

## Article

# Deep-Learning-Based Automatic Mineral Grain Segmentation and Recognition

Ghazanfar Latif <sup>1,\*</sup>, Kévin Bouchard <sup>1</sup>, Julien Maitre <sup>1</sup>, Arnaud Back <sup>2</sup> and Léo Paul Bédard <sup>2</sup>

<sup>1</sup> Department of Computer Sciences and Mathematics, Université du Québec à Chicoutimi, 555 Boulevard de l'Université, Chicoutimi, QC G7H 2B1, Canada; kevin\_bouchard@uqac.ca (K.B.); julien1\_maitre@uqac.ca (J.M.)

<sup>2</sup> LabMaTer, Sciences de la Terre, Université du Québec à Chicoutimi, 555 Boulevard de l'Université, Chicoutimi, QC G7H 2B1, Canada; arnaud-louis-abel.back1@uqac.ca (A.B.); paul\_bedard@uqac.ca (L.P.B.)

\* Correspondence: ghazanfar.latif1@uqac.ca

**Abstract:** A multitude of applications in engineering, ore processing, mineral exploration, and environmental science require grain recognition and the counting of minerals. Typically, this task is performed manually with the drawback of monopolizing both time and resources. Moreover, it requires highly trained personnel with a wealth of knowledge and equipment, such as scanning electron microscopes and optical microscopes. Advances in machine learning and deep learning make it possible to envision the automation of many complex tasks in various fields of science at an accuracy equal to human performance, thereby, avoiding placing human resources into tedious and repetitive tasks, improving time efficiency, and lowering costs. Here, we develop deep-learning algorithms to automate the recognition of minerals directly from the grains captured from optical microscopes. Building upon our previous work and applying state-of-the-art technology, we modify a superpixel segmentation method to prepare data for the deep-learning algorithms. We compare two residual network architectures (ResNet 1 and ResNet 2) for the classification and identification processes. We achieve a validation accuracy of 90.5% using the ResNet 2 architecture with 47 layers. Our approach produces an effective application of deep learning to automate mineral recognition and counting from grains while also achieving a better recognition rate than reported thus far in the literature for this process and other well-known, deep-learning-based models, including AlexNet, GoogleNet, and LeNet.

**Keywords:** grain segmentation; deep learning; convolutional neural networks; ResNet; mineral recognition



**Citation:** Latif, G.; Bouchard, K.; Maitre, J.; Back, A.; Bédard, L.P. Deep-Learning-Based Automatic Mineral Grain Segmentation and Recognition. *Minerals* **2022**, *12*, 455. <https://doi.org/10.3390/min12040455>

Academic Editor: Amin Beiranvand Pour

Received: 21 February 2022

Accepted: 2 April 2022

Published: 7 April 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The advent of machine learning and automated classification has demonstrated the potential of technology in many fields, such as medical/health, legal, transportation, and mining [1–6]. For example, in exploration geology and mining, the process of identifying economic minerals has always been done manually, where a specialized and trained individual (a mineralogist) is required to identify minerals grains, such as gold, diamond indicator minerals, or sulfides, to discover new deposits [7–9]. This manual process has many limitations, including errors in identification and mineralogist fatigue and its time-consuming and, hence, costly [10]. Moreover, trained mineralogists are able to count around 60 grains per minute without distractions to provide grain percentage rather than the more useful area percentage [11]. With new advances in technology, mineral grain identification and counting can now be performed using optical microscopy and scanning electron microscopy (SEM). However, even with SEM technology, the process remains expensive and time consuming. A scanning electron microscope costs between USD 0.5 and USD 2 million and requires highly qualified personnel to operate.

Nevertheless, the process of identifying and counting mineral grains in sands or sediments is a crucial step for many mineral exploration and engineering projects, environmental studies, and mining (extractive metallurgy); for example, minerals can be economic (e.g., ore, building materials) or toxic (e.g., acid mine drainage production or release of toxic elements such as lead or arsenic) [12,13]. The use of certain sands in building materials can be a major problem, and identifying such grains is crucial in engineering projects [14,15]. In glacial sediments (tills) and soils, the number and count of certain mineral grains can indicate the proximity of a potential deposit; in diamond exploration, for instance, certain minerals, such as chromium-bearing pyrope or diopside, are used to confirm the presence of proximal diamond deposits [16,17].

Machine learning offers an alternative to manual identification. Recent advances in deep learning for image-based tasks offer the possibility of automating, at least partially, grain identification and counting, saving time and money. Moreover, as opposed to relying on SEM [18], the deep-learning-based approach can potentially be carried out in the field, in remote areas where mineral potential is high. Such a method would allow a more rapid identification of economic minerals or toxic minerals to allow effective environmental surveys [19]. This automated approach could work in real time to sort minerals moving along on a conveyor [20]. A robot with specialized tools and equipment could be used to capture images of grain as it explores the terrains [21]. If images are tagged with a location, real-time processing is not obligatory, thus, simplifying the challenge of embedding a deep-learning model in a remote and potentially smaller computer.

In this paper, we propose an automated machine-learning approach to classify grains from a sample using optical microscopy that builds upon our previous work published in [22]. With this approach, the task of mineral identification requires minimal human intervention. The images of grains are collected using inexpensive photomicrographic systems or through the use of robotic machines or automated microscopes. The images (photomicrographs) can then be processed to isolate the grain images within the complete image and, thus, classify and count these grains. Our approach uses an improved superpixel method that segments the grains quickly and automatically. To solve it using deep learning, the segmentation must be very accurate for the model to automatically learn the features representing each class. Then, the segmented grains can be used as an input into the trained deep-learning model. Although deep learning frequently outperforms classical machine learning, it is only recently that mineral identification has been investigated with deep learning. With the new segmentation and state-of-the-art deep-learning models, we can achieve better results than observed in published classical machine-learning-based approaches.

## 2. Literature Review

Currently, there are mainly two distinct methods for grain recognition: traditional engineering devices [23,24] and computational methods [25].

### 2.1. Traditional, Device-Based Methods

Traditional methods for classifying and counting mineral grains rely on the use of SEM or optical microscopes. The use of an optical microscope is the most common method for estimating mineral abundance in sediments or milled rock, although this requires highly trained personnel to sort the mineral grains. Mineral sorting is possible using the specific polarized transmitted and reflected light properties of minerals and the morphological properties of the grains. Advances in the use of optical microscopes have been successfully applied to mineral grain analyses, although the main limitations discussed above remain [26–29]. Significant improvement of this method will require a technical breakthrough. Automated SEM provides an alternative means of counting minerals [11,30], and the SEM-based approaches include QEMSCAN, TIMA-X, and MLA [31]. SEM uses a focused electron beam to scan the material and generate an image of the grains. The interaction of the electrons with atoms on the grain surface provides additional information captured by the various sensors (e.g., X-ray fluorescence) to determine the chemical

composition of the mineral. SEM output includes the chemical composition with grain size, shape, and proportion. Grain counting can be performed using an electron microprobe [32]; however, this method is time consuming [33].

In [34], the authors presented an image processing workflow for characterizing pore- and grain-size distributions in porous geological samples using SEM images and X-ray microcomputed tomography ( $\mu$ CT). Their samples included the Buff Berea, Berea, Nugget, Bentheimer, and Castlegate sandstones and the carbonate Indiana Limestone. The produced 2D distribution from the SEM appeared biased toward smaller sizes. In [35], the authors developed a grain count technique using a laser particle counter sensor (Wenglor) to count stainless-steel beads and sand grains of different size classes. They compared the count with that obtained using high-speed cameras. They found that the Wenglor can count grain sizes between  $210 \pm 3 \mu\text{m}$  and  $495 \pm 10 \mu\text{m}$  and that only grains passing through the center of the beam were counted. In [36], the authors used a less expensive light microscope able to produce images of grain shape profiles sufficient in quality for identification and counting. Their key finding was that roundness, sphericity, circularity, ModRatio, and aspect ratio were the key shape parameters for differentiating grains.

