and E. Serrano, "Internet of Intelligent Things: Bringing Artificial Intelligence into Things and Communication Networks," *Studies in Computational Intelligence*, vol. 495, pp. 1–37, 2014.

[22] R. Baghezza, K. Bouchard, A. Bouzouane, and C. Gouin-Vallerand, "From Offline to Real-Time Distributed Activity Recognition in Wireless Sensor Networks for Healthcare: A Review," *Sensors*, vol. 21, no. 8, pp. 1–34, 2021.

[23] S. Yi, C. Li, and Q. Li, "A Survey of Fog Computing: Concepts, Applications and Issues," *Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, vol. 2015-June, pp. 37–42, 2015.

[24] K. Cao, Y. Liu, G. Meng, and Q. Sun, "An Overview on Edge Computing Research," *IEEE Access*, vol. 8, pp. 85 714–85 728, 2020.

[25] J. J. Rodrigues, D. B. De Rezende Segundo, H. A. Junqueira, M. H. Sabino, R. M. I. Prince, J. Al-Muhtadi, and V. H. C. De Albuquerque, "Enabling Technologies for the Internet of Health Things," *IEEE Access*, vol. 6, no. January, pp. 13 129–13 141, 2018.

[26] M. R. Alam, M. B. I. Reaz, and M. A. M. Ali, "A Review of Smart Homes - Past, present, and Future," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 42, no. 6, pp. 1190–1203, 2012.

[27] United Nations, *World Population Ageing, 1950-2050*, 2019, vol. 40, no. 03.

[28] A. Alzheimer, "2019 Alzheimer's Disease Facts and Figures," *The Journal of Alzheimer's Association*, vol. 15, pp. 321–387, 2019.

[29] S. Katz, "Assessing Delf-Maintenance: Activities of Daily Living, Mobility, and Instrumental Activities of Daily Living," *Journal of the American Geriatrics Society*, vol. 31, no. 12, pp. 721–727, 1983.

[30] G. Demiris, M. J. Rantz, M. A. Aud, K. D. Marek, H. W. Tyrer, M. Skubic, and A. A. Hussam, "Older Adults' Attitudes Towards and Perceptions of 'Smart Home' Technologies: A Pilot study," *Medical Informatics and the Internet in Medicine*, vol. 29, no. 2, pp. 87–94, 2004.

[31] E. Nazerfard, P. Rashidi, and D. J. Cook, "Using Association Rule Mining to Discover Temporal Eelations of Daily Activities," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6719 LNCS, pp. 49–56, 2011.

[32] D. J. Cook, A. S. Crandall, B. L. Thomas, and N. C. Krishnan, "CASAS: A Smart Home in a Box," *Computer*, vol. 135, no. 2, pp. 62–69, 2013. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3886862/pdf/nihms-495602.pdf/?tool=EBI

[33] A. Benmansour, A. Bouchachia, and M. Feham, "Multioccupant Activity Recognition in Pervasive Smart Home Environments," *ACM Computing Surveys*, vol. 48, no. 3, pp. 1–36, 2015. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2856149.2835372

[34] G. Fortino, A. Giordano, A. Guerrieri, G. Spezzano, and A. Vinci, "A Data Analytics Schema for Activity Recognition in Smart Home Environments," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9454, pp. 91–102, 2015.

[35] T. hoon Kim, C. Ramos, and S. Mohammed, "Smart City and IoT," *Future Generation Computer Systems*, vol. 76, no. July 2014, pp. 159–162, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.future.2017.03.034

[36] F. Leccese, M. Cagnetti, and D. Trinca, "A Smart City Application: A Fully Controlled Street Lighting Isle Based on Raspberry-Pi Card, a ZigBee Sensor Network and WiMAX," *Sensors (Switzerland)*, vol. 14, no. 12, pp. 24 408–24 424, 2014.

[37] M. A. Kafi, Y. Challal, D. Djenouri, M. Doudou, A. Bouabdallah, and N. Badache, "A Study of Wireless Sensor Networks for Urban Traffic Monitoring: Applications and Architectures," *Procedia Computer Science*, vol. 19, no. Ant, pp. 617–626, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.procs.2013.06.082

[38] A. Bagula, L. Castelli, and M. Zennaro, "On the Design of Smart Parking Networks in the Smart Cities: An Optimal Sensor Placement Model," *Sensors (Switzerland)*, vol. 15, no. 7, pp. 15 443–15 467, 2015.

[39] V. Kalyuzhnyy, M. Costa, P. Silva, and J. Santos, "Smart City IoT System - Collect-MyWaste," *4th International Symposium on Multidisciplinary Studies and Innovative*

Technologies, ISMSIT 2020 - Proceedings, vol. c, pp. 1–5, 2020.

[40] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for Smart Cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, 2014.

[41] J. Bricout, P. M. Baker, N. W. Moon, and B. Sharma, "Exploring the Smart Future of Participation: Community, Inclusivity, and People with Disabilities," *International Journal of E-Planning Research*, vol. 10, no. 2, pp. 94–108, 2021.

[42] M. Boukhechba, A. Bouzouane, B. Bouchard, C. Gouin-Vallerand, and S. Giroux, "Online Recognition of People's Activities from Raw GPS Data," in *Proceedings of the 8th ACM International Conference on Pervasive Technologies Related to Assistive Environments*, 2015, pp. 1–8.

[43] J. Yao, J. Yang, and D. Otte, "Head Injuries in Child Pedestrian Accidents - In-Depth Case Analysis and Reconstructions," *Traffic Injury Prevention*, vol. 8, no. 1, pp. 94–100, 2007.

[44] P. Kannus, J. Parkkari, S. Niemi, and M. Palvanen, "Fall-Induced Deaths Among Elderly People," *American Journal of Public Health*, vol. 95, no. 3, pp. 422–424, 2005.

