

Article

Convolutional Neural Networks for Off-Line Writer Identification Based on Simple Graphemes

Marco Mora ^{1,2,*} , José Naranjo-Torres ^{1,†}  and Verónica Aubin ^{3,†} 

¹ Laboratory of Technological Research in Pattern Recognition, Faculty of Engineering Science, Universidad Católica del Maule, Talca 3480112, Maule, Chile; jnaranjo@ucm.cl

² Department of Computer Science and Industries, Faculty of Engineering Science, Universidad Católica del Maule, Talca 3480112, Maule, Chile

³ Department of Engineering and Technological Research, Universidad Nacional de La Matanza, San Justo B1754JEC, Provincia de Buenos Aires, Argentina; vaubin@unlam.edu.ar

* Correspondence: mmora@ucm.cl

† These authors contributed equally to this work.

Received: 29 September 2020; Accepted: 5 November 2020; Published: 11 November 2020



Abstract: The writer's identification/verification problem has traditionally been solved by analyzing complex biometric sources (text pages, paragraphs, words, signatures, etc.). This implies the need for pre-processing techniques, feature computation and construction of also complex classifiers. A group of simple graphemes ("S", "∩", "C", "∩" and "U") has been recently introduced in order to reduce the structural complexity of biometric sources. This paper proposes to analyze the images of simple graphemes by means of Convolutional Neural Networks. In particular, the AlexNet, VGG-16, VGG-19 and ResNet-18 models are considered in the learning transfer mode. The proposed approach has the advantage of directly processing the original images, without using an intermediate representation, and without computing specific descriptors. This allows to dramatically reduce the complexity of the simple grapheme processing chain and having a high hit-rate of writer identification performance.

Keywords: writer identification; off-line analysis; simple graphemes; convolutional neural networks

1. Introduction

There are different biometric features that allow the verification or identification of people, among them is writing. The rhythm of writing, which is unrepeatable and unique, captures particular graphic characteristics in the text which allow the identification of the author. People recognition through the analysis of handwritten texts is widely used in different tasks such as identifying authorship, detecting forgeries, fraud, threats and theft, in documents of different types such as holographic wills, letters, checks, and so forth [1].

Most state-of-the-art works analyze complex text structures to extract features, such as full pages, text and paragraphs [2–6], words [7–9] and signatures [10–12]. Working with very complex sources in order to obtain a high verification ratio results in complexity throughout the entire processing sequence: developing sophisticated segmentation algorithms for the region of interest, complexity in the automatic computation of descriptors to represent the original data with low dimensionality and high execution times for the algorithms.

Contrary to the more traditional literature characterized by the complexity of the structures used, a new approach begins to consider simple elements of handwritten text to solve the problem of writer verification. Along these lines, in Reference [13] a new database is proposed containing 6 remarkably simple grapheme types: "e", "S", "∩", "C", "∩" and "U". In addition, a new descriptor is introduced to represent the texture of the handwritten strokes (relative position of the minimum

gray value points within the stroke) and successful verification tests are performed with a Support Vector Machine (SVM) based classifier. In Reference [14], it is proposed to represent the texture of simple graphemes by means of B-Spline transformation coefficients and classifiers based on banks of SVMs. In Reference [15], the character “e” is excluded because it presents crosses in its structure, which generates complexity in the computation of descriptors, the Local Binary Patterns (LBP) are introduced to represent the surface of the simple graphemes, and a classifier based on SVM is built. Recently, in Reference [16], it was proposed to simplify the structure of the classifier and reduce training time using Neural Networks of Extreme Learning (ELM). In the aforementioned works, preprocessing and transformation of the original image are performed, descriptors representing the surface texture of the grapheme are computed, and classifiers are constructed for the verification of the writer.

In order to simplify the pipeline of simple graphemes processing, without to perform pre-processing (working directly with the original image), without to compute descriptors, and to achieve a high rate of writer identification accuracy, this paper proposes to analyze the image of the Simple Grapheme using Convolutional Neural Networks (CNN). The advantages of this approach are as follows:

- Directly working with the original image without making any transformations.
- Biometric features are obtained automatically through CNN filters.
- The use of CNN allows a high success rate in the test set because the constructed classifiers correspond to highly non-linear transformations.
- There are consolidated frameworks for the implementation of CNN networks [17,18], which use high-performance computing techniques (multi-core and GPUs) to reduce network training time.

In this work, experiments are performed with the network models AlexNet [19], VGG (VGG-16 and VGG-19) [20] and ResNet (ResNet-18) [21]. AlexNet and VGGs networks can be considered classic convolutional neural networks, as they follow the basic serial connection scheme, that is, a series of convolutions, pooling, activation layers and finally some completely connected classification layers. The idea of the ResNet models (ResNet-18/50/101), is to use residual blocks of direct access connections, with double or triple layer jumps where the input is not weighted and it is passed to a deeper layer. In this work, this group of CNN networks is adopted because they present a good compromise between performance, structural complexity and training time.

The structure of this paper is as follows. Section 2 presents an overview of the simple grapheme database and its traditional representation. Section 3 presents the CNN models adopted in this research. Section 4 shows the experiments performed. Finally, section 5 presents the conclusions of this paper.

2. An Overview of Simple Graphemes

Simple graphemes were recently reported in Reference [15]. This repository contains five types of simple graphemes: “S”, “∩”, “C”, “~” and “U”, for 50 writers, with 100 samples of each simple grapheme per writer. The images are 24-bit color, 800 × 800 pixels in size, with a scanner resolution of 1200 dpi. Figure 1 shows sample images of the simple graphemes contained in the image repository.

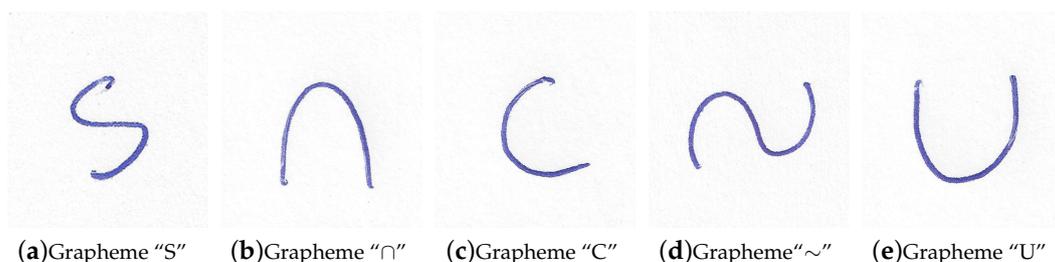


Figure 1. Simple grapheme images.

The images in this repository have a resolution of 1200 dpi, this is due to the fact that the simple character methodology used by Aubin et al. [15] is based on texture, and in order to have enough information, higher resolution is required to provide more detail of the stroke texture, which is enough to extract biometric information from small text elements. It should be noted that the public databases resolution of handwritten text (IAM [22], CEDAR [23], CVL [24], RIMES [25]) is 300 dpi. This low resolution is due to the fact that traditional databases were not designed to analyze small elements of handwritten text.