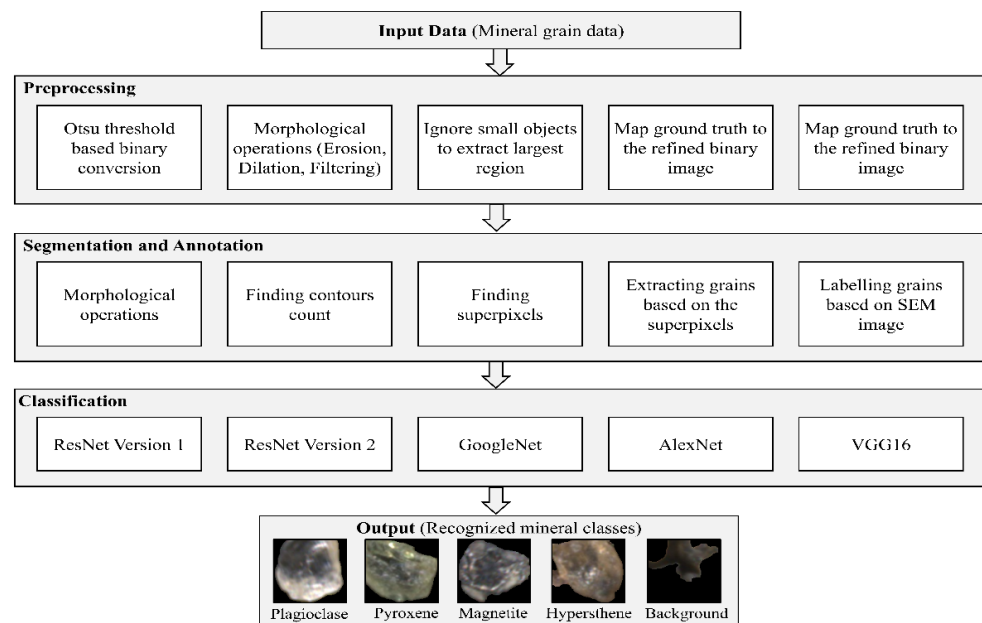
## 2.2. Computer Vision-Based Computational Methods

Computational or machine-learning methods are increasingly applied in a multitude of spheres, including automated driving and navigation, automated image recognition, automated medical diagnosis, and agricultural processes [37–39]. The ability to apply machine-learning tools to a vast suite of applications also extends to the environmental and geological sciences.

The integration of machine learning to automate the process of mineral grain recognition was first explored by Maitre et al. [22]. The authors used linear iterative clustering segmentation to generate superpixels, thereby, isolating individual grains. The applied feature extraction method, using a series of classifiers, produced an 89% recognition rate. In [25], cluster analysis through a k-means algorithm for mineral recognition divided the data set into categories according to the similarity, computed by distance, e.g., Euclidean distance. Baklanova and Shvets extracted the colors and textures of grains using a stereoscopic binocular microscope. However, the authors failed to compare clusters found with labeled clusters that actually belonged to a certain species of minerals. In fact, their work was used only to classify rocks and not minerals and, thus, their work is only applicable to petrography. Other methods of mineral classification, although limited to copper minerals, have produced an acceptable, approximate 75% accuracy using laser-induced breakdown spectroscopy (LIBS) analyzers [40]. In [41], the authors classified heavy minerals collected from rivers. Using 3067 grains in 22 classes, they achieved 98.8% accuracy using 26 decision attributes and a random forest algorithm.

## 3. Materials and Methods

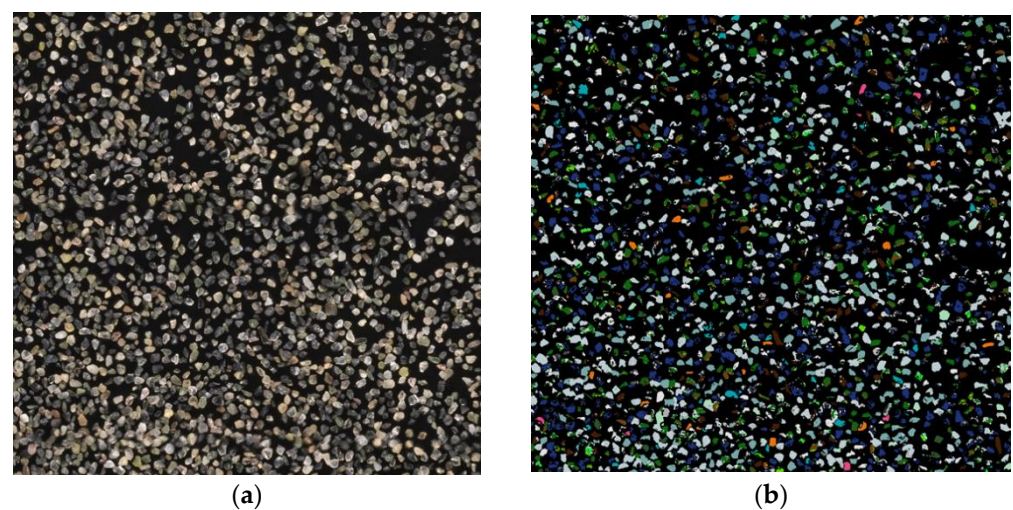
Our approach consisted of four main stages (Figure 1). The first stage involved data collection followed by preprocessing the original mosaic and SEM images to remove noise and outlier objects. In the third stage, the grains were segmented by utilizing the contours and superpixel-based techniques. We selected five classes for recognition on the basis of classes with the greatest number of grains. In the final stage, we input the segmented grains into various convolutional neural network (CNN) models.



**Figure 1.** Workflow of the grains recognition method used in this paper.

### 3.1. Data Set Acquisition

We collected 10 kg of till grains from the field, and sediments were sieved to less than 1 mm. The samples were then processed with a fluidized bed to obtain a superconcentrate of heavy minerals (approximately 100 mg) containing approximately 2 million grains smaller than 50  $\mu\text{m}$ . The superconcentrate was sprinkled onto carbon tape to provide a black backdrop for the images. Images were then obtained using a camera mounted onto a binocular microscope, and we created a photomosaic. To acquire the groundtruthed data, i.e., mineral grain identities, we acquired a backscattered image of the grains using SEM with X-ray fluorescence [42]. The groundtruthed data were the mineral map and referenced with the RGB mosaic. The end result, after using the motorized conventional microscope and 6-megapixel camera, was an approximate 2 GB mosaic image ( $34,674 \times 33,720$  pixels) to be used as the data set for the machine-learning algorithm. We acquired 238 fields of view with a 10% overlap between adjacent fields in the images. Figure 2 shows the sample of the grains and the corresponding, annotated SEM image.



**Figure 2.** Sample segment of the original (a) and SEM-based groundtruth image (b).



### 3.2. Data Preprocessing

The original image background consisted of outlier grains that are not part of the SEM annotated image; therefore, preprocessing, using various morphological operations, served to remove outlier particles. An outlier grain is a phantom image of a grain lying outside of the field of view. To reduce processing time, we cropped the original image to include only 1/3 of the original image by discarding 12,000 border pixels on all sides that did not contain grains. This new image was further divided into  $5608 \times 5608$  equally sized subimages. We considered only five classes for classification because of unbalanced data and a low number of instances for some of the discarded classes.

The groundtruthed image was converted into a binary image and morphological operations were applied, i.e., dilation, filling holes, and erosion, to remove the outlier grains, the background, and other noise. The largest filled segment of the SEM-based, labeled image was extracted by discarding all outlier grains and other noise. The erosion and dilation work was based on kernel size to reduce the size of the input image. Similarly, dilation increased the size of the input image on the basis of kernel size. We applied a kernel size of  $7 \times 7$ . The erosion and dilation for the binary image were calculated using Equations (1) and (2), respectively, where  $A$  represents the original binary image, and  $B$  represents the kernel. In Equation (1),  $B_z$  is translation  $B$  by vector  $z$ . Similarly, in Equation (2),  $A_b$  is translation  $A$  by vector  $b$ .

$$A \ominus B = \{z \in E \mid B_z\} \quad (1)$$

$$A \oplus B = \bigcup_{b \in E} A_b \quad (2)$$

Figure 3 shows the outcome of different preprocessing steps and mapping of the SEM ground truth image and the Original Image based on the processed SEM binary image.

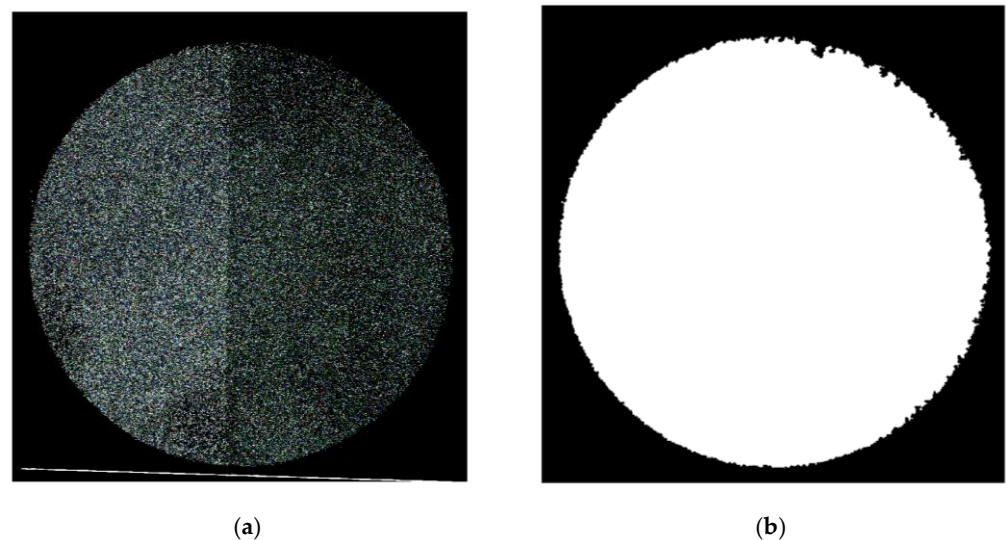
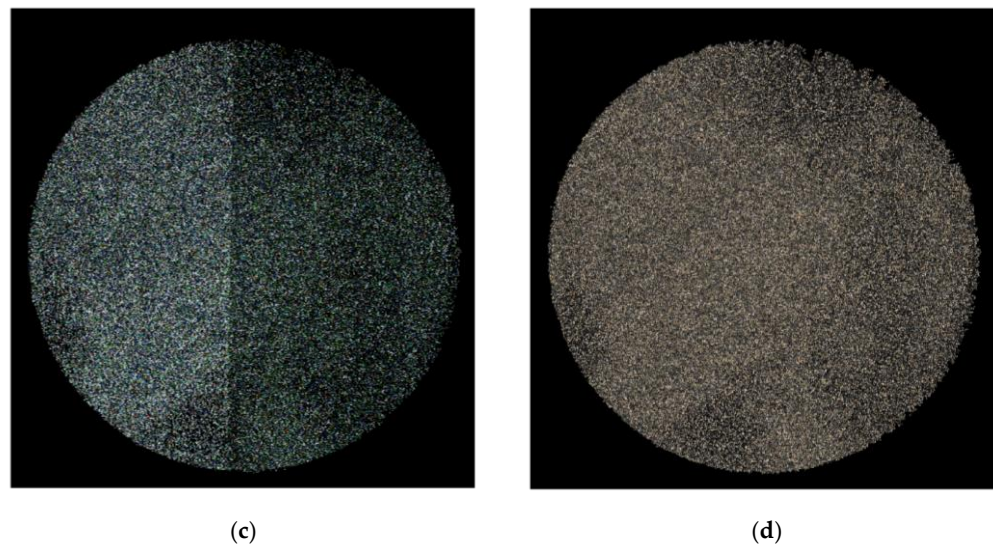