[45] M. McCahill and C. Norris, "On the Threshold to Urban Panopticon? Analysing the Employment of Cctv in European Cities Assessing Its Social Political Impacts," *On the Threshold to Urban Panopticon? Analysing the Employment of Cctv in European Cities Assessing Its Social Political Impacts*, no. June, 2002. [Online]. Available: http://ezp-prod1.hul.harvard.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=cja&AN=CJA0310010000423&site=ehost-live&scope=site

[46] R. Gade, T. B. Moeslund, S. Z. Nielsen, H. Skov-Petersen, H. J. Andersen, K. Basselbjerg, H. T. Dam, O. B. Jensen, A. Jørgensen, H. Lahrmann, T. K. O. Madsen, E. S. Bala, and B. Povey, "Thermal Imaging Systems for Real-Time Applications in Smart Cities," *International Journal of Computer Applications in Technology*, vol. 53, no. 4, pp. 291–308, 2016.

[47] B. J. Goold, "Privacy Rights and Public Spaces: CCTV and the Problem of the "Unobservable Observer"," *Criminal Justice Ethics*, vol. 21, no. 1, pp. 21–27, 2002.

[48] S. Sarkar and L. Zongyi, "Gait Recognition," in *Encyclopedia of Cryptography and Security*, H. C. A. Van Tilborg and S. Jajodia, Eds. Boston, MA: Springer US, 2011, pp. 503–506.

[49] S. Yu, D. Tan, and T. Tan, "A Framework for Evaluating the Effect of View Angle, Clothing and Carrying Condition on Gait Recognition," *Proceedings - International Conference on Pattern Recognition*, vol. 4, pp. 441–444, 2006.

[50] H. Iwama, M. Okumura, Y. Makihara, and Y. Yagi, "The OU-ISIR Gait Database Comprising the Large Population Dataset and Performance Evaluation of Gait Recognition," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 5, pp. 1511–1521, 2012.

[51] K. Bashir, T. Xiang, and S. Gong, "Gait Recognition Without Subject Cooperation," *Pattern Recognition Letters*, vol. 31, no. 13, pp. 2052–2060, 2010. [Online]. Available: http://dx.doi.org/10.1016/j.patrec.2010.05.027

[52] M. Nabila, A. I. Mohammed, and B. J. Yousra, "Gait-Based Human Age Classification using a Silhouette Model," *IET Biometrics*, vol. 7, no. 2, pp. 116–124, 2018.

[53] R. Baghezza, K. Bouchard, A. Bouzouane, and C. Gouin-Vallerand, "Activity Recognition in the City using Embedded Systems and Anonymous Sensors," *Procedia Computer Science*, vol. 170, no. 2019, pp. 67–74, 2020. [Online]. Available: https://doi.org/10.1016/j.procs.2020.03.140

[54] R. Baghezza, K. Bouchard, C. Gouin-Vallerand, and A. Bouzouane, "Profile Recognition for Accessibility and Inclusivity in Smart Cities using a Thermal Imaging Sensor in an Embedded System," *IEEE Internet of Things Journal*, 2021.

[55] P. Gogas, T. Papadimitriou, and A. Agrapetidou, "Forecasting Bank Failures and Stress Testing: A Machine Learning Approach," *SSRN Electronic Journal*, no. Mis 380292, pp. 1–31, 2017.

[56] N. Sharma, P. Sharma, D. Irwin, and P. Shenoy, "Predicting Solar Generation from Weather Forecasts using Machine Learning," *2011 IEEE International Conference on Smart Grid Communications, SmartGridComm 2011*, pp. 528–533, 2011.

[57] I. Habernal, T. Ptáček, and J. Steinberger, "Sentiment Analysis in Social Media using

Machine Learning Techniques," in *Proceedings of the 4th workshop on computational approaches to subjectivity, sentiment and social media analysis*, 2013, pp. 65–74.

[58] K. Alsabti, S. Ranka, and V. Singh, "An Efficient k-Means Clustering Algorithm," *Electrical Engineering and Computer Science*, 1997.

[59] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "Density-Based Clustering Methods," in *KDD-96*, 1996.

[60] S. C. Johnson, "Hierarchical Clustering Schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967.

[61] A. Ng, "Sparse Autoencoder," Tech. Rep., 2011.

[62] T. Kohonen, S. Kaski, P. Somervuo, K. Lagus, and M. Oja, "The Self-Organizing Map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[63] X. Zhu and A. B. Goldberg, *Introduction to Semi-Supervised Learning*, 2009, vol. 6.

[64] F. Lotte, "Signal Processing Approaches to Minimize or Suppress Calibration Time in Oscillatory Activity-Based Brain-Computer Interfaces," *Proceedings of the IEEE*, vol. 103, no. 6, pp. 871–890, 2015.

[65] R. Sousa and J. Gama, "Comparison Between Co-Training and Self-Training for Single-Target Regression in Data Streams using AMRules," *CEUR Workshop Proceedings*, vol. 1958, 2017.

[66] B. Settles, "Active Learning Literature Survey," University of Wisconsin, Computer Science Department, Tech. Rep. January, 2009.

[67] I. H. Witten and E. Frank, "Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations," vol. 31, p. 371, 1999. [Online]. Available: http://www.amazon.com/Data-Mining-Techniques-Implementations-Management/dp/1558605525

[68] G. Hripcsak and A. S. Rothschild, "Agreement, the F-measure, and Reliability in Informa-

tion Retrieval," *Journal of the American Medical Informatics Association*, vol. 12, no. 3, pp. 296–298, 2005.

[69] A. B. Cantor, "Sample-Size Calculations for Cohen's Kappa," *Psychological Methods*, vol. l, pp. 150–153, 1996.

[70] S. Amari and S. Wu, "Improving Support Vector Machine Classifiers by Modifying Kernel Functions," *Neural Networks*, vol. 12, no. 6, pp. 783–789, 1999.

[71] S. Sharma, S. Sharma, and A. Athaiya, "Activation Functions in Neural Networks," *International Journal of Engineering Applied Sciences and Technology*, vol. 04, no. 12, pp. 310–316, 2020.

[72] G. Sanderson, "Neural Networks," 2017. [Online]. Available: https://www.3blue1brown.com/neural-networks

[73] L. Bottou, "Large-Scale Machine Learning with Stochastic Gradient Descent," *Proceedings of COMPSTAT'2010*, pp. 177–186, 2010.