As Figure 1 shows, the image of the grapheme has many white pixels (background pixels) that contain no information. In order to obtain an image that considers only the pixels of the grapheme, a rectified image is constructed that consists of a “stretched” version of the grapheme [15].

3. Convolutional Neural Network Models for Simple Grapheme Analysis

The CNNs are capable of automatically extracting the characteristics of images [26], making them suitable for the study of images [27]. The CNN typical architecture is composed in the following way (illustrated in Figure 2):

- Convolutional Layer: It is a set of convolutional filters which activate the characteristics of the image.
- Layer of activation function: It is a non-linear activation function.
- Subsampling Layer or pooling layer: It reduces the dimension of the feature banks at the output of the convolutional layer.
- Fully connected Layer: It flattens the output of the previous layers by converting the output to 1D.
- Softmax Layer: It gives the probabilities of each category as established in the database at the beginning to perform the classification.

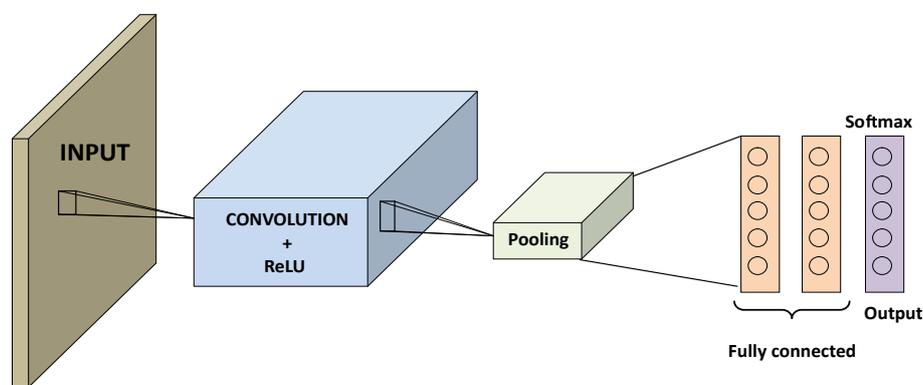


Figure 2. Basic architecture of a Convolutional Neural Network (CNN).

There are CNNs previously trained for image classification that have learned to extract characteristics and information from the images, thus using them as a starting point to learn a new task. Most of these CNNs were trained using the ImageNet database [28], which is used in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [29]. The three main uses of pre-trained CNNs are shown in Table 1.

Table 1. Applications of pre-trained CNNs [17].

Purpose	Description
Transfer Learning	Fine-Tune on new dataset
Feature Extraction	Use of pre-trained network as a features extractor
Classification	Apply pre-trained networks directly to classifications problems

However, because the original and rectified graphemes are very different from the images included in the Imagenet database, the graphemes cannot be classified directly using the pretrained CNNs. Consequently, a learning transfer process invariably takes place. This process consists of properly adjusting and training the previously trained CNN with the new images. The idea is usually to adjust the CNN output layers keeping the rest of the network unchanged and taking the pre-trained weights. Figure 3 illustrates a simplified diagram of the learning transfer process with pre-trained CNNs.

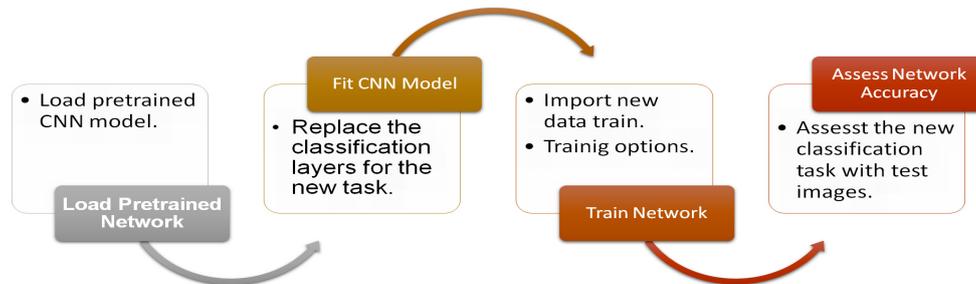


Figure 3. Simplified diagram of the learning transfer process.

This paper adopts CNN models widely known in the literature:

- AlexNet [19]: was one of the first deep networks in and a significant step in the development of CNNs. It is composed of 5 convolutional layers followed by 3 fully connected layers.
- VGG [20] versions VGG-16 and VGG-19, Developed by the Visual Geometry Group (VGG) of the University of Oxford, it is an AlexNet enhancement by replacing large kernel-sized filters with multiple 3×3 kernel size filters one after another, increasing network depth and thus being able to learn more complex features.
- ResNet (ResNet-18) [21], is an innovation over previous architecture, solving many of the problems of deep networks. It uses residual blocks of direct access connections, managing to reduce the number of parameters to be trained, with a good compromise between performance, structural complexity and training time.

Table 2 shows the general characteristics of these networks: depth, size of the network, number of parameters and dimension of the input image. Figure 4 shows the architecture of the AlexNet, VGG-16, VGG-19 and ResNet-18 networks. The description of the elements that form the blocks of this figure is as follows:

- Conv: The size of the convolutional filters.
- @: The number of filters to apply.
- s: The stride of the filter over the image.
- ReLU: The activation function at the output of the convolutional filters
- MaxPool: The subsampling operation with the filter dimension.

Table 2. Parameters and dimensions of pre-trained CNNs used [17].

Network	Depth	Size	Parameters (Millions)	Image Input Size
AlexNet	8	227 MB	61.0	227-by-227
VGG-16	16	515 MB	138	224-by-224
VGG-19	19	535 MB	144	224-by-224
ResNet-18	18	47 MB	25.6	224-by-224