Figure 3. Cont.



**Figure 3.** Outcome of the preprocessing steps. Images show the original SEM images (a), the outliers removed using the morphological and contours-based method from the binary converted image (b), the preprocessed outcome SEM image after removing outliers (c), and the preprocessed outcome of the original image after removing outliers (d). Wrong format.

### 3.3. Grain Segmentation

We used superpixel segmentation to separate mineral grain data (see Algorithm 1). The image was first converted to binary, and morphological operations—erosion and dilation—were applied to the image to separate the grains from each other. To convert the image into binary, the image threshold was calculated using Otsu’s method [43]. Using the resulting binary image, we calculated the total number of external, closed contours to represent the possible grains in the image. Contours are closed curves that are calculated using the edges of objects with the same values or pixel intensities. The contour count  $C$  then serves as a seed for the superpixel segmentation method rather than using a fixed number  $K$  as a seed. We applied Equation (3) to calculate the superpixel center grid interval of approximately equal-sized superpixels of an input image of size  $N$ .

$$S = \sqrt{\frac{N}{C}}. \quad (3)$$

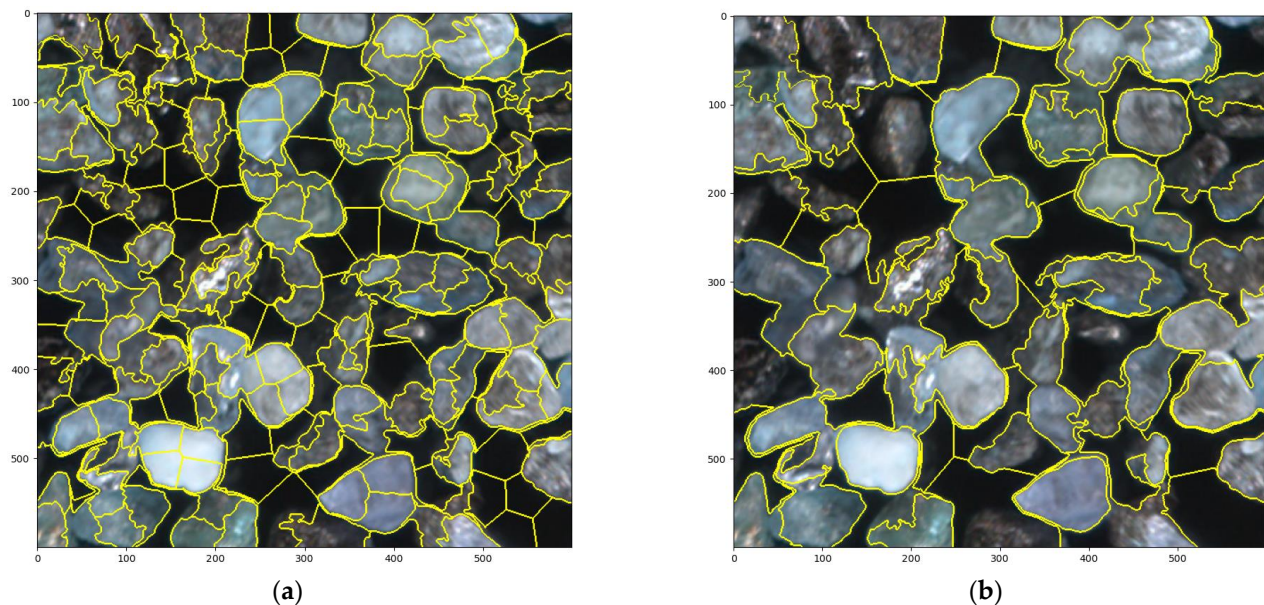
The superpixel segmentation method relies on oversegmenting the image while simultaneously decreasing the complexity of the image processing tasks. We applied a simple linear iterative clustering (SLIC) method to produce high-quality segmentation in a timely manner [44]. The method performs local k-mean clustering of the image pixels using color similarity and proximity in the subimages. The method also uses the five-dimensional spaces provided by the *labxy* image plane, where  $l$ ,  $a$ , and  $b$  are the pixel vector colors provided by the CIELAB color space, and the  $x$  and  $y$  values are the coordinates of the pixels which represent the spatial distances. To merge the color proximity and spatial proximity distances, we normalized the distances using Equations (4) and (5). To use the *labxy* space to cluster the pixels, we required the distance measure  $D$ , which considers approximately equal-sized superpixels.

$$D_c = \sqrt{(l_m - l_n)^2 + (a_m - a_n)^2 + (b_m - b_n)^2}. \quad (4)$$

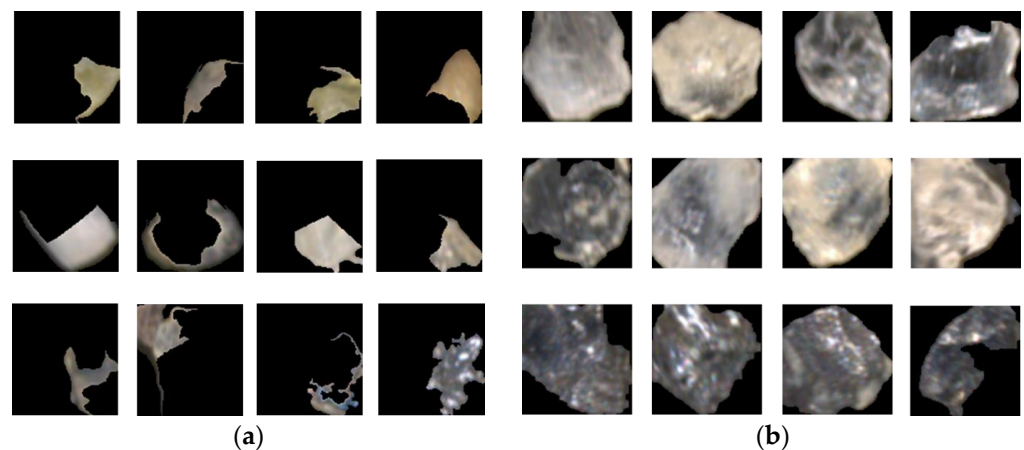
$$D_s = \sqrt{(x_m - x_n)^2 + (y_m - y_n)^2}. \quad (5)$$

$$D = \sqrt{\left(\frac{D_c}{N_c}\right)^2 + \left(\frac{D_s}{N_s}\right)^2}. \quad (6)$$

The segmentation provided the  $xy$  coordinates of each superpixel. The method was further enhanced by increasing the contrast of the images to allow the discrimination of the grain borders. In Maitre et al. [22], the superpixel method was applied using a fixed-size input seed value for the superpixels. This approach worked well for the color feature-based method with classical machine-learning methods; however, this method did not rely on deep learning. Thus, we proposed to automate the calculation of the seed values in the segmentation method to prepare the data for deep-learning networks. The comparisons of the superpixel boundaries and the outcome for the segmented grains for both methods are presented in Figures 4 and 5, respectively.



**Figure 4.** Visual comparison of the detected superpixel boundaries of grains using the method of Maitre et al. (a) and our approach (b) [22].



**Figure 5.** Samples of the outcome of the segmented grains using the method of Maitre et al. [22] (a) and our approach (b).