[74] S. Lawrence, C. Giles, A. Tsoi, and A. Back, "Face Recognition: A Convolutional Neural-Network Approach," *IEEE transactions on neural networks*, vol. 8, no. 1, pp. 98–113, 1997.

[75] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A Convolutional Neural Network for Modelling Sentences," *52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference*, vol. 1, pp. 655–665, 2014.

[76] D. Maturana and S. Scherer, "VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2015-Decem, pp. 922–928, 2015.

[77] G. Tolias, R. Sicre, and H. Jégou, "Particular Object Retrieval with Integral Max-Pooling of CNN Activations," *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, pp. 1–12, 2016.

[78] M. Auli, M. Galley, C. Quirk, and G. Zweig, "Joint Language and Translation Modeling with Recurrent Neural Networks," *EMNLP 2013 - 2013 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, no. October, pp. 1044–1054, 2013.

[79] D. Liu and L. Lin, "Memory-Based Gait Recognition," in *The British Machine Vision Conference*, York, UK, 2016, pp. 1–12.

[80] K. Jun, D. W. Lee, K. Lee, S. Lee, and M. S. Kim, "Feature Extraction using an RNN Autoencoder for Skeleton-Based Abnormal Gait Recognition," *IEEE Access*, vol. 8, pp. 19 196–19 207, 2020.

[81] P. J. Werbos, "Backpropagation Through Time: What it Does and How to do it," in *Proceedings of the IEEE*, 1990, pp. 1550–1560.

[82] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," in *Routledge Library Editions: Linguistics Mini-Set A General Linguistics*, 2013, vol. 2-11, no. 8, pp. 13–35.

[83] A. Graves, A. R. Mohamed, and G. Hinton, "Speech Recognition with Deep Recurrent Neural Networks," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, no. 3, pp. 6645–6649, 2013.

[84] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," pp. 1–9, 2014. [Online]. Available: http://arxiv.org/abs/1412.3555

[85] M. W. Whittle, *Gait Analysis*. Butterworth-Heinemann Ltd, 1993, vol. 4, no. 1. [Online]. Available: http://dx.doi.org/10.1016/B978-0-7506-0170-2.50017-0

[86] B. Pogorelc, Z. Bosnić, and M. Gams, "Automatic Recognition of Gait-Related Health Problems in the Elderly using Machine Learning," *Multimedia Tools and Applications*, vol. 58, no. 2, pp. 333–354, 2012.

[87] N. V. Boulgouris, D. Hatzinakos, and K. N. Plataniotis, "Gait recognition: A challening Signal Processing Technology for Biometric Identification," *IEEE Signal Processing Magazine*, vol. 22, no. 6, pp. 78–90, 2005.

[88] I. Bouchrika, "A Survey of using Biometrics for Smart Visual Surveillance: Gait Recognition," *Advanced Sciences and Technologies for Security Applications*, pp. 3–23, 2018.

[89] O. Elharrouss, N. Almaadeed, S. Al-Maadeed, and A. Bouridane, "Gait Recognition for Person Re-Identification," *Journal of Supercomputing*, vol. 77, no. 4, pp. 3653–3672, 2021. [Online]. Available: https://doi.org/10.1007/s11227-020-03409-5

[90] M. Hofmann, J. Geiger, S. Bachmann, B. Schuller, and G. Rigoll, "The TUM Gait from Audio, Image and Depth (GAID) Database: Multimodal Recognition of Subjects and Traits," *Journal of Visual Communication and Image Representation*, vol. 25, no. 1, pp. 195–206, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.jvcir.2013.02.006

[91] N. R. Baek, S. W. Cho, J. H. Koo, N. Q. Truong, and K. R. Park, "Multimodal Camera-Based Gender Recognition using Human-Body Image with Two-Step Reconstruction Network," *IEEE Access*, vol. 7, pp. 104 025–104 044, 2019.

[92] K. Lenac, D. Sušanj, A. Ramakić, and D. Pinčić, "Extending Appearance Based Gait Recognition with Depth Data," *Applied Sciences (Switzerland)*, vol. 9, no. 24, 2019.

[93] G. Huang and W. Yunhong, "Gender Classification Based on Fusion of Multi-View Gait Sequences," in *Conference on Computer Vision*. Springer Berlin Heidelberg, 2007, pp. 462–471.

[94] Z. Lu, Y. Xu, Z. Dai, and B. Ma, "A Gait Recognition Based on Link Model of Infrared Thermal Imaging," *Proceedings of 2016 2nd International Conference on Control Science and Systems Engineering, ICCSSE 2016*, pp. 165–168, 2016.

[95] Y. Makihara, M. Okumura, H. Iwama, and Y. Yagi, "Gait-based Age Estimation using a Whole-Generation Gait Database," *2011 International Joint Conference on Biometrics, IJCB 2011*, 2011.

[96] E. Cippitelli, S. Gasparrini, S. Spinsante, and E. Gambi, "Kinect as a Tool for Gait Analysis: Validation of a Real-Time Joint Extraction Algorithm Working in Side View," *Sensors (Switzerland)*, vol. 15, no. 1, pp. 1417–1434, 2015.

[97] J. Snoek, J. Hoey, L. Stewart, R. S. Zemel, and A. Mihailidis, "Automated Detection of

Unusual Events on Stairs," *Image and Vision Computing*, vol. 27, no. 1-2, pp. 153–166, 2009. [Online]. Available: http://dx.doi.org/10.1016/j.imavis.2008.04.021

[98] N. Dalal, B. Triggs, and S. Diego, "Histograms of Oriented Gradients for Human Detection," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 886–893, 2010.

[99] A. Suleiman, Y.-h. Chen, J. Emer, and V. Sze, "Towards Closing the Energy Gap Between HOG and CNN Features for Embedded Vision," in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2017, pp. 1–4.

[100] J. Han and B. Bhanu, "Individual Recognition using Gait Energy Image," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 2, pp. 316–322, 2006.