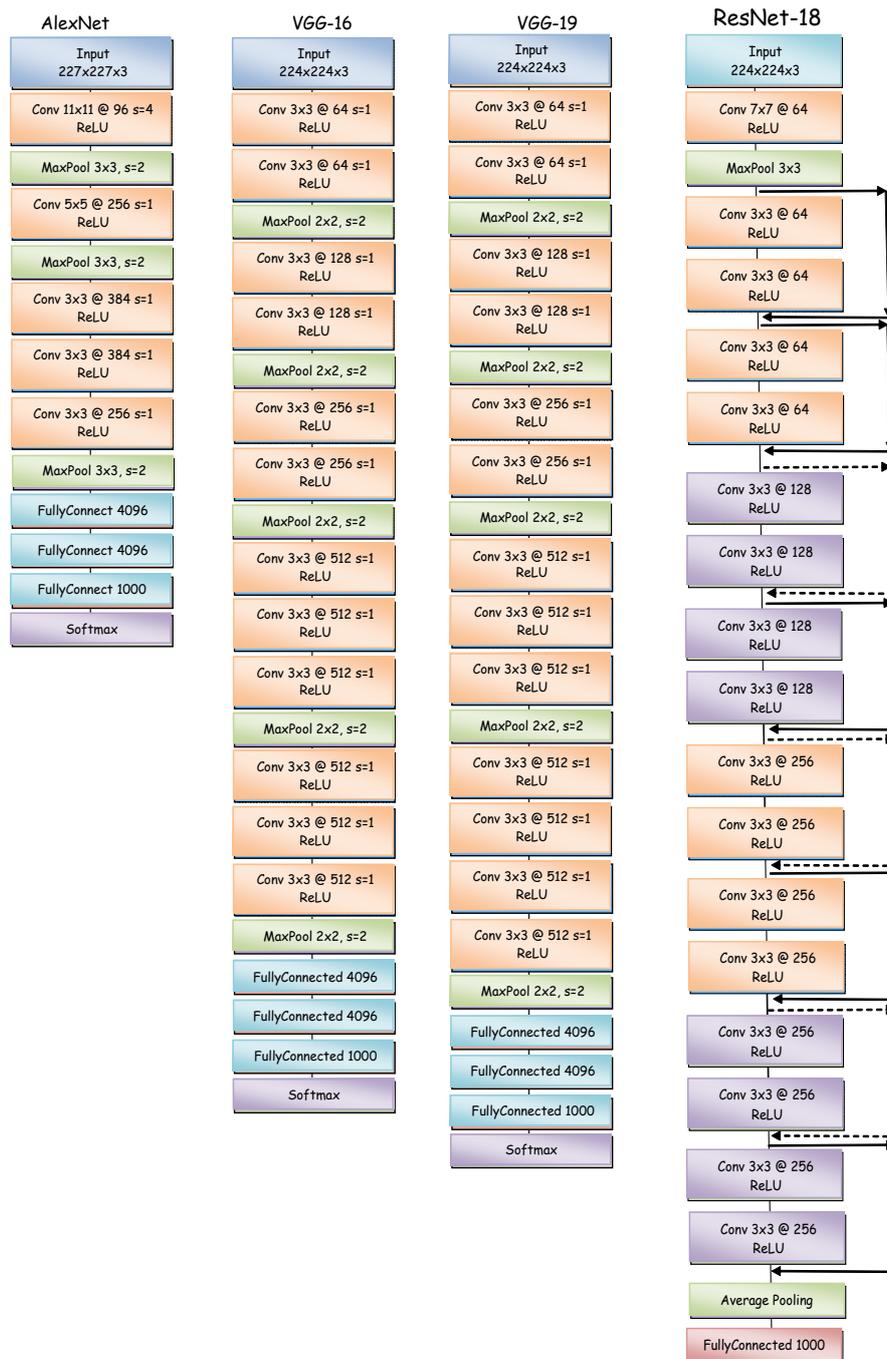


Figure 4. General architecture of AlexNet, VGG-16, VGG-19 and ResNet-18.

4. Experiments with Convolutional Neural Networks

This section describes the experiments carried out with simple graphemes and the pretrained CNNs AlexNet, VGG-16, VGG-19 and ResNet-18, performing learning transfer modality.

Two variants of the grapheme image are considered for the experiments. The first one consists of the rectified grapheme, which is the approach used in most articles that work with simple graphemes. The second one consists of the RGB image of the original grapheme, in order to carry out experiments without transforming the original image. All the images used in this article make up the LITRP-SGDB database (LITRP- Simple Grapheme Data Base), which is available for download under the signature of a license agree form on the official site of the database <http://www.litrp.cl/repository.html#LITRP-SGDB>.

The rectification procedure is composed of a sequence of simple image processing operations that are graphically represented in Figure 5. The operations sequence is explained in detail in Reference [15], and can be summarized as follows:

- Convert the color image of the grapheme to a grayscale image using the achromatic component V, or V channel of the HSV model, generating a single channel grayscale image [30].
- Binarize the grayscale image of the V channel using the well-known Otsu algorithm [31].
- Obtain the morphological skeleton of the binary image of the H channel (white line in Figure 5b).
- Obtain the lines perpendicular to the morphological skeleton (black lines in Figure 5b).
- Finally, build an image with the pixels of the grayscale image that lie on the perpendicular lines.

Figure 5c shows the resulting image from the rectification process. It is important to note that this rectified image, being grayscale and not including background pixels, dramatically reduces the dimensionality of the color image of the original grapheme.

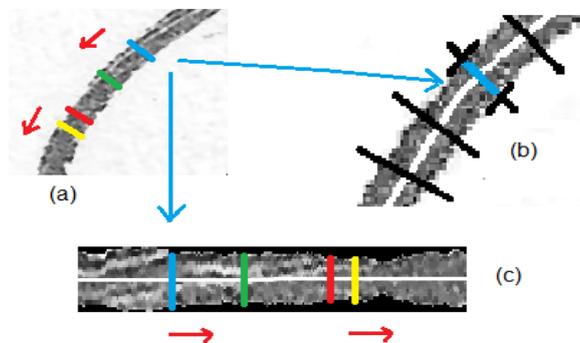


Figure 5. Rectification of Graphemes. (a) Segment of Original Simple Grapheme ; (b) Construction of Rectified Image; (c) Resulting Rectified Image.

In the neural networks constructed, the input corresponds to one of the two representations of the image and the output corresponds to a vector of 50 elements to represent the number of people that form the repository. In the training of the CNNs, 3 sets (Training, Validation and Test) are considered and balanced training sets are created per class. This process consists of: First taking the original set of images for a grapheme, dividing it randomly into the Training (80%), Validation (10%) and Test (10%) sets. Second, to avoid bias or imbalance in the network training, the Training set, the number of samples per person is equated to the smallest number that one of the people contains. This process is carried out for each grapheme individually, as well as for the rectified graphemes as for the original ones, in order to have sets with the same number of samples. Table 3 shows the number of samples from the training, validation and test sets by grapheme. The last row shows the composition of the sets grouping all the person’s graphemes.

Table 3. Datasets Training-Validation-Test.

Strokes	DataTraining		DataValidation	DataTest
	Samples	Samples/Persons		
“C”	2450	49	432	442
“~”	2000	40	418	424
“∩”	2000	40	428	432
“S”	1950	39	401	401
“U”	2050	41	420	427
Grouped graphemes	9750	195	2114	2119

To carry out the experiments, the MatLab Deep Learning Toolbox [17] was used, which provides a framework for designing and implementing deep neural networks with algorithms, pre-trained models, and applications. The experiments were carried out with a computer server of the following characteristics: 2x Intel Xeon Gold 6140 CPU @ 2.30 GHz, 36 total physical cores, 24.75 MB L3 Cache Memory 126 GB, Operatin System Debian GNU/Linux 10 (buster) Kernel 4.19.0-10-amd64 x86_64.

4.1. Experiments with Rectified Simple Grapheme Images

In this experiment, the images of the rectified graphemes obtained by Aubin et al. [15] are used, these are rectangular images in single channel grayscale of the form $w \times h \times 1$ with w much greater than h (50×700 approximately). Then, these images must be resized according to the corresponding CNN input layer, for AlexNet it is $227 \times 227 \times 3$ and for VGGs and ResNet it is $224 \times 224 \times 3$. The process consists of first resizing the rectangular image of a channel to a square image of $n \times n \times 1$ ($n = 224$ or $n = 227$). The grayscale image is then converted into an RGB image, using the same matrix for the three channels, as shown in Figure 6. This is to adapt the image to the input layer of the previously trained network.