**Algorithm 1:** Segmentation and Annotation of Grains**Input:** Grains mosaic image  $M$  with BSE groundtruthing image  $G$  and classes  $n = 5$ **Output:** Segmented grains  $S$  with their annotation  $A$ 

```

read  $M$ , read  $G$ 
 $B \leftarrow \text{binary}(\text{grayscale}(M), \text{Otsu})$ 
 $B_e \leftarrow \text{erosion}(B, \text{ones}(15, 15))$ 
 $B_c \leftarrow \text{find external contours}(B_e, \text{chain approx simple})$ 
 $B_{\text{count}} \leftarrow \text{length}(B_c)$ 
 $\text{GrainsApprox} \leftarrow B_{\text{count}} \times 2.5$ 
 $M_{\text{hq}} \leftarrow \text{histogram equalization}(M)$ 
 $S' \leftarrow \text{superpixel}(M_{\text{hq}}, \text{GrainsApprox}, \text{compactness} = 20, \text{sigma} = 1)$ 
 $D \leftarrow \text{unique colors}(G)$ 
 $c \leftarrow 0$ 
for  $g$  in  $S'$ 
   $S_c \leftarrow g$ 
  if  $(\sum_{i=0}^n \sum_{j=0}^m g(i, j) \neq [0, 0, 0]) \geq \left(\frac{(i \times j)}{1.5} \& 25\right)$ 
    for  $d$  in  $D$ 
       $M_d \leftarrow \text{count}(g(i, j) = d)$ 
       $A_{c,1}, A_{c,2} \leftarrow \text{Max}_1(M), \text{Max}_1(M)$ 
    else
       $A_{c,1}, A_{c,2} \leftarrow 0$ 
     $c \leftarrow c + 1$ 
end for
Select  $n$  classes with maximum grain count

```

**3.4. Grain Class Annotation**

We selected five main classes on the basis of the number of segmented grains for each class and the group distribution of visually similar, rock-forming minerals (Table 1). We selected six types of individual grain that were further mapped to five classes, including the background class. These segmented images were labeled by mapping the original subimages to the SEM-based subimages using the superpixel-based method. The bounding-box method was then applied to extract the grains that had a rectangular format. The grains with a height:width ratio greater than 1.75 were discarded. A total of 21,091 images were segmented.

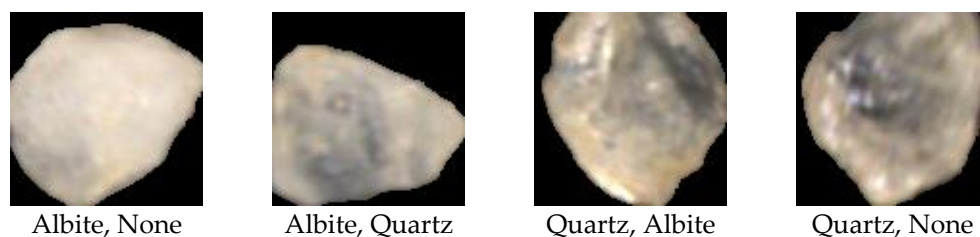
**Table 1.** Summary of the selected grain classes.

Class Label	Primary Grain Type	Secondary Grain Type	Number of Grains
C1	Albite	None	6879 images
	Quartz	None	
	Quartz	Albite	
	Albite	Quartz	
	Albite	Any class > 256 pixels	
	Quartz	Any class > 256 pixels	
C2	Augite	None	3295 images
	Tschermakite	Any class > 256 pixels	
	Tschermakite	Augite	
	Augite	Tschermakite	
C3	Augite	Any class > 256 pixels	3823 images
	Magnetite	Any class > 256 pixels	
C4	Magnetite	None	988 images
	Hypersthene	Any class > 256 pixels	
C5	Hypersthene	None	6106 images
	Background	-	

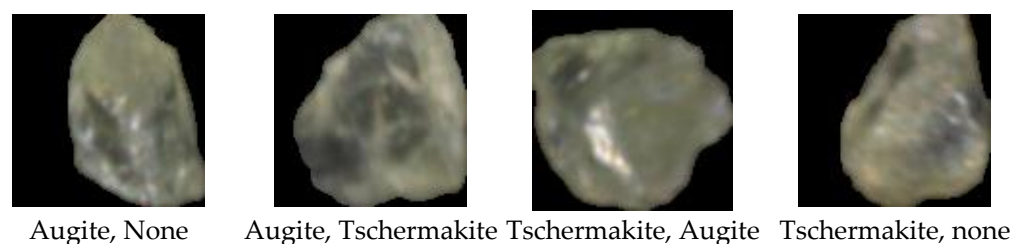
The final data set consisted of 21,091 images divided into five classes. Albite grain and quartz grain images were merged into one class because they are visually similar,



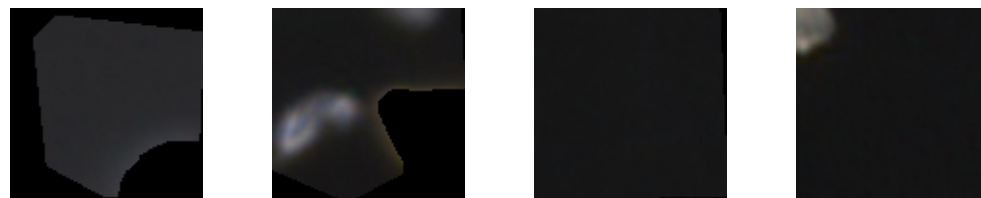
rock-forming minerals. The sample images of albite grain and quartz grain are shown in Figure 6, which clearly indicate their visual similarity. Augite grain and tschermakite grain images were also merged into one class, as are the samples shown in Figure 7, due to their visual similarity. The background class contained images which were either entirely black or contained very small grains (the total number of nonblack pixels was less than 256) or contained noise in the background. Figure 8 shows the sample images of the background class. For the experiments, these five classes' data set images were divided into 20% for training, and the remaining 80% was divided again into 80%/20% for validation/training sets.



**Figure 6.** Sample images of the albite and quartz mineral grains.



**Figure 7.** Sample images of the augite and tschermakite mineral grains.



**Figure 8.** Sample images of background class.

### 3.5. ResNet Models for Grain Recognition

With the growing difficulties in the functions of computer vision and artificial intelligence, deep neural network models are becoming increasingly complex. Such strong models demand more data for learning to prevent overfitting. Recent deep-learning methods have been successfully applied to artificial intelligence [45,46]. Interest in convolutional neural systems (CNN) began in 2012 with AlexNet, which was based on LeNet. New CNN-based models have since been developed, including GoogleNet and residual neural networks (ResNet) [47–49]. CNN's major advantage is its ability to learn the critical features best representing the data without any human intervention.

ResNet overcomes model complexity and the vanishing gradient problems to produce satisfactory accuracies by training deeper networks [50]. Each ResNet block comprises four layers. The weight layer is expressed as  $(Z_{n+1} = W_{n+1}X_n + Y_{n+1})$ . The ReLU layer, a nonlinear layer, is expressed as  $(X_{n+1} = H(Z_{n+1}))$ , and a third layer is a weight layer  $(Z_{n+2} = W_{n+2}X_{n+1} + Y_{n+2})$ .  $X_n$  is the input to the three layers combined, and  $F(X_n)$  is produced in the output. All these variables are matrices, and the subscripts are used to denote the layer numbers. In ResNet, a skip or shortcut link is used to bypass the three layers to pass  $X_n$  to an adder. Thus, the fourth layer, ReLU, is applied to  $F(X_n) = Z_{n+2}$  to produce

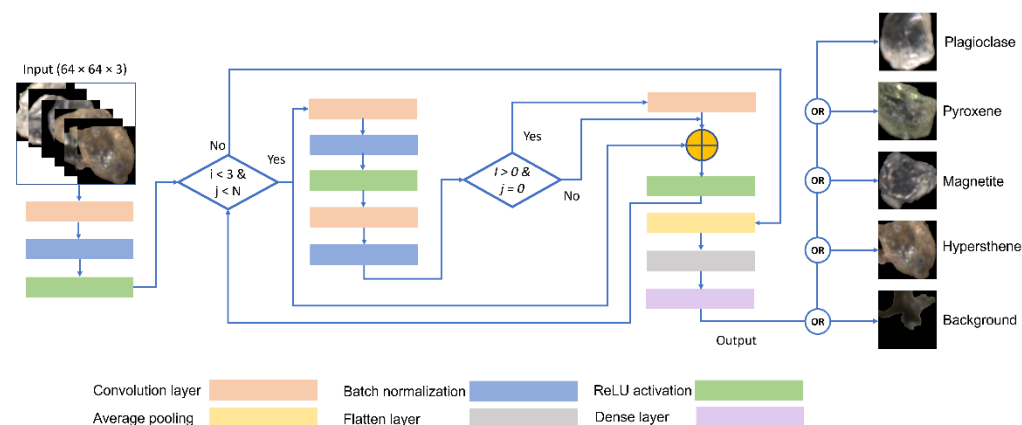
$X_{n+2} = H(Z_{n+2} + X_n)$ . With this skip,  $F(X_n) = H(Z_{n+2})$  is added to  $X_n$  before passing through the second ReLU layer to generate  $X_{n+2}$ .