[101] M. J. Black and P. Anandan, "The Robust Estimation of Multiple Motions: Parametric and Piecewise-Smooth Flow Fields," *Computer Vision and Image Understanding*, vol. 63, no. 1, pp. 75–104, 1996.

[102] Z. Mahfouf, H. F. Merouani, I. Bouchrika, and N. Harrati, "Investigating the Use of Motion-Based Features from Optical Flow for Gait Recognition," *Neurocomputing*, vol. 283, pp. 140–149, 2018. [Online]. Available: https://doi.org/10.1016/j.neucom.2017.12.040

[103] Y. Makihara, R. Sagawa, and Y. Mukaigawa, "Gait Recognition using a View Transformation in the Frequency Domain," in *European conference on computer vision*. Springer Berlin Heidelberg, 2006, pp. 151–163.

[104] D. P. Tian, "A Review on Image Feature Extraction and Representation Techniques," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 8, no. 4, pp. 385–395, 2013.

[105] C. Wan, L. Wang, and V. V. Phoha, "A Survey on Gait Recognition," *ACM Computing Surveys*, vol. 51, no. 5, 2018.

[106] D. Gafurov, E. Snekkenes, and P. Bours, "Gait Authentication and Identification using Wearable Accelerometer Sensor," *2007 IEEE Workshop on Automatic Identification Advanced Technologies - Proceedings*, pp. 220–225, 2007.

[107] D. J. Berndt, "Using Dynamic Time Warping to Find Patterns in Time Series," AAAI, Tech. Rep., 1994.

[108] S. Eddy, "What is a hidden Markov model?" *Nature biotechnology*, pp. 1–5, 2004. [Online]. Available: https://scholar.google.com/citations?view_op=view_citation&continue=/scholar%3Fhl%3Den%26as_sdt%3D0,14%26scilib%3D1&citilm=1&citation_for_view=NzNAerUAAAAJ:_FxGoFyzp5QC&hl=en&oi=p

[109] T. K. Bajwa, S. Garg, and K. Saurabh, "GAIT Analysis for Identification by using SVM with K-NN and NN Techniques," *2016 4th International Conference on Parallel, Distributed and Grid Computing, PDGC 2016*, pp. 259–263, 2016.

[110] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[111] A. Sepas-Moghaddam and A. Etemad, "Deep Gait Recognition: A Survey," pp. 1–19, 2021. [Online]. Available: http://arxiv.org/abs/2102.09546

[112] J. D. Shutler, M. G. Grant, M. S. Nixon, and J. N. Carter, "On a Large Sequence-Based Human Gait Database," *Applications and Science in Soft Computing*, pp. 339–346, 2004.

[113] L. Wang, T. Tan, H. Ning, and W. Hu, "Silhouette Analysis-Based Gait Recognition for Human Identification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, pp. 1505–1518, 2003.

[114] S. Sarkar, P. J. Phillips, Z. Liu, I. R. Vega, P. Grother, and K. W. Bowyer, "The humanID Gait Challenge Problem: Data sets, Performance, and Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 2, pp. 162–177, 2005.

[115] M. Hofmann, J. Geiger, S. Bachmann, B. Schuller, and G. Rigoll, "The TUM Gait from Audio, Image and Depth (GAID) Database: Multimodal Recognition of Subjects and Traits," *Journal of Visual Communication and Image Representation*, vol. 25, no. 1, pp. 195–206, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.jvcir.2013.02.006

[116] R. A. Güler, N. Neverova, and I. Kokkinos, "DensePose: Dense Human Pose Estimation in

the Wild," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 7297–7306, 2018.

[117] Y. Wang, J. Sun, J. Li, and D. Zhao, "Gait Recognition Based on 3D Skeleton Joints Captured by Kinect," *Proceedings - International Conference on Image Processing, ICIP*, vol. 2016-Augus, pp. 3151–3155, 2016.

[118] C. Wang, J. Zhang, J. Pu, X. Yuan, and L. Wang, "Chrono-Gait Image: A Novel Temporal Template for gait recognition," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6311 LNCS, no. PART 1, pp. 257–270, 2010.

[119] C. Chen, J. Liang, H. Zhao, H. Hu, and J. Tian, "Frame Difference Energy Image for Gait Recognition with Incomplete Silhouettes," *Pattern Recognition Letters*, vol. 30, no. 11, pp. 977–984, 2009. [Online]. Available: http://dx.doi.org/10.1016/j.patrec.2009.04.012

[120] K. Bashir, T. Xiang, and S. Gong, "Gait Recognition using Gait Entropy Image," *IET Seminar Digest*, vol. 2009, no. 2, 2009.

[121] M. A. Martín and J. M. Rey, "On the Role of Shannon's Entropy as a Measure of Heterogeneity," *Geoderma*, vol. 98, no. 1-2, pp. 1–3, 2000.

[122] Y. He, J. Zhang, H. Shan, and L. Wang, "Multi-Task GANs for View-Specific Feature Learning in Gait Recognition," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 1, pp. 102–113, 2019.

[123] R. Liao, C. Cao, E. B. Garcia, S. Yu, and Y. Huang, "Pose-Based Temporal-Spatial Network (PTSN) for Gait Recognition with Carrying and Clothing Variations," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10568 LNCS, pp. 474–483, 2017.

[124] B. Lin, S. Zhang, and F. Bao, "Gait Recognition with Multiple-Temporal-Scale 3D Convolutional Neural Network," *MM 2020 - Proceedings of the 28th ACM International Conference on Multimedia*, vol. 1, no. c, pp. 3054–3062, 2020.

[125] N. Li, X. Zhao, and C. Ma, "JointsGait: A model-based Gait Recognition Method based on

Gait Graph Convolutional Networks and Joints Relationship Pyramid Mapping," no. Na Li, 2020. [Online]. Available: http://arxiv.org/abs/2005.08625

[126] A. Sepas-Moghaddam and A. Etemad, "View-Invariant Gait Recognition With Attentive Recurrent Learning of Partial Representations," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 3, no. 1, pp. 124–137, 2020.