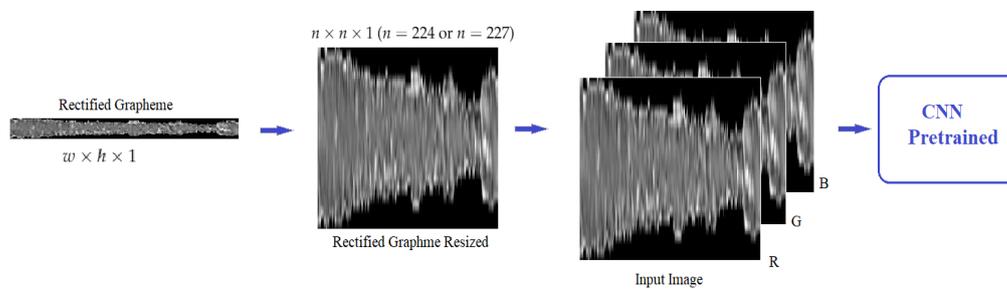


Figure 6. Rectified Simple Grapheme Resizing Process for Pretrained CNN Input.

Tables 4–7 show the experiments with AlexNet, VGG-16, VGG-19 and ResNet-18 networks, respectively. For each network, experiments have been carried out with a different number of epochs, but the table shows the smallest number of epochs that gives the best result on validation set (there comes a time when increasing the epochs does not improve the accuracy). Training and test time are expressed in seconds (s).

For the AlexNet, VGG-16 and VGG-19 networks considered, the rectified graphemes have an average yield close to 90%, the training took 80 epochs. For the ResNet-18 network, the accuracy results are lower than those of the previous ones, despite training with a more epochs (100 epochs) and from this point on, the increase in the number of epochs does not improve results. The moderate level of performance is explained because a lot of information is lost when transforming the image of the rectified grapheme to the input format of the CNN networks.

Table 4. Rectified Simple Grapheme-AlexNet (epoch = 80).

Strokes	Training		Validation		Test		
	Accuracy	Time (s)	Loss	Accuracy	Loss	Accuracy	Time (s)
“C”	98%	3.3887×10^3	0.0331	90%	0.4201	93%	0.6226
“~”	98%	2.7596×10^3	0.0307	86%	0.5061	89%	0.6210
“∩”	100%	2.7833×10^3	0.0147	91%	0.4291	92%	0.6262
“S”	100%	2.6778×10^3	0.0054	91%	0.4042	88%	0.6035
“U”	98%	2.8385×10^3	0.0245	92%	0.3004	93%	0.5738
Grouped	100%	2.1066×10^4	0.0271	90%	0.4195	90%	2.3997

Table 5. Rectified Simple Grapheme - VGG-16 (epoch = 80).

Strokes	Training			Validation		Test	
	Accuracy	Time (s)	Loss	Accuracy	Loss	Accuracy	Time (s)
"C"	100%	3.3154×10^4	0.0126	89%	0.6058	90%	2.6648
"~"	100%	2.6981×10^4	0.0016	90%	0.3463	87%	2.5534
"∩"	100%	2.6975×10^4	0.0008	90%	0.4733	90%	2.5969
"S"	100%	2.6196×10^4	0.0058	91%	0.4791	81%	2.4709
"U"	100%	2.7947×10^4	0.0026	89%	0.4293	90%	2.5750
"Grouped"	100%	1.9607×10^5	0.0004	90%	0.4573	90%	11.3092

Table 6. Rectified Simple Grapheme - VGG-19 (epoch = 80).

Strokes	Training			Validation		Test	
	Accuracy	Time (s)	Loss	Accuracy	Loss	Accuracy	Time (s)
"C"	100%	3.9447×10^4	0.0076	88%	0.6127	89%	3.0580
"~"	100%	3.2043×10^4	0.0002	89%	0.5779	90%	2.8303
"∩"	100%	3.2032×10^4	0.0101	90%	0.4322	90%	2.9161
"S"	100%	3.0957×10^4	0.0002	88%	0.4601	89%	2.6845
"U"	100%	3.3076×10^4	0.0029	88%	0.4250	88%	2.8272
"Grouped"	100%	2.3232×10^5	0.0052	90%	0.4330	91%	12.5139

Table 7. Rectified Simple Grapheme - ResNet-18 (epoch = 100).

Strokes	Training			Validation		Test	
	Accuracy	Time (s)	Loss	Accuracy	Loss	Accuracy	Time (s)
"C"	91%	1.0918×10^4	0.7492	73%	1.2662	77%	0.8327
"~"	92%	8.9056×10^3	1.1301	61%	1.7404	67%	0.9581
"∩"	97%	8.9344×10^3	0.9568	69%	1.5359	77%	0.9423
"S"	92%	8.6529×10^3	1.0892	62%	1.7339	65%	0.8589
"U"	91%	9.2122×10^3	1.0614	68%	1.5047	74%	0.8495
"Grouped"	98%	5.1184×10^4	0.2912	69%	0.9947	70%	3.3401

Figure 7 shows the test accuracy of applying the pre-trained CNNs to the rectified graphemes. It is observed that AlexNet, which is the simplest neural network, has the best results in general. Results get worse as network size increases.

Figure 8 shows the network training times for each rectified grapheme. The AlexNet and VGG16/VGG19 networks of similar architecture, as is known, the execution time increases as the depth of the network increases (epochs = 80). For ResNet-18, despite having trained with a greater number of epochs (epochs = 100) and being similar in depth to the VGGs, the training time is much less similar to that of AlexNet, which is due to the fact that it trains significantly fewer parameters than the other networks.

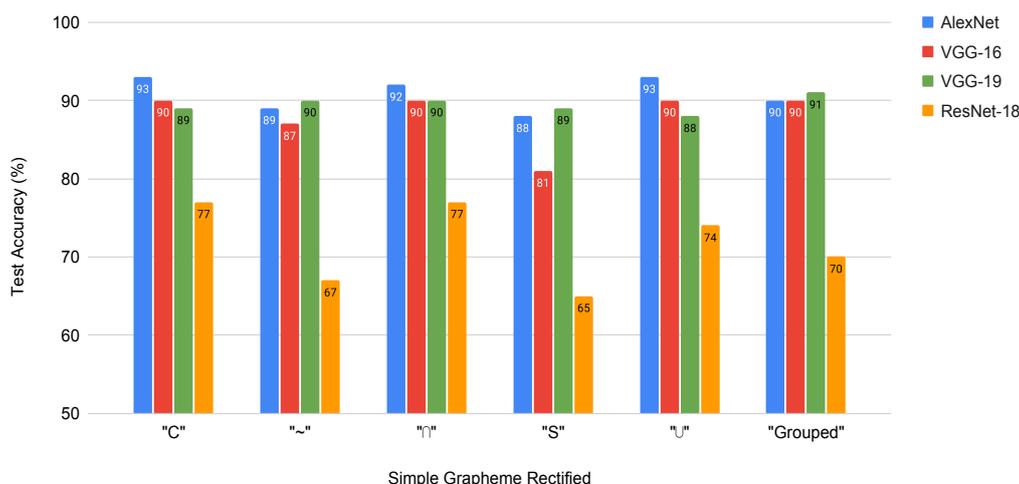


Figure 7. Rectified Simple Grapheme Test accuracy with AlexNet, VGG-16/VGG-19 and ResNet-18 networks.

