Offline Handwritten Signature Verification Using Deep Neural Networks

José A. P. Lopes 1, Bernardo Baptista 1, Nuno Lavado 1 and Mateus Mendes 1,2, *

1 Polytechnic of Coimbra, Instituto Superior de Engenharia de Coimbra, 3030-199 Coimbra, Portugal
2 Instituto de Sistemas e Robótica, University of Coimbra, 3030-194 Coimbra, Portugal
* Correspondence: mmendes@iseec.pt

Abstract: Prior to the implementation of digitisation processes, the handwritten signature in an attendance sheet was the preferred way to prove the presence of each student in a classroom. The method is still preferred, for example, for short courses or places where other methods are not implemented. However, human verification of handwritten signatures is a tedious process. The present work describes two methods for classifying signatures in an attendance sheet as valid or not. One method based on Optical Mark Recognition is general but determines only the presence or absence of a signature. The other method uses a multiclass convolutional neural network inspired by the AlexNet architecture and, after training with a few pieces of genuine training data, shows over 85% of precision and recall recognizing the author of the signatures. The use of data augmentation and a larger number of genuine signatures ensures higher accuracy in validating the signatures.

Keywords: handwritten signature recognition; OMR; signature classification; CNN

1. Introduction

A signature has the power to identify a person as responsible for a legal act, to register presence in a place at a certain time, to authorise a transaction or even to become memorabilia. What makes it valuable is its exclusivity and belonging—a subject has a singular signature and it belongs to them. The present project aims to verify the signature of a student to prove their presence in a class and verify its authenticity. Ideally, the process will require minimal or no human intervention in the process. The old manual count and verification of attendance sheets and signatures will be replaced or at least greatly facilitated.

Digitisation is one of the main trends in the progress of society and organisations. Even the most traditional and bureaucratic tasks tend to be digitised, with advantages in terms of productivity and reduction in wasted paper. Digitisation is defined as the material process of converting analogue streams of information into a digital format [1]. Digitisation brings about paradigm shifts and