[127] I. Rida, "Towards Human Body-Part Learning for Model-Free Gait Recognition," 2019. [Online]. Available: http://arxiv.org/abs/1904.01620

[128] A. Sepas-Moghaddam, S. Ghorbani, N. F. Troje, and A. Etemad, "Gait Recognition using Multi-Scale Partial Representation Transformation with Capsules," *Proceedings - International Conference on Pattern Recognition*, pp. 8045–8052, 2020.

[129] H. Chao, Y. He, J. Zhang, and J. Feng, "GaitSet: Regarding Gait as a Set for Cross-View Gait Recognition," *33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019*, no. 16, pp. 8126–8133, 2019.

[130] Y. Feng, Y. Li, and J. Luo, "Learning Effective Gait Features using LSTM," *Proceedings - International Conference on Pattern Recognition*, vol. 0, pp. 325–330, 2016.

[131] S. Lange and M. Riedmiller, "Deep Auto-Encoder Neural Networks in Reinforcement Learning," in *The 2010 International Joint Conference on Neural Networks (IJCNN)*, 2010, pp. 1–8.

[132] S. Yu, H. Chen, Q. Wang, L. Shen, and Y. Huang, "Invariant Feature Extraction for Gait Recognition using Only One Uniform Model," *Neurocomputing*, vol. 239, pp. 81–93, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.neucom.2017.02.006

[133] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Neural Computation*, vol. 18, pp. 1527–1554, 2006.

[134] M. Benouis, M. Senouci, R. Tlemsani, and L. Mostefai, "Gait Recognition Based on Model-Based Methods and Deep Belief Networks," *International Journal of Biometrics*,

vol. 8, no. 3-4, pp. 237–253, 2016.

[135] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[136] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[137] S. Yu, R. Liao, W. An, H. Chen, E. B. García, Y. Huang, and N. Poh, "GaitGANv2: Invariant Gait Feature Extraction using Generative Adversarial Networks," *Pattern Recognition*, vol. 87, pp. 179–189, 2019.

[138] Y. He, J. Zhang, H. Shan, and L. Wang, "Multi-Task GANs for View-Specific Feature Learning in Gait Recognition," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 1, pp. 102–113, 2019.

[139] X. Li, Y. Makihara, C. Xu, Y. Yagi, and M. Ren, "Gait Recognition Invariant to Carried Objects using Alpha Blending Generative Adversarial Networks," *Pattern Recognition*, vol. 105, p. 107376, 2020. [Online]. Available: https://doi.org/10.1016/j.patcog.2020.107376

[140] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic Routing Between Capsules," *Advances in Neural Information Processing Systems*, vol. 2017-Decem, no. Nips, pp. 3857–3867, 2017.

[141] C. Xu, Y. Makihara, Y. Yagi, and J. Lu, "Gait-Based Age Progression/Regression: a Baseline and Performance Evaluation by Age Group Classification and Cross-Age Gait Identification," *Machine Vision and Applications*, vol. 30, no. 4, pp. 629–644, 2019. [Online]. Available: https://doi.org/10.1007/s00138-019-01015-x

[142] G. Batchuluun, H. S. Yoon, J. K. Kang, and K. R. Park, "Gait-Based Human Identification by Combining Shallow Convolutional Neural Network-Stacked Long Short-Term Memory and Deep Convolutional Neural Network," *IEEE Access*, vol. 6, no. c, pp. 63 164–63 186, 2018.

[143] Y. Zhang, Y. Huang, S. Yu, and L. Wang, "Cross-View Gait Recognition by Discriminative Feature Learning," *IEEE Transactions on Image Processing*, vol. 29, pp. 1001–1015, 2020.

[144] Z. Zhang, L. Tran, F. Liu, and X. Liu, "On Learning Disentangled Representations for Gait Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.

[145] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," *Proceedings of SSST 2014 - 8th Workshop on Syntax, Semantics and Structure in Statistical Translation*, pp. 103–111, 2014.

[146] J. Verghese, A. LeValley, C. B. Hall, M. J. Katz, A. F. Ambrose, and R. B. Lipton, "Epidemiology of Gait Disorders in Community-Residing Older Adults," *Journal of the American Geriatrics Society*, vol. 54, no. 2, pp. 255–261, 2006.

[147] M. Montero-Odasso, M. Schapira, E. R. Soriano, M. Varela, R. Kaplan, L. A. Camera, and L. M. Mayorga, "Gait Velocity as a Single Predictor of Adverse Events in Healthy Seniors Aged 75 Years and Older," *Journals of Gerontology - Series A Biological Sciences and Medical Sciences*, vol. 60, no. 10, pp. 1304–1309, 2005.

[148] X. Geng, Z. H. Zhou, and K. Smith-Miles, "Automatic Age Estimation Based on Facial Aging Patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2234–2240, 2007.

[149] R. Zazo, P. Sankar Nidadavolu, N. Chen, J. Gonzalez-Rodriguez, and N. Dehak, "Age Estimation in Short Speech Utterances Based on LSTM Recurrent Neural Networks," *IEEE Access*, vol. 6, pp. 22 524–22 530, 2018.

[150] T. B. Aderinola, T. Connie, T. S. Ong, W. C. Yau, and A. B. J. Teoh, "Learning Age from Gait: A Survey," *IEEE Access*, vol. 9, no. July, pp. 100 352–100 368, 2021.

[151] M. L. Callisaya, L. Blizzard, M. D. Schmidt, J. L. McGinley, and V. K. Srikanth, "Sex Modifies the Relationship Between Age and Gait: A Population-Based Study of Older Adults," *Journals of Gerontology - Series A Biological Sciences and Medical Sciences*, vol. 63, no. 2, pp. 165–170, 2008.

[152] Y. Makihara, H. Mannami, and Y. Yagi, "Gait Analysis of Gender and Age using a Large-Scale Multi-View Gait Database," in *Asian Conference on Computer Vision*, vol. 6493, 2011, pp. 440–451.