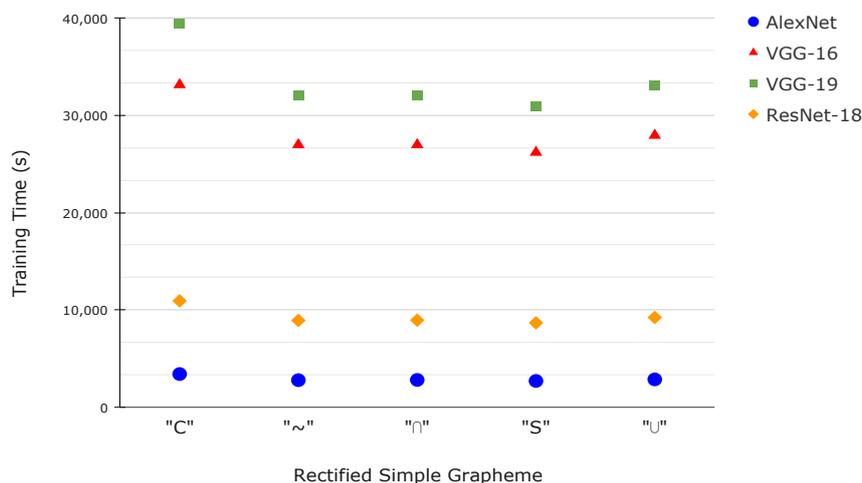


Figure 8. Rectified Simple Grapheme Training time of AlexNet, VGG-16/VGG-19 and ResNet-18 networks.

4.2. Experiments with Original Simple Grapheme Images

In order not to carry out the procedure of calculating the rectified grapheme and thus avoid this stage of the study process of the original graphemes, experiments are carried out with the RGB image of the original grapheme. The original image should be resized to match the size of the input image for each network, as the original dimension of the graphemes is about $800 \times 800 \times 3$. For AlexNet it resizes to $(227 \times 227 \times 3)$ and for VGG/ResNet to $(224 \times 224 \times 3)$. This is illustrated in Figure 9.

Tables 8–11 show the experiments with AlexNet, VGG-16, VGG-19 and ResNet-18 networks, respectively. Network training is performed by increasing the number of epochs until the error in the validation set reaches a minimum value. This process is carried out for all graphemes. For AlexNet, VGG-16 and VGG-19 networks the case of 50 epochs and for ResNet-18 the case of 80 epochs is shown. Likewise, the Tables show the execution times of the training of the CNNs and the classification times for each grapheme once the CNNs have been trained with the new images.

It can be observed that the results are very similar for the all networks, both for the individual graphemes and for the grouping of all the graphemes, ranging between 95% and 98%. An important result is that, for this type of images, a small network such as the VGG-16 is sufficient to obtain high performance. For instance, with the VGG-16 network, the characters that presented the best

performance are “S” and “~”, reaching a 98% hit-rate in the test set. Besides, it is observed that ResNet-18 with dimensions similar to VGGs but with different architecture achieves adequate performance but with substantially shorter training times.

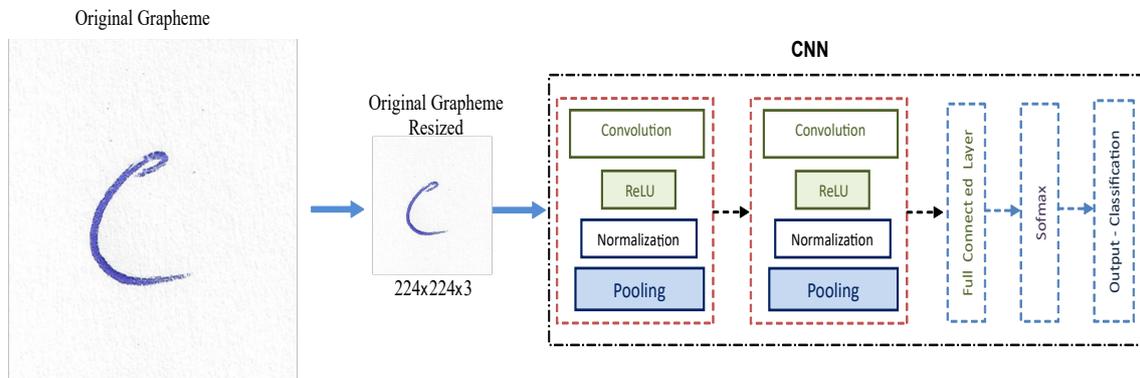


Figure 9. Simple Grapheme Resizing Process for Pretrained CNN Input.

Table 8. Original Simple Grapheme - Alexnet (epoch = 50).

Strokes	Training			Validation		Test	
	Accuracy	Time (s)	Loss	Accuracy	Loss	Accuracy	Time (s)
“C”	100%	2.5583×10^3	0.0072	92%	0.1592	95%	2.2047
“~”	100%	2.1007×10^3	0.0138	96%	0.0964	96%	2.1739
“∩”	98%	2.0193×10^3	0.0551	96%	0.1359	95%	2.3551
“S”	98%	2.0381×10^3	0.0261	96%	0.1269	95%	2.1078
“U”	100%	2.0381×10^3	0.0102	97%	0.1222	97%	2.1078
“Grouped”	98%	1.6506×10^3	0.0302	98%	0.0674	98%	8.3148

Table 9. Original Simple Grapheme - VGG-16 (epoch = 50).

Strokes	Training			Validation		Test	
	Accuracy	Time (s)	Loss	Accuracy	Loss	Accuracy	Time (s)
“C”	100%	2.1117×10^3	0.0053	96%	0.1232	97%	3.8204
“~”	100%	1.7282×10^3	0.0051	96%	0.1372	98%	3.4027
“∩”	98%	1.7286×10^3	0.0209	96%	0.1252	95%	3.4645
“S”	100%	1.6690×10^3	0.0134	99%	0.0495	98%	3.2534
“U”	100%	1.6690×10^3	0.0046	95%	0.0620	97%	3.2534
“Grouped”	100%	1.250×10^4	0.0063	98%	0.0657	98%	13.9850

Table 10. Original Simple Grapheme - VGG-19 (epoch = 50).