Skip, or shortcut, connection is a term used to refer to the  $X$  input to the adder. Because  $X$  is passed from one layer to another, the shortcut connection then permits the residual network so that  $F(X) = 0$ , thus, allowing a simple task to be performed by  $X$ . If this shortcut connection is absent, then the network needs to learn that the weights layer is equivalent to the identity matrix multiplied by  $X$ , which adds more complexity to the task. In cases where  $X$  is not required to pass through layers, the network generates  $F(X)$  normally, as is achieved when backpropagation is used. In this case, it is easier to train  $F(x)$  to be the residual  $D(X) - X$ , which results in the desired output of  $D(X)$  when added to  $X$  using the shortcut connection. Because the shortcut connection does not require weights, the gradient values remain unchanged, thus, overcoming the vanishing gradient problem.

Building a sequence of ResNet blocks produces a ResNet architecture with deeper networks with low training errors and excellent accuracies. The ResNet blocks might require pooling layers when convolution or weight layers generate different  $F(X)$  matrices than the original  $X$  matrix. The pooling adds  $X$  to  $F(X)$ , which resizes  $X$  to match the size of the  $F(X)$  matrix. This can be achieved by adding  $(W.X)$  to  $F(X)$ .  $W$ , in this case, is a zero-padded matrix in both the rows and columns missing from the original  $X$ .

### 3.5.1. ResNet Version 1

We used two ResNet architectures, referred to as “ResNet 1” and “ResNet 2”. Figure 9 details the design of ResNet 1 architecture at the block level. No overfitting is present in the ResNet architecture because no additional parameters are introduced. This implies that ResNet is an efficient deep-learning network even for hundreds of network layers. In ResNet 1, a convolutional layer splits the feature map into two at the beginning, and the filter size is doubled to map the convolutional layer, batch layer, and ReLU layer to  $32 \times 32 \times 16$ ,  $16 \times 16 \times 32$ , and  $8 \times 8 \times 64$ , respectively, on the basis of the  $i$  and  $j$  values, where  $i$  represents how many times the filter size must be doubled, and  $j$  represents the number of ResNet block iterations on the basis of  $N$ . The deep-network performance is enhanced by adjusting the input layer using the batch normalization block. ResNet 1 has an input image dimension of  $48 \times 48 \times 3$ , with each layer in the architecture consisting of a convolutional layer, batch normalization layer, and a rectified linear unit (ReLU).

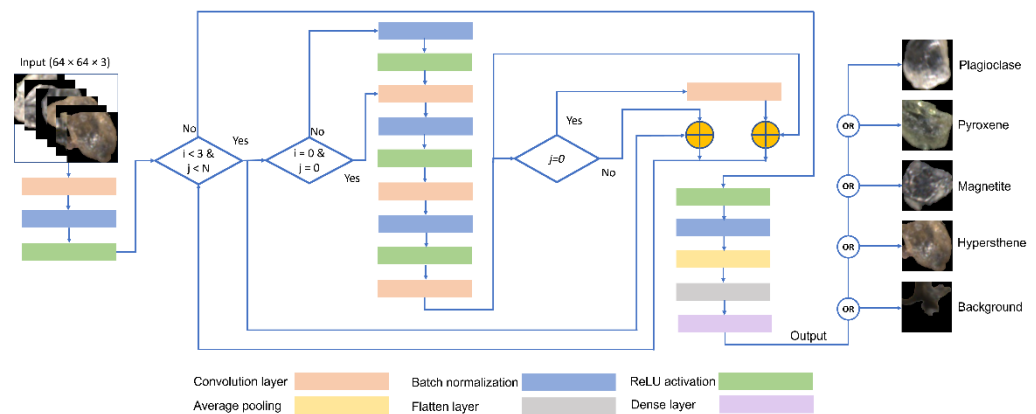


**Figure 9.** ResNet version 1 architecture for mineral recognition.

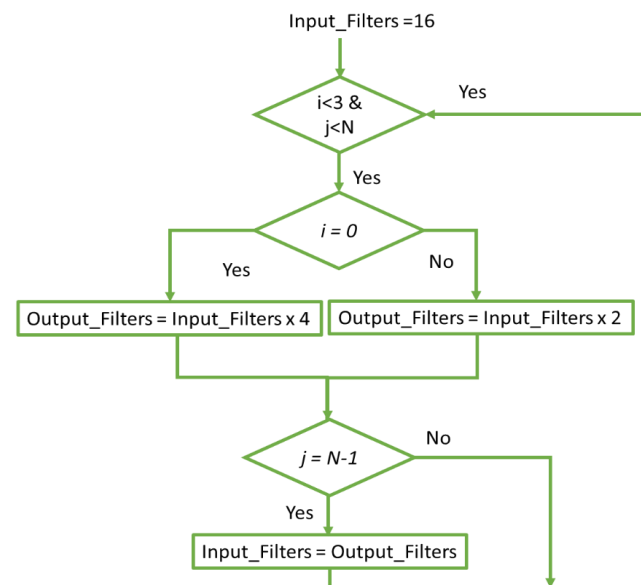
### 3.5.2. ResNet Version 2

ResNet 2 architecture at the block level is detailed in Figure 10, and the filter size for each step is calculated using a flowchart in Figure 11. As for ResNet 1, the feature maps are initially split into two, and the filter maps are doubled. A bottleneck connection is introduced in ResNet 2 with the filter size calculated as shown in Figure 11. In addition, the block size of the skip connection is tripled. The three layers that exist within a residual function block are the convolutional layers sized  $[1 \times 1]$ ,  $[3 \times 3]$ , and  $[1 \times 1]$ , in which the

increase and decrease of input dimensions are performed using the  $1 \times 1$  layer, and the  $3 \times 3$  layer is the bottleneck with reduced dimensions. The stages of ResNet 2 include a convolutional layer  $32 \times 32 \times 16$  in step 1 which produces an output of size  $32 \times 32 \times 64$ . Step 2 produces a  $16 \times 16 \times 128$  output, and step 3 produces an  $8 \times 8 \times 256$  output size. These ResNet 2 outputs are based on the  $i$  and  $j$  values, where  $i$  represents how many times the filter size must be doubled, and  $j$  represents the number of ResNet block iterations based on  $N$ .



**Figure 10.** ResNet version 2 architecture for mineral recognition.



**Figure 11.** ResNet version 2 calculation of filter size for each step.

Note that, in both ResNet 1 and ResNet 2, after the initial concatenation of the blocks in the sequence weights  $\rightarrow$  batch normalization  $\rightarrow$  ReLU, the concatenated sequenced block is repeated. The main difference between the two architectures is:

The sequence that follows the initial weight, batch normalization, and activation block differs between the architectures. For ResNet 1, the following sequence is convolutional block  $\rightarrow$  batch normalization block  $\rightarrow$  activation block, whereas, in ResNet 2, the sequence is batch normalization block  $\rightarrow$  activation block  $\rightarrow$  convolutional block.

Postactivation is supported in ResNet 1.

Preactivation is supported in ResNet 2.

In ResNet 1, the second ReLU nonlinearity is added after adding  $F(X)$  to  $X$ .

In ResNet 2, the last ReLU nonlinearity is deleted, thus, allowing output of the addition of the residual mapping and identity mapping to be passed with no changes to the

consecutive block. In addition, the gradient value at the output layer is passed back during backpropagation, as is the input layer, thus, overcoming the vanishing gradient problem in deep-learning networks that have hundreds or thousands of layers, thereby, improving their performance and limiting/reducing the associated training errors.

For both ResNet models, we used experimentation to fine-tune the hyperparameters. The final hyperparameter settings were an activation function (ReLU) learning rate = 0.001, number of epochs = 50, and batch size = 20. These hyperparameters produced the experimental results discussed in Section 4.

#### 4. Experimental Results

The experimental setup included the use of a high processing computing machine holding 256 GB memory with a graphical processing unit (GPU) Nvidia Tesla-V100 with 5120 CUDA cores. We applied Python 3.8 for the programming of all phases, including the preprocessing, classification, and identification. The data set was split so that 80% was used for training and the remaining 20% was available for testing. Note, however, that the 80% training portion was actually divided again into an 80% training and 20% validation split. We tested variable epoch sizes, and the ideal epoch size was chosen to ensure that the system avoided over- and underfitting. We tested various parameter settings for ResNet 1 and ResNet 2 to obtain the optimal results and evaluated their performance against the better-known deep-learning approaches of LeNet, AlexNet, and GoogleNet.

ResNet 1 and ResNet 2 achieved higher validation accuracies than LeNet, AlexNet, and GoogleNet (Table 2). The validation accuracy of ResNet 2 was slightly higher than for ResNet 1. We obtained these scores by applying the segmentation methods presented in [22]. In the latter paper, they achieved a global accuracy of 89% using a RF classifier; however, their data were not effective when deep-learning algorithms were applied.