[153] C. Xu, Y. Makihara, G. Ogi, X. Li, Y. Yagi, and J. Lu, "The OU-ISIR Gait Database Comprising the Large Population Dataset with Age and Performance Evaluation of Age Estimation," *IPSJ Transactions on Computer Vision and Applications*, vol. 9, no. 1, 2017.

[154] N. Takemura, Y. Makihara, D. Muramatsu, T. Echigo, and Y. Yagi, "Multi-View Large Population Gait Dataset and its Performance Evaluation for Cross-View Gait Recognition," *IPSJ Transactions on Computer Vision and Applications*, vol. 10, no. 1, 2018.

[155] H. Dou, W. Zhang, P. Zhang, Y. Zhao, S. Li, Z. Qin, F. Wu, L. Dong, and X. Li, "VersatileGait: A Large-Scale Synthetic Gait Dataset with Fine-GrainedAttributes and Complicated Scenarios," 2021. [Online]. Available: http://arxiv.org/abs/2101.01394

[156] F. M. Castro, M. J. J. Marín-Jiménez, and N. Guil, "Multimodal Features Fusion for Gait, Gender and Shoes Recognition," *Machine Vision and Applications*, vol. 27, no. 8, pp. 1213–1228, 2016.

[157] F. Prince, H. Corriveau, R. Hébert, and D. A. Winter, "Gait in the Elderly," *Gait and Posture*, vol. 5, no. 2, pp. 128–135, 1997.

[158] J. W. Davis, "Visual Categorization of Children and Adult Walking Styles," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2091 LNCS, no. June, pp. 295–300, 2001.

[159] D. Zhang, Y. Wang, and B. Bhanu, "Age Classification Based on Gait using HMM," *Proceedings - International Conference on Pattern Recognition*, pp. 3834–3837, 2010.

[160] M. Hu, Y. Wang, and Z. Zhang, "Maximisation of Mutual Information for Gait-Based Soft Biometric Classification using Gabor Features," *IET Biometrics*, vol. 1, no. 1, pp. 55–62, 2012.

[161] V. Krüger and G. Sommer, "Gabor Wavelet Networks for Object Representation," pp. 309–316, 2000.

[162] R. Mehrotra, K. Namuduri, and A. Pellegrino, "Gabor Filter-Based Edge Detection," *Pattern recognition*, vol. 25, no. 12, pp. 1479–1494, 1992.

[163] T. P. Weldon, W. E. Higgins, and D. F. Dunn, "Efficient Filter Design for Texture Segmentation," *Image Processing*, pp. 1–17, 1996.

[164] S. Allagwail, O. S. Gedik, and J. Rahebi, "Face Recognition with Symmetrical Face Training Samples Based on Local Binary Patterns and the Gabor Filter," *Symmetry*, vol. 11, no. 2, 2019.

[165] A. Nurhadiyatna, A. L. Latifah, and D. Fryantoni, "Gabor Filtering for Feature Extraction in Real Time Vehicle Classification System," *9th International Symposium on Image and Signal Processing and Analysis, ISPA 2015*, no. September, pp. 19–24, 2015.

[166] E. Avineri, D. Shinar, and Y. O. Susilo, "Pedestrians' Behaviour in Cross Walks: The Effects of Fear of Falling and Age," *Accident Analysis and Prevention*, vol. 44, no. 1, pp. 30–34, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.aap.2010.11.028

[167] J. Lu and Y. P. Tan, "Gait-Based Human Age Estimation," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 4, pp. 761–770, 2010.

[168] M. L. Zhang and Z. H. Zhou, "ML-KNN: A Lazy Learning Approach to Multi-Label Learning," *Pattern Recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.

[169] C. E. Rasmussen, "Gaussian Processes in Machine Learning," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3176, pp. 63–71, 2004.

[170] J. C. Platt, N. Cristianini, and J. Shawe-Taylor, "Large Margin DAGs for Multiclass Classification," *Advances in Neural Information Processing Systems*, pp. 547–553, 2000.

[171] A. Sakata, Y. Makihara, N. Takemura, D. Muramatsu, and Y. Yagi, "Gait-Based Age Estimation using a DenseNet," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11367 LNCS, pp. 55–63, 2019.

[172] G. Huang, L. Zhuang, V. D. M. Laurens, and W. Kilian Q., "Densely Connected Convolutional Networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[173] L. Bottou and O. Bousquet, "The Tradeoffs of Large Scale Learning," *Advances in Neural Information Processing Systems 20 - Proceedings of the 2007 Conference*, 2007.

[174] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the Importance of Initialization and Momentum in Deep Learning," *30th International Conference on Machine Learning, ICML 2013*, no. PART 3, pp. 2176–2184, 2013.

[175] S. Zhang, Y. Wang, and A. Li, "Gait-Based Age Estimation with Deep Convolutional Neural Network," *2019 International Conference on Biometrics, ICB 2019*, 2019.

[176] K. He, Z. Xiangyu, R. Shaoqing, and S. Jian, "Deep Residual Learning for Image Recognition," in *Proceedings on the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[177] D. P. Kingma and J. L. Ba, "Adam: A Method for Stochastic Optimization," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–15, 2015.

[178] X. Li, Y. Makihara, C. Xu, Y. Yagi, and M. Ren, "Make the Bag Disappear: Carrying Status-invariant Gait-based Human Age Estimation using Parallel Generative Adversarial Networks," *2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems, BTAS 2019*, 2019.

[179] H. Zhu, Y. Zhang, G. Li, J. Zhang, and H. Shan, "Ordinal Distribution Regression for Gait-Based Age Estimation," *Science China Information Sciences*, vol. 63, no. 2, pp. 1–14, 2020.

[180] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–14, 2015.

[181] M. J. Marín-Jiménez, F. Castro, N. Guil, F. De la Torre, and R. Medina-Carnicer, "Deep

Multi-Task Learning for Gait-Based Biometrics," *Conference : IEEE International Conference on Image Processing (ICIP)*, pp. 106–110, 2017.