Strokes	Training			Validation		Test	
	Accuracy	Time (s)	Loss	Accuracy	Loss	Accuracy	Time (s)
“C”	100%	2.5171×10^4	0.0003	95%	0.1667	96%	3.8216
“~”	100%	2.0387×10^4	0.0019	99%	0.0551	96%	3.6044
“∩”	100%	2.0408×10^4	0.0032	97%	0.1133	95%	3.6994
“S”	100%	1.9715×10^4	0.0002	98%	0.0613	98%	3.4340
“U”	100%	2.1154×10^4	0.0017	98%	0.0912	97%	3.6035
“Grouped”	100%	1.4746×10^5	0.0045	99%	0.0368	98%	14.8049

Table 11. Original Simple Grapheme - ResNet-18 (epoch = 80).

Strokes	Training			Validation		Test	
	Accuracy	Time (s)	Loss	Accuracy	Loss	Accuracy	Time (s)
"C"	100%	9.3605×10^3	0.3718	96%	0.5731	97%	3.9659
"~"	100%	7.5222×10^3	0.4863	96%	0.6277	97%	2.1457
"∩"	98%	7.6248×10^3	0.5288	96%	0.6752	96%	3.7939
"S"	100%	7.3106×10^3	0.4845	96%	0.6772	96%	2.1489
"U"	100%	7.7572×10^3	0.4832	96%	0.6468	97%	2.0816
"Grouped"	100%	5.5884×10^4	0.0435	97%	0.1408	98%	8.5559

Figure 10 shows the test accuracy of applying the pre-trained CNNs to the original simple graphemes. It is observed that all the used networks achieve good results, being the VGG-16 the one with the best performance.

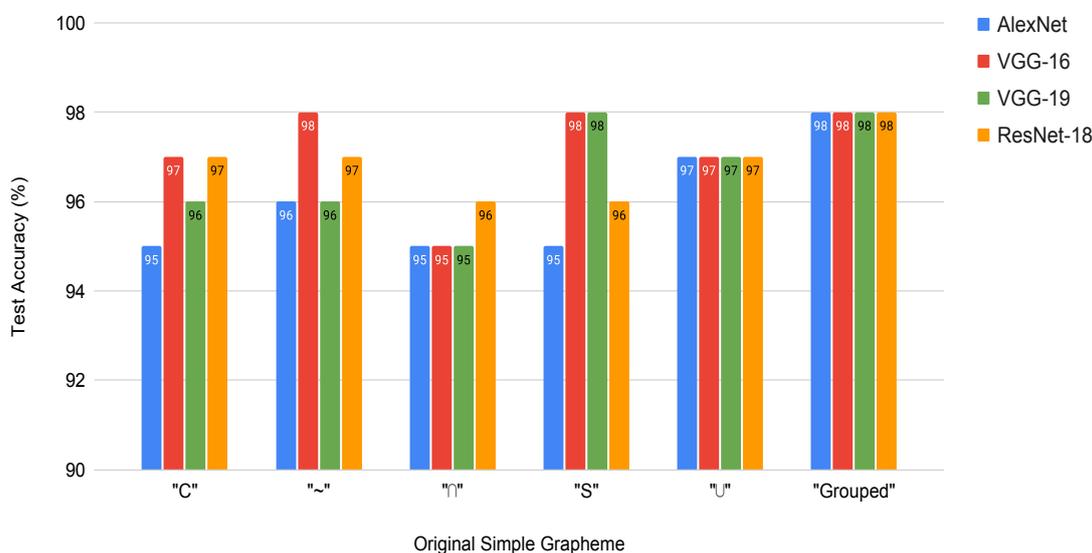


Figure 10. Original Simple Grapheme Test accuracy with AlexNet, VGG-16/VGG-19 and ResNet-18 networks.

Figure 11 shows the boxplots of the test set classification for all the networks used, in order to show the classification distribution of each grapheme by person. It is observed that the standard deviation of the classification results is very low for all networks, that the central tendency is high, and that there is very little presence of outliers. In particular, it is observed that the AlexNet network is the one with the greatest deviation. From these figures, it is possible to conclude a correct training and an adequate generalization (classification of the Test set).

Figure 12 shows the training times of the networks used in this work (third column of Tables 8–11). It is observed that for networks of the same type (AlexNet and VGGs) the training time increases as the depth of the network increases. The network that stands out is ResNet-18, having a depth similar to that of VGGs networks and being trained with a greater number of epochs, the training time is less. It can be objectively concluded that in time/accuracy ResNet is the network with the best performance.

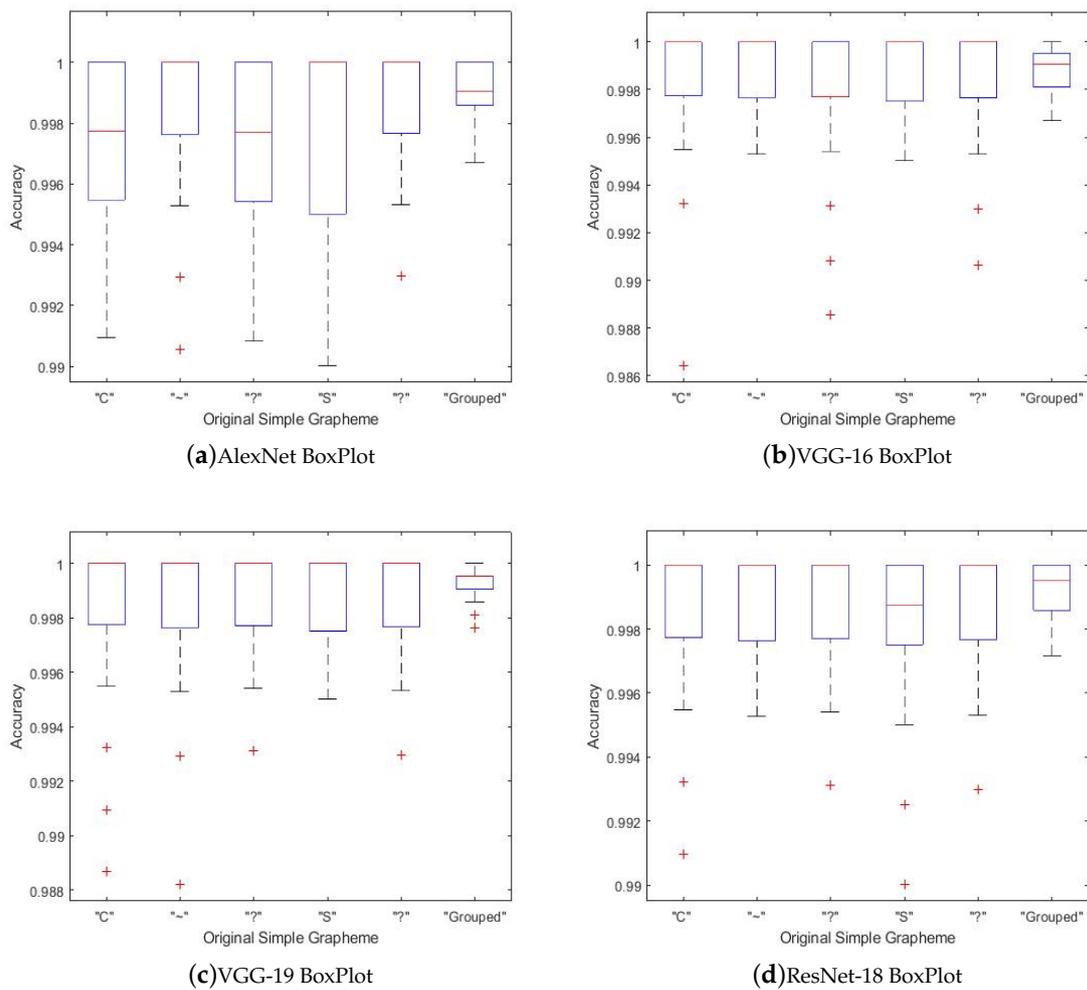


Figure 11. BoxPlots of Test Classification.