**Table 2.** Results using minerals segmentation used in [22] with different CNN models.

CNN Model	Training Loss	Validation Loss	Training Accuracy (%)	Validation Accuracy (%)
LeNet	1.1329	1.5627	66.67	39.29
AlexNet	0.3917	2.8706	88.89	39.88
GoogleNet	0.9911	1.4571	83.33	43.37
ResNet 1 (32)	1.0715	1.3784	72.29	45.61
ResNet 2 (47)	1.0263	1.3269	76.94	49.23

We used superpixel segmentation combined with the proposed ResNet architectures to produce much higher validation accuracies than those achieved in [22]. LeNet, AlexNet, and GoogleNet produced validation accuracies ranging from 74.4% to 86.3%, with the highest accuracy achieved by AlexNet as shown in Table 3. However, the proposed ResNet 1 and ResNet 2 achieved a higher validation accuracy of 89.8% and 90.6%, respectively. Notice that, compared to the highest achieved validation accuracy in [22], which was 49%, our proposed method increased by 84.69%, which is a significant increase by all measures. The highest achieved validation accuracy of 90.5% produced by the ResNet 2 architecture of 47 layers sets a new threshold for researchers in the field of grain recognition. It is also an improvement of 1.69% when compared to the accuracy achieved in [22] using an RF classifier.

We varied the number of layers for ResNet 1 and ResNet 2 to determine the best parameters for achieving the highest accuracy. The best accuracy for ResNet 1 was achieved using 74 layers (Table 4); however, although there was a slight improvement going from 32 to 74 layers, training times increased markedly for 74 layers. Hence, ResNet 1 with 32 layers was the chosen architecture for this application. For ResNet 2, we found the highest validation accuracy using 47 layers, accompanied by a reasonable training time. Although



the training time between 29 layers and 47 almost doubled, the increased validation accuracy justified using the 47 layers for this application.

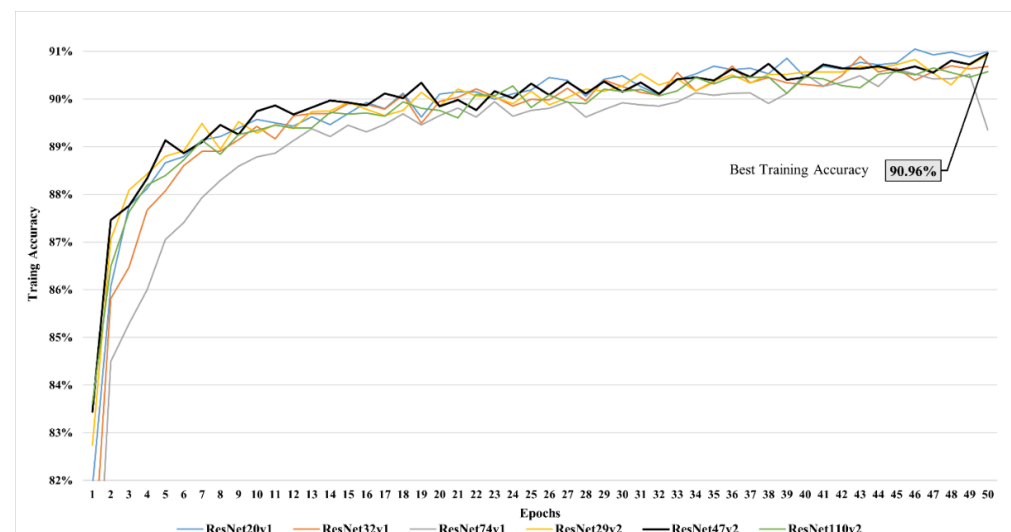
**Table 3.** Results using proposed minerals segmentation with different CNN models.

CNN Model	Training Loss	Validation Loss	Training Accuracy (%)	Validation Accuracy (%)
LeNet	1.063	0.6374	61.60	74.43
AlexNet	0.3425	0.3847	90.00	86.30
GoogleNet	0.7875	0.626	72.40	76.23
ResNet 1 (32)	0.3418	0.3668	90.40	89.80
ResNet 2 (47)	0.3523	0.3621	90.40	90.56

**Table 4.** Comparison of results using proposed minerals segmentation with different ResNet models and varying number of layers.

Model	# of Layers	Training Loss	Validation Loss	Training Accuracy (%)	Validation Accuracy (%)	Training Time (h)	Validation Time (h)
ResNet 1	20	0.3219	0.3579	90.76	89.77	75.00	0.18
ResNet 1	32	0.3418	0.3668	90.40	89.80	133.76	0.30
ResNet 1	74	0.3586	0.3771	90.62	89.88	278.83	0.55
ResNet 2	29	0.3491	0.3770	90.38	89.86	173.74	0.29
ResNet 2	47	0.3523	0.3621	90.40	90.56	291.30	0.55
ResNet 2	110	0.3738	0.3895	90.07	90.05	671.26	0.96

We compared the various ResNet-model-layer combinations in terms of training accuracy (Figure 12), validation accuracies (Figure 13), training loss (Figure 14), and validation loss (Figure 15). A consistent pattern emerged of ResNet 1 (32 layers) and ResNet 2 (47 layers) being the best models of the series.



**Figure 12.** Training accuracy curves for the various ResNet models.

The confusion matrices in Figure 16 show the comparison of each class's accuracy for the best proposed model (ResNet version 2 with 47 layers). The left confusion matrix shows the percentage accuracies for each class, and the right confusion matrix shows correctly classified grain images for each class. The results in the confusion matrix indicate that the classes C1 and C5 achieved higher accuracies as they had more grain images for the training.

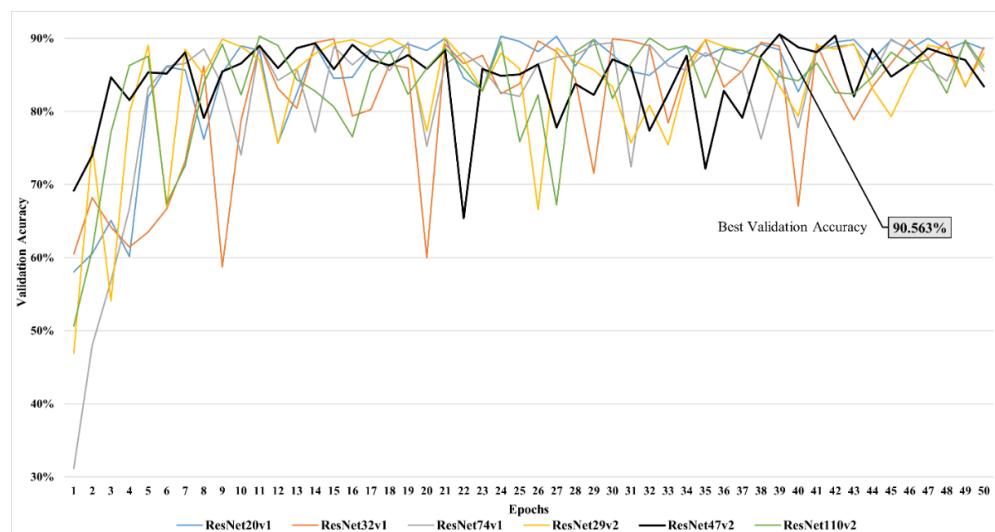


Figure 13. Validation accuracy curves for the various ResNet models.

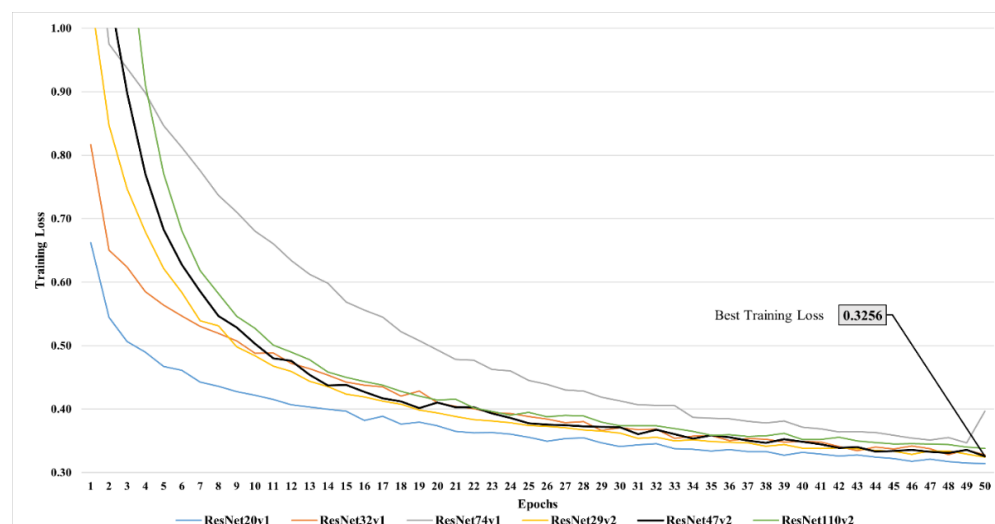


Figure 14. Training loss curves for the various ResNet models.