[182] B. Abirami, T. S. Subashini, and V. Mahavaishnavi, "Automatic Age-Group Estimation From Gait Energy Images," *Materials Today: Proceedings*, vol. 33, pp. 4646–4649, 2020. [Online]. Available: https://doi.org/10.1016/j.matpr.2020.08.298

[183] C. Xu, Y. Makihara, R. Liao, H. Niitsuma, X. Li, Y. Yagi, and J. Lu, "Real-Time Gait-Based Age Estimation and Gender Classification from a Single Image," *Proceedings - 2021 IEEE Winter Conference on Applications of Computer Vision, WACV 2021*, pp. 3459–3469, 2021.

[184] N. S. Russel and A. Selvaraj, "Gender Discrimination, Age Group Classification and Carried Object Recognition from Gait Energy Image using Fusion of Parallel Convolutional Neural Network," *IET Image Processing*, vol. 15, no. 1, pp. 239–251, 2021.

[185] J. W. Davis and H. Gao, "Gender Recognition from Walking Movements using Adaptive Three-Mode PCA," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 2004-Janua, no. January, 2004.

[186] D. Zhang and Y. Wang, "Investigating the Separability of Features from Different Views for Gait Based Gender Classification," *Proceedings - International Conference on Pattern Recognition*, pp. 3–6, 2008.

[187] P. C. Chang, M. C. Tien, J. L. Wu, and C. S. Hu, "Real-Time Gender Classification from Human Gait for Arbitrary View Angles," *ISM 2009 - 11th IEEE International Symposium on Multimedia*, pp. 88–95, 2009.

[188] K. Arai and R. A. Asmara, "Human Gait Gender Classification in Spatial and Temporal Reasoning," *International Journal of Advanced Research in Artificial Intelligence*, vol. 1, no. 6, pp. 1–6, 2012.

[189] T. Liu, B. Sun, M. Chi, and X. Zeng, "Gender Recognition using Dynamic Gait Energy Image," *Proceedings of the 2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference, ITNEC 2017*, vol. 2018-Janua, pp. 1078–1081, 2018.

[190] T. Liu, X. Ye, and B. Sun, "Combining Convolutional Neural Network and Support Vector

Machine for Gait-based Gender Recognition," *Proceedings 2018 Chinese Automation Congress, CAC 2018*, pp. 3477–3481, 2019.

[191] K. Kitchat, N. Khamsemanan, and C. Nattee, "Gender Classification From Gait Silhouette using Observation Angle-Based GEIs," *Proceedings of the IEEE 2019 9th International Conference on Cybernetics and Intelligent Systems and Robotics, Automation and Mechatronics, CIS and RAM 2019*, pp. 485–490, 2019.

[192] P. Smith and C. Chen, "Transfer Learning with Deep CNNs for Gender Recognition and Age Estimation," *Proceedings - 2018 IEEE International Conference on Big Data, Big Data 2018*, pp. 2564–2571, 2019.

[193] Pyrosales, "Termal Imaging - Where is it Used?" 2018. [Online]. Available: https://www.pyrosales.com.au/blog/thermal-imaging/thermal-imaging-where-is-it-used/

[194] V. Karneichyk, "Infrared And Thermal Imaging Design." [Online]. Available: https://www.opticsforhire.com/blog/design-of-ir-lenses

[195] D. Tan, K. Huang, S. Yu, and T. Tan, "Efficient Night Gait Recognition Based on Template Matching," *Proceedings - International Conference on Pattern Recognition*, vol. 3, pp. 1000–1003, 2006.

[196] E. Goubet, J. Katz, and F. Porikli, "Pedestrian Tracking using Thermal Infrared Imaging," *Infrared Technology and Applications XXXII*, vol. 6206, p. 62062C, 2006.

[197] D. Ming, Z. Xue, L. Meng, B. Wan, Y. Hu, and K. D. Luk, "Identification of Humans using Infrared Gait Recognition," *2009 IEEE International Conference on Virtual Environments, Human-Computer Interfaces, and Measurements Systems, VECIMS 2009 - Proceedings*, no. 60501005, pp. 319–322, 2009.

[198] B. DeCann and A. Ross, "Gait Curves for Human Recognition, Backpack Detection, and Silhouette Correction in a Nighttime Environment," *Biometric Technology for Human Identification VII*, vol. 7667, p. 76670Q, 2010.

[199] T. Bourlai, N. Kalka, D. Cao, B. Decann, Z. Jafri, F. Nicolo, C. Whitelam, J. Zuo, D. Adjeroh, B. Cukic, J. Dawson, L. Hornak, A. Ross, and N. A. Schmid, "Distributed Video

Sensor Networks," *Distributed Video Sensor Networks*, pp. 451–467, 2011.

[200] K. Arai and R. Andrie, "Human Gait Gender Classification in Natural Condition Utilizing NIR Camera," in *The Institute of Image Electronics Engineers of Japan*, 2012, pp. 203–210.

[201] A. Isar, S. Moga, and X. Lurton, "A Statistical Analysis of the 2d Discrete Wavelet Transform," *Proceedings of the International Conference AMSDA*, pp. 17–20, 2010.

[202] D. T. Nguyen and K. R. Park, "Body-Based Gender Recognition using Images from Visible and Thermal Cameras," *Sensors (Switzerland)*, vol. 16, no. 2, 2016.

[203] J. H. Lee, J. S. Choi, E. S. Jeon, Y. G. Kim, T. T. Le, K. Y. Shin, H. C. Lee, and K. R. Park, "Robust Pedestrian Detection by Combining Visible and Thermal Infrared Cameras," *Sensors (Switzerland)*, vol. 15, no. 5, pp. 10 580–10 615, 2015.

[204] D. T. Nguyen, K. W. Kim, H. G. Hong, J. H. Koo, M. C. Kim, and K. R. Park, "Gender Recognition from Human-Body Images using Visible-Light and Thermal Camera Videos Based on a Convolutional Neural Network for Image Feature Extraction," *Sensors (Switzerland)*, vol. 17, no. 3, 2017.