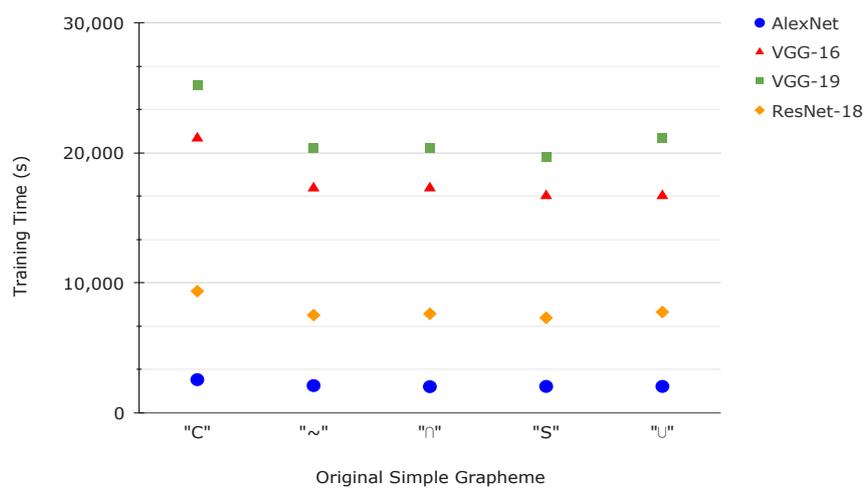


Figure 12. Original Simple Grapheme Training Time with AlexNet, VGG-16/VGG-19 and ResNet-18 networks.

4.3. Comparison with Other Approaches

Table 12 shows the results obtained in different works regarding writer verification on the repository of simple graphemes. The upper part of the table concentrates other approaches, and the lower part presents the results of this paper.

In Reference [13] a descriptor called Relative Position of the Minimum Gray Level Points (RPofMGLP) is proposed. The final descriptor is a vector whose elements correspond to the euclidean distance between the lower-gray value line and the considered reference edge. Said distance is measured over the perpendicular line that joins the point of the skeleton to the appropriate edge.

In Reference [14] a descriptor is proposed that corresponds to the coefficients of the B-Spline transformation of the signal of the descriptor RPofMGLP (BSC-RPofMGLP).

In Reference [15] various descriptors are proposed to represent the simple grapheme. The first one corresponds to the gray level of the morphological skeleton points (GLOfSK). It assumes that there is not a significant variation in the gray level perpendicularly to the skeleton. The second one corresponds to the Average Gray Level of the Skeleton Perpendicular Line (AGLOfSPL), which attempts to represent the horizontal and vertical variability of the gray levels with respect to the skeleton. The third one corresponds to the width of the grapheme, which was measured using the lines perpendicular to the skeleton (WofGra). Finally, it proposes the Local Binary Patterns of the grapheme surface (LBPofGra).

In Reference [16] the LBPofGra descriptor is considered but building classifiers based on Single Layer Extreme Learning Machine (ELM) networks and on Multiple Layer Extreme Learning Machine (ML-ELM).

Table 12 reinforces the idea that simple graphemes have enough biometric information for the writer verification. The best descriptors from other works are AGLOfSPL [15] and LBPofGra [15], both with an average performance of 98%. Processing the Original Graphemes through CNN gives a performance of 97% for the case of VGG-16. The CNN-based approach allows to obtain performance similar to the best results of other works but substantially simplifying the Simple Grapheme processing line.

Table 12. Comparison respect to other approaches.

Descriptor	Classifier	"C"	"~"	"∩"	"S"	"U"	Average	Grouped
RPofMGLP [13]	SVM	97%	97%	97%	98%	97%	97%	–
BSC-RPofMGLP [14]	SVM	97%	97%	97%	98%	97%	97%	–
GLOfSK [15]	SVM	83%	80%	82%	79%	83%	81%	–
AGLOfSPL [15]	SVM	98%	98%	98%	98%	98%	98%	–
WofGra [15]	SVM	96%	93%	96%	92%	94%	94%	–
LBPofGra [15]	SVM	98%	98%	98%	100%	98%	98%	–
LBPofGra [16]	ELM	91%	93%	91%	91%	92%	92%	90%
LBPofGra [16]	ML-ELM	95%	96%	96%	95%	95%	96%	92%
Original Grapheme	AlexNet	95%	96%	95%	95%	96%	96%	98%
	VGG-16	97%	98%	95%	98%	97%	97%	98%
	VGG-19	96%	96%	95%	98%	97%	96%	98%
	ResNet-18	97%	97%	96%	96%	97%	97%	98%

5. Conclusions

In this work, a scheme for processing simple graphemes for writer identification is presented. The approach is based on the use of convolutional neural networks.

The experimentation considered the image of rectified grapheme (traditional representation of simple graphemes) and the image of the original grapheme. The AlexNet, VGG-16, VGG-19 and

ResNet-18 models have been adopted, due to the fact that they present an adequate compromise between accuracy and training time.

The best results have been obtained with the original grapheme image and ResNet-18 Neural Network, considering the accuracy and time trade-off. Using ResNet-18, an average hit-rate of 97% has been achieved considering individual graphemes, and 98% of hit-rate considering grouped graphemes. The results show a high level of performance of the original grapheme, without the need to transform the image or compute specific descriptors, drastically reducing the complexity of the simple grapheme processing chain.

Author Contributions: M.M. Conceptualization, methodology, software, J.N.-T. and V.A.; software, M.M., J.N.-T. and V.A.; formal analysis, M.M., J.N.-T. and V.A.; investigation, M.M., J.N.-T. and V.A.; writing—original draft preparation, M.M., J.N.-T. and V.A.; writing—review and editing, M.M. and J.N.-T.; project administration, M.M.; funding acquisition, M.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Innovation Fund for Competitiveness—FIC, Government of Maule, Chile—Project Transfer Development Equipment Estimation Quality of Raspberry, code 40.001.110-0 (Esta investigación fue financiada por el Fondo de Innovación para la Competitividad—FIC, Gobierno de Maule, Chile—Proyecto de Transferencia de Desarrollo de Equipo de Estimación de Calidad de la Frambuesa, código 40.001.110-0).