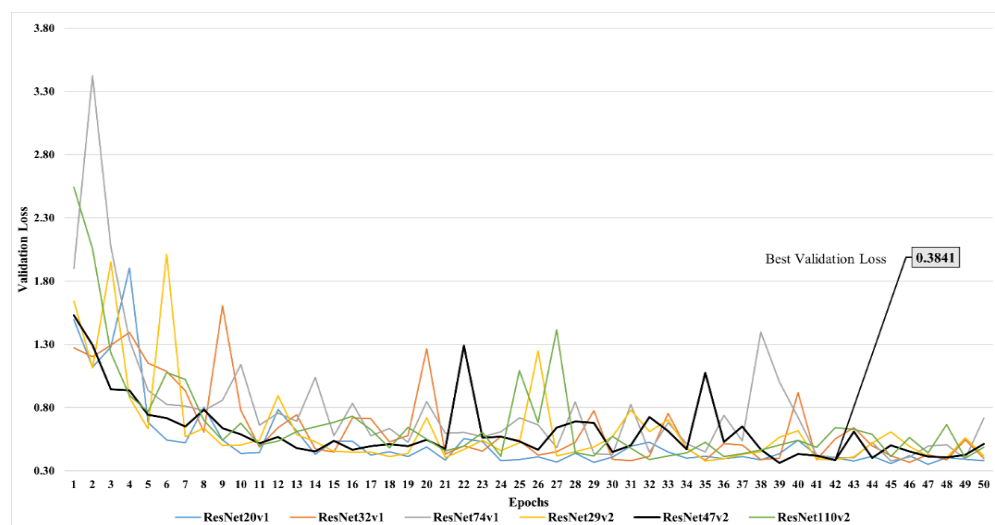


Figure 15. Validation loss curves for various ResNet models.

C1	C2	C3	C4	C5		C1	C2	C3	C4	C5	
96.61%	1.69%	1.04%	0.66%	0.00%	C1	1026	18	11	7	0	C1
16.89%	70.20%	10.43%	0.33%	2.15%	C2	102	424	63	2	13	C2
7.57%	1.44%	89.19%	0.18%	1.62%	C3	42	8	495	1	9	C3
13.09%	1.05%	4.19%	80.10%	1.57%	C4	25	2	8	153	3	C4
0.10%	0.00%	0.31%	0.00%	99.58%	C5	1	0	3	0	958	C5

**Figure 16.** Confusion matrix for the best ResNet version 2 (47 layers) model. Confusion matrix of percentage accuracies for each class (**left**) and confusion matrix of correctly classified images (**right**).

When we compared our ResNet 2 model (47 layers) with techniques published in the recent literature—using the published method on our grain data set—we observed that the superpixel-based grain segmentation and the ResNet 2 (47 layers) clearly outperformed the existing techniques and achieved the highest accuracy values (Table 5).

**Table 5.** Comparison of the proposed method with existing methods.

Reference	Methodology	Accuracy (%)
This paper	Modified superpixel grains with ResNet2 with 47 layers	90.56
Julien et al. (2019) [22]	Superpixel color features with random forests	89.00
Julien et al. (2019) [22]	Superpixel segmented grains with CNN	49.23
Brian et al. (2021) [50]	Neighborhood component analysis and cubic SVM	65.75
Brian et al. (2021) [50]	Neighborhood component analysis quadratic SVM	39.72%

## 5. Discussion and Conclusions

We presented two improved residual network architectures to automate the detection and count of individual mineral grains. These algorithms, ResNet 1 and ResNet 2, are modified versions of ResNet. We adopted the superpixel segmentation method and applied preprocessing techniques to provide the seed for the segmentation method, which made the data more appropriate for deep-learning algorithms. The ResNet 2 architecture with 47 layers produced the highest validation accuracy of 90.5%. To our knowledge, this is the highest reported accuracy achieved using deep-learning networks for this particular application.

Few papers explore the use of machine-learning techniques and deep-learning algorithms for the automatic recognition, classification, and counting of grain minerals; however, the existing approaches offer benchmarks against which we can compare our results. Our ResNet 1 and ResNet 2 outperformed the deep-learning algorithms LeNet, AlexNet, and GoogleNet in automatic grain detection and count application. Despite these very encouraging results, improvements must be made prior to the application of our deep-learning techniques in the field. The data set must be enhanced to eliminate problems of mislabeling, unbalanced data, and fusion. Moreover, the developed approach is limited by:

- The scarcity of mineral data sets. A key contribution of this work is the development of such a data set, because they are not readily available for grain mineral classification.
- Unbalanced data for different classes. In the developed data set, there was an unequal number of images available for each class.
- High-performance GPUs are required for training. We had access to a GPU system; however, the training step required a considerable amount of time to be performed.

Future work will include developing data sets for the purpose of grain mineral recognition and enhancing new and current methods to achieve a higher recognition rate with more mineral classes. These advances will include applying various image fusion and registration techniques to greatly improve the mapping of the original images with the labeled images. We will also explore other techniques for segmentation that may enhance accuracy. These may include the region-growing-based method, fuzzy C-means, and deep-learning segmentation.

**Author Contributions:** G.L.: Planning, methodology, analysis, experiments, initial draft writing. K.B.: Supervision, methodology, original draft writing and revision. J.M.: Methodology, review and editing. A.B.: Data collection. L.P.B.: Funding procurement, supervision, review and editing. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research is funded by a Fonds de Recherche du Québec—Nature et Technologies (FRQ-NT) grant to L.P.B. (Programme de recherche en partenariat sur le développement durable du secteur minier-II, grant number: 2020-MN-283346) with contributions from IOS Servives Géoscientifiques Inc.

**Data Availability Statement:** The data presented in this study can be requested from ghazanfar.latif1@uqac.ca.

**Acknowledgments:** We are thankful to the IOS Servives Géoscientifiques Inc. for providing technical support.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Jung, D.; Choi, Y. Systematic Review of Machine Learning Applications in Mining: Exploration, Exploitation, and Reclamation. *Minerals* **2021**, *11*, 148. [\[CrossRef\]](#)
2. Sengupta, S.; Dave, V. Predicting applicable law sections from judicial case reports using legislative text analysis with machine learning. *J. Comput. Soc. Sci.* **2021**, 1–14. [\[CrossRef\]](#)
3. Zantalis, F.; Koulouras, G.; Karabetsos, S.; Kandris, D. A Review of Machine Learning and IoT in Smart Transportation. *Future Internet* **2019**, *11*, 94. [\[CrossRef\]](#)
4. Latif, G.; Shankar, A.; Alghazo, J.; Kalyanasundaram, V.; Boopathi, C.S.; Jaffar, M.A. I-CARES: Advancing health diagnosis and medication through IoT. *Wirel. Netw.* **2019**, *26*, 2375–2389. [\[CrossRef\]](#)
5. Ali, D.; Frimpong, S. Artificial intelligence, machine learning and process automation: Existing knowledge frontier and way forward for mining sector. *Artif. Intell. Rev.* **2020**, *53*, 6025–6042. [\[CrossRef\]](#)
6. Chow, B.H.Y.; Reyes-Aldasoro, C.C. Automatic Gemstone Classification Using Computer Vision. *Minerals* **2021**, *12*, 60. [\[CrossRef\]](#)
7. Girard, R.; Tremblay, J.; Neron, A.; Longuépée, H. Automated Gold Grain Counting. Part 1: Why Counts Matter! *Minerals* **2021**, *11*, 337. [\[CrossRef\]](#)
8. Boivin, J.-F.; Bédard, L.P.; Longuépée, H. Counting a pot of gold: A till golden standard (AuGC-1). *J. Geochem. Explor.* **2021**, *229*, 106821. [\[CrossRef\]](#)
9. Plouffe, A.; McClenaghan, M.B.; Paulen, R.C.; McMartin, I.; Campbell, J.E.; Spirito, W.A. Processing of glacial sediments for the recovery of indicator minerals: Protocols used at the Geological Survey of Canada. *Geochem. Explor. Environ. Anal.* **2013**, *13*, 303–316. [\[CrossRef\]](#)
10. Xu, C.S.; Hayworth, K.J.; Lu, Z.; Grob, P.; Hassan, A.M.; García-Cerdán, J.G.; Niyogi, K.K.; Nogales, E.; Weinberg, R.J.; Hess, H.F. Enhanced FIB-SEM systems for large-volume 3D imaging. *eLife* **2017**, *6*, e25916. [\[CrossRef\]](#)
11. Nie, J.; Peng, W. Automated SEM-EDS heavy mineral analysis reveals no provenance shift between glacial loess and interglacial paleosol on the Chinese Loess Plateau. *Aeolian Res.* **2014**, *13*, 71–75. [\[CrossRef\]](#)
12. Akcil, A.; Koldas, S. Acid Mine Drainage (AMD): Causes, treatment and case studies. *J. Clean. Prod.* **2006**, *14*, 1139–1145. [\[CrossRef\]](#)
13. Hudson-Edwards, K.A. Sources, mineralogy, chemistry and fate of heavy metal-bearing particles in mining-affected river systems. *Miner. Mag.* **2003**, *67*, 205–217. [\[CrossRef\]](#)
14. Hobbs, D.W. 4 Structural Effects and Implications and Repair. In *Alkali-Silica Reaction in Concrete*; Thomas Telford Publishing: London, UK, 1988; pp. 73–87. [\[CrossRef\]](#)
15. Lawrence, P.; Cyr, M.; Ringot, E. Mineral admixtures in mortars effect of type, amount and fineness of fine constituents on compressive strength. *Cem. Concr. Res.* **2005**, *35*, 1092–1105. [\[CrossRef\]](#)
16. Erlich, E.I.; Hausel, W.D. *Diamond Deposits: Origin, Exploration, and History of Discovery*; SME: Littleton, CO, USA, 2003.
17. Towie, N.J.; Seet, L.H. Diamond laboratory techniques. *J. Geochem. Explor.* **1995**, *53*, 205–212. [\[CrossRef\]](#)