[205] A. Wu, W. S. Zheng, S. Gong, and J. Lai, "RGB-IR Person Re-identification by Cross-Modality Similarity Preservation," *International Journal of Computer Vision*, vol. 128, no. 6, pp. 1765–1785, 2020. [Online]. Available: https://doi.org/10.1007/s11263-019-01290-1

[206] N. R. Baek, S. W. Cho, J. H. Koo, and K. R. Park, "Pedestrian Gender Recognition by Style Transfer of Visible-Light Image to Infrared-Light Image Based on an Attention-Guided Generative Adversarial Network," *Mathematics*, vol. 9, no. 20, 2021.

[207] A. Javed, *Building Arduino Projects for the Internet of Things*, 2016.

[208] W. Commons, "Arduino Due Front," 2015. [Online]. Available: https://commons.wikimedia.org/wiki/File:ArduinoDue_Front.jpg

[209] SparkFun Electronics, "Sparkfun Start Something," 2018. [Online]. Available: https://www.sparkfun.com/

[210] M. Beyeler, *OpenCV with Python Blueprints Design and Develop Advanced Computer Vision Projects using OpenCV with Python*. Packt Publishing, 2015.

[211] J. Hrisko, "High Resolution Thermal Camera with Raspberry Pi and MLX90640," 2020. [Online]. Available: https://makersportal.com/blog/2020/6/8/high-resolution-thermal-camera-with-raspberry-pi-and-mlx90640

[212] R. Wadhwa, "Improving Saguenay," 2017. [Online]. Available: https://storymaps.arcgis.com/stories/c1a6bbceefaa4935bedf88a997ee1980

[213] J. Bergstra and Y. Bengio, "Random Search for Hyper-Parameter Optimization," *Journal of Machine Learning Research*, vol. 13, pp. 281–305, 2012.

[214] F. Zou, L. Shen, Z. Jie, W. Zhang, and W. Liu, "A Sufficient Condition for Convergences of Adam and RMSProp," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, no. 1, pp. 11 119–11 127, 2019.

[215] A. Kathuria, "Intro to Optimization in Deep Learning: Momentum, RMSProp and Adam," 2018. [Online]. Available: https://blog.paperspace.com/intro-to-optimization-momentum-rmsprop-adam/

[216] A. F. Agarap, "Deep Learning using Rectified Linear Units (ReLU)," no. 1, pp. 2–8, 2018. [Online]. Available: http://arxiv.org/abs/1803.08375

[217] L. Danqing, "A Practical Guide to ReLU," 2017. [Online]. Available: https://medium.com/@danqing/a-practical-guide-to-relu-b83ca804f1f7

[218] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical Evaluation of Rectified Activations in Convolutional Network," 2015. [Online]. Available: http://arxiv.org/abs/1505.00853

[219] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[220] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, "How Does Batch Normalization Help

Optimization?" *Advances in Neural Information Processing Systems*, vol. 2018-Decem, no. NeurIPS, pp. 2483–2493, 2018.

[221] J. Kukačka, V. Golkov, and D. Cremers, "Regularization for Deep Learning: A Taxonomy," pp. 1–23, 2017. [Online]. Available: http://arxiv.org/abs/1710.10686

[222] R. Nardelli, Z. Dall, and S. Skevoulis, "Comparing Tensorflow Deep Learning Performance and Experiences using CPUs via Local PCs and Cloud Solutions," in *Future of Information and Communication Conference*, 2019, pp. 118–130.

[223] A. A. Suzen, B. Duman, and B. Sen, "Benchmark Analysis of Jetson TX2, Jetson Nano and Raspberry PI using Deep-CNN," *HORA 2020 - 2nd International Congress on Human-Computer Interaction, Optimization and Robotic Applications, Proceedings*, pp. 3–7, 2020.

[224] J. Rueterbories, E. G. Spaich, B. Larsen, and O. K. Andersen, "Methods for Gait Event Detection and Analysis in Ambulatory Systems," *Medical Engineering and Physics*, vol. 32, no. 6, pp. 545–552, 2010. [Online]. Available: http://dx.doi.org/10.1016/j.medengphy.2010.03.007

[225] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Poloushkin, "Attention is All you Need," in *Advances in neural information processing systems*, vol. 30, 2017.

[226] S. Zhai, S. Dingrong, W. Shuhuan, and D. Susu, "DF-SSD: An Improved SSD Object Detection Algorithm Based on DenseNet and Feature Fusion," *IEEE access*, vol. 8, pp. 24 344–24 357, 2021.

[227] A. Kolli, A. Fasih, F. A. Machot, and K. Kyamakya, "Non- intrusive Car Driver's Emotion Recognition using Thermal Camera," in *Proceedings of the Joint INDS 11 & ISTET 11, IEEE*, 2011, pp. 1–5.

[228] J. Hossen, E. Jacobs, and F. K. Chowdhury, "Human Suspicious Activity Recognition in Thermal Infrared Video," *Infrared Sensors, Devices, and Applications IV*, vol. 9220, p. 92200E, 2014.

[229] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "Mixmatch:

A Holistic Approach to Semi-Supervised Learning," in *Advances in Neural Information Processing Systems*, no. 32, 2019.

[230] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese Neural Networks for One-shot Image Recognition," Tech. Rep., 2015.

[231] M. Gong and Y. Shu, "Real-Time Detection and Motion Recognition of Human Moving Objects Based on Deep Learning and Multi-Scale Feature Fusion in Video," *IEEE Access*, vol. 8, pp. 25 811–25 822, 2020.

[232] M. Kristo, M. Ivasic-Kos, and M. Pobar, "Thermal Object Detection in Difficult Weather Conditions using YOLO," *IEEE Access*, vol. 8, pp. 125 459–125 476, 2020.

[233] U. Mittal, S. Srivastava, and P. Chawla, "Object Detection and Classification from Thermal Images using Region Based Convolutional Neural Network," *Journal of Computer Science*, vol. 15, no. 7, pp. 961–971, 2019.

[234] K. Agrawal and A. Subramanian, "Enhancing Object Detection in Adverse Conditions using Thermal Imaging," 2019. [Online]. Available: http://arxiv.org/abs/1909.13551