Acknowledgments: The authors thank the Laboratory of Technological Research in Recognition of Patterns (www.litrp.cl) of the Universidad Católica del Maule, Chile, for providing the computer servers where the experiments have been carried out.

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

Abbreviations

The following abbreviations are used in this manuscript:

SVM	Support Vector Machine
LBP	Local Binary Pattern
ELM	Single Layer Extreme Learning Machine Neural Network
ML-ELM	Multiple Layer Extreme Learning Machine Neural Network
CNN	Convolutional Neural Network
VGG-16	VGG-16 Convolutional Neural Network Model
VGG-19	VGG-19 Convolutional Neural Network Model
AlexNet	AlexNet Convolutional Neural Network Model
ResNet-18	Residual Convolutional Neural Network Model
HSV	Hue-Saturation-Value Color Model
RPofMGLP	Relative Position of the Minimum Gray Level Points
BSC-RPofMGLP	B-Spline Coefficient of Relative Position of the Minimum Gray Level Points Signal
GLofSK	Gray level of the Skeleton Points
AGLofSPL	Average Gray Level of the Skeleton Perpendicular Line
WofGra	Width of the Grapheme
LBPofGra	Local Binary Pattern of the Grapheme Surface.

References

1. Morris, R.; Morris, R.N. *Forensic Handwriting Identification: Fundamental Concepts and Principles*; Academic Press: London, UK, 2000.
2. Marcelli, A.; Parziale, A.; De Stefano, C. Quantitative evaluation of features for forensic handwriting examination. In Proceedings of the 2015 13th International Conference on Document Analysis and Recognition (ICDAR), Tunis, Tunisia, 23–26 August 2015; pp. 1266–1271.
3. Bulacu, M.; Schomaker, L. Text-independent writer identification and verification using textural and allographic features. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 701–717. [[CrossRef](#)] [[PubMed](#)]
4. Hanusiak, R.K.; Oliveira, L.S.; Justino, E.; Sabourin, R. Writer verification using texture-based features. *Int. J. Doc. Anal. Recognit. (IJ DAR)* **2012**, *15*, 213–226. [[CrossRef](#)]

5. Marcelli, A.; Parziale, A.; Santoro, A. Modelling visual appearance of handwriting. In *International Conference on Image Analysis and Processing*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 673–682.
6. Christlein, V.; Bernecker, D.; Hönig, F.; Maier, A.; Angelopoulou, E. Writer Identification Using GMM Supervectors and Exemplar-SVMs. *Pattern Recognit.* **2017**, *63*, 258–267. [[CrossRef](#)]
7. Vásquez, J.L.; Ravelo-García, A.G.; Alonso, J.B.; Dutta, M.K.; Travieso, C.M. Writer identification approach by holistic graphometric features using off-line handwritten words. *Neural Comput. Appl.* **2018**, *32*, 1–14. [[CrossRef](#)]
8. Chu, J.; Shaikh, M.A.; Chauhan, M.; Meng, L.; Srihari, S. Writer Verification using CNN Feature Extraction. In *Proceedings of the 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Niagara Falls, NY, USA, 5–8 August 2018; pp. 181–186.
9. He, S.; Schomaker, L. Deep adaptive learning for writer identification based on single handwritten word images. *Pattern Recognit.* **2019**, *88*, 64–74. [[CrossRef](#)]
10. Plamondon, R.; Lorette, G. Automatic signature verification and writer identification—the state of the art. *Pattern Recognit.* **1989**, *22*, 107–131. [[CrossRef](#)]
11. Impedovo, D.; Pirlo, G.; Plamondon, R. Handwritten signature verification: New advancements and open issues. In *Proceedings of the 2012 International Conference on Frontiers in Handwriting Recognition*, Bari, Italy, 18–20 September 2012; pp. 367–372.
12. Hafemann, L.G.; Sabourin, R.; Oliveira, L.S. Offline handwritten signature verification—Literature review. In *Proceedings of the 2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*, Montreal, QC, Canada, 28 November–1 December 2017; pp. 1–8.
13. Aubin, V.; Mora, M. A new descriptor for person identity verification based on handwritten strokes off-line analysis. *Exp. Syst. Appl.* **2017**, *89*, 241–253. [[CrossRef](#)]
14. Aubin, V.; Mora, M.; Santos, M. A new descriptor for writer identification based on B-Splines. In *Proceedings of the 8th International Conference of Pattern Recognition Systems (ICPRS 2017)*, Madrid, Spain, 11–13 July 2017; pp. 1–5.
15. Aubin, V.; Mora, M.; Santos, M. Off-line Writer Verification based on Simple Graphemes. *Pattern Recognit.* **2018**, *79*, 414–426. [[CrossRef](#)]
16. Vasquez-Coronel, A.; Mora, M.; Aubin, V. Writer Verification based on Simple Graphemes and Extreme Learning Machine Approaches. In *Proceedings of the VII International Conference Days of Applied Mathematics*, San Jose de Cucuta, Colombia, 22 September 2020.
17. MathWorks Institute. Deep Learning Toolbox™—Matlab. Available online: <https://www.mathworks.com/products/deep-learning.html> (accessed on 21 July 2020).
18. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv* **2016**, arXiv:1603.04467.
19. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2020**, *60*, 84–90. [[CrossRef](#)]
20. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
21. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
22. Marti, U.V.; Bunke, H. The IAM-database: An English sentence database for offline handwriting recognition. *Int. J. Doc. Anal. Recognit.* **2002**, *5*, 39–46. [[CrossRef](#)]
23. Blumenstein, M.; Verma, B. Analysis of segmentation performance on the CEDAR benchmark database. In *Proceedings of the Sixth International Conference on Document Analysis and Recognition*, Seattle, WA, USA, 13–13 September 2001; pp. 1142–1146.
24. Kleber, F.; Fiel, S.; Diem, M.; Sablatnig, R. Cvl-database: An off-line database for writer retrieval, writer identification and word spotting. In *Proceedings of the 2013 12th International Conference on Document Analysis and Recognition*, Washington, DC, USA, 25–28 August 2013; pp. 560–564.
25. Augustin, E.; Brodin, J.M.; Carre, M.; Geoffrois, E.; Grosicki, E.; Preteux, F. RIMES evaluation campaign for handwritten mail processing. In *Proceedings of the Workshop on Frontiers in Handwriting Recognition*, La Baule, France, 23–26 October 2006; pp. 1–6.

26. de Andrade, A. Best practices for convolutional neural networks applied to object recognition in images. *arXiv* **2019**, arXiv:1910.13029.
27. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [[CrossRef](#)]
28. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
29. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [[CrossRef](#)]
30. Gonzalez, R.C.; Woods, R.E.; Eddins, S.L. *Digital Image Processing Using MATLAB*; Pearson Education: Tamil Nadu, India, 2004.
31. Otsu, N. A Threshold Selection Method from Gray-Level Histograms. *IEEE Trans. Syst. Man Cybernet.* **1979**, *9*, 62–66. [[CrossRef](#)]

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



© 2020 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 (<http://creativecommons.org/licenses/by/4.0/>).