18. Chen, Z.; Liu, X.; Yang, J.; Little, E.C.; Zhou, Y. Deep learning-based method for SEM image segmentation in mineral characterization, an example from Duvernay Shale samples in Western Canada Sedimentary Basin. *Comput. Geosci.* **2020**, *138*, 104450. [\[CrossRef\]](#)
19. Hyder, Z.; Siau, K.; Nah, F. Artificial Intelligence, Machine Learning, and Autonomous Technologies in Mining Industry. *J. Database Manag.* **2019**, *30*, 67–79. [\[CrossRef\]](#)
20. Dalm, M.; Buxton, M.W.; van Ruitenbeek, F.; Voncken, J.H. Application of near-infrared spectroscopy to sensor based sorting of a porphyry copper ore. *Miner. Eng.* **2014**, *58*, 7–16. [\[CrossRef\]](#)
21. McCoy, J.; Auret, L. Machine learning applications in minerals processing: A review. *Miner. Eng.* **2018**, *132*, 95–109. [\[CrossRef\]](#)
22. Maitre, J.; Bouchard, K.; Bedard, L. Mineral grains recognition using computer vision and machine learning. *Comput. Geosci.* **2019**, *130*, 84–93. [\[CrossRef\]](#)
23. Makvandi, S.; Pagé, P.; Tremblay, J.; Girard, R. Exploration for Platinum-Group Minerals in Till: A New Approach to the Recovery, Counting, Mineral Identification and Chemical Characterization. *Minerals* **2021**, *11*, 264. [\[CrossRef\]](#)
24. Kim, C.S. Characterization and speciation of mercury-bearing mine wastes using X-ray absorption spectroscopy. *Sci. Total Environ.* **2000**, *261*, 157–168. [\[CrossRef\]](#)
25. Baklanova, O.; Shvets, O. Cluster analysis methods for recognition of mineral rocks in the mining industry. In Proceedings of the 2014 4th International Conference on Image Processing Theory, Tools and Applications (IPTA), Paris, France, 14–17 October 2014. [\[CrossRef\]](#)
26. Iglesias, J.C.A.; Gomes, O.D.F.M.; Paciornik, S. Automatic recognition of hematite grains under polarized reflected light microscopy through image analysis. *Miner. Eng.* **2011**, *24*, 1264–1270. [\[CrossRef\]](#)
27. Gomes, O.D.F.M.; Iglesias, J.C.A.; Paciornik, S.; Vieira, M.B. Classification of hematite types in iron ores through circularly polarized light microscopy and image analysis. *Miner. Eng.* **2013**, *52*, 191–197. [\[CrossRef\]](#)
28. Figueroa, G.; Moeller, K.; Buhot, M.; Gloy, G.; Haberla, D. Advanced Discrimination of Hematite and Magnetite by Automated Mineralogy. In Proceedings of the 10th International Congress for Applied Mineralogy (ICAM), Trondheim, Norway, 1–5 August 2011; Springer: Berlin/Heidelberg, Germany, 2012; pp. 197–204.
29. Iglesias, J.C.; Santos, R.B.M.; Paciornik, S. Deep learning discrimination of quartz and resin in optical microscopy images of minerals. *Miner. Eng.* **2019**, *138*, 79–85. [\[CrossRef\]](#)
30. Philander, C.; Rozendaal, A. The application of a novel geometallurgical template model to characterise the Namakwa Sands heavy mineral deposit, West Coast of South Africa. *Miner. Eng.* **2013**, *52*, 82–94. [\[CrossRef\]](#)
31. Sylvester, P.J. *Use of the Mineral Liberation Analyzer (MLA) for Mineralogical Studies of Sediments and Sedimentary Rocks*; Mineralogical Association of Canada: Quebec City, QC, USA, 2012; Volume 1, pp. 1–16.
32. Goldstein, J. *Practical Scanning Electron Microscopy: Electron and Ion Microprobe Analysis*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012.
33. Potts, P.J.; Bowles, J.F.; Reed, S.J.; Cave, R. *Microprobe Techniques in the Earth Sciences*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012.
34. Safari, H.; Balcom, B.J.; Afrough, A. Characterization of pore and grain size distributions in porous geological samples—An image processing workflow. *Comput. Geosci.* **2021**, *156*, 104895. [\[CrossRef\]](#)
35. Duarte-Campos, L.; Wijnberg, K.M.; Gálvez, L.O.; Hulscher, S.J. Laser particle counter validation for aeolian sand transport measurements using a highspeed camera. *Aeolian Res.* **2017**, *25*, 37–44. [\[CrossRef\]](#)
36. Cox, M.R.; Budhu, M. A practical approach to grain shape quantification. *Eng. Geol.* **2008**, *96*, 1–16. [\[CrossRef\]](#)
37. Latif, G.; Iskandar, D.A.; Alghazo, J.; Butt, M.M. Brain MR Image Classification for Glioma Tumor detection using Deep Convolutional Neural Network Features. *Curr. Med. Imaging* **2021**, *17*, 56–63. [\[CrossRef\]](#)
38. Alghazo, J.; Latif, G.; Elhassan, A.; Alzubaidi, L.; Al-Hmouz, A.; Al-Hmouz, R. An Online Numeral Recognition System Using Improved Structural Features—A Unified Method for Handwritten Arabic and Persian Numerals. *J. Telecommun. Electron. Comput. Eng.* **2017**, *9*, 33–40.
39. Wang, Y.; Balmos, A.D.; Layton, A.W.; Noel, S.; Ault, A.; Krogmeier, J.V.; Buckmaster, D.R. *An Open-Source Infrastructure for Real-Time Automatic Agricultural Machine Data Processing*; American Society of Agricultural and Biological Engineers: St. Joseph, MI, USA, 2017.
40. Wójcik, M.; Brinkmann, P.; Zdunek, R.; Riebe, D.; Beitz, T.; Merk, S.; Cieřlik, K.; Mory, D.; Antończak, A. Classification of Copper Minerals by Handheld Laser-Induced Breakdown Spectroscopy and Nonnegative Tensor Factorisation. *Sensors* **2020**, *20*, 5152. [\[CrossRef\]](#) [\[PubMed\]](#)
41. Hao, H.; Guo, R.; Gu, Q.; Hu, X. Machine learning application to automatically classify heavy minerals in river sand by using SEM/EDS data. *Miner. Eng.* **2019**, *143*, 105899. [\[CrossRef\]](#)
42. Vos, K.; Vandenberghe, N.; Elsen, J. Surface textural analysis of quartz grains by scanning electron microscopy (SEM): From sample preparation to environmental interpretation. *Earth-Sci. Rev.* **2014**, *128*, 93–104. [\[CrossRef\]](#)
43. Sundaresan, V.; Zamboni, G.; Le Heron, C.; Rothwell, P.M.; Husain, M.; Battaglini, M.; De Stefano, N.; Jenkinson, M.; Griffanti, L. Automated lesion segmentation with BIANCA: Impact of population-level features, classification algorithm and locally adaptive thresholding. *NeuroImage* **2019**, *202*, 116056. [\[CrossRef\]](#) [\[PubMed\]](#)
44. Li, Z.; Chen, J. Superpixel Segmentation using Linear Spectral Clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1356–1363.

- 
45. Hechler, E.; Oberhofer, M.; Schaeck, T. *Deploying AI in the Enterprise IT Approaches for Design, DevOps, Governance, Change Management, Blockchain, and Quantum Computing*; Springer: Berkeley, CA, USA, 2020.
  46. Alghmgham, D.A.; Latif, G.; Alghazo, J.; Alzubaidi, L. Autonomous Traffic Sign (ATSR) Detection and Recognition using Deep CNN. *Procedia Comput. Sci.* **2019**, *163*, 266–274. [[CrossRef](#)]
  47. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2012**, *60*, 84–90. [[CrossRef](#)]
  48. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. *arXiv* **2014**, arXiv:1409.4842v1.
  49. Wu, Z.; Shen, C.; Van Den Hengel, A. Wider or deeper: Revisiting the resnet model for visual recognition. *Pattern Recognit.* **2019**, *90*, 119–133. [[CrossRef](#)]
  50. Sinaice, B.; Owada, N.; Saadat, M.; Toriya, H.; Inagaki, F.; Bagai, Z.; Kawamura, Y. Coupling NCA Dimensionality Reduction with Machine Learning in Multispectral Rock Classification Problems. *Minerals* **2021**, *11*, 846. [[CrossRef](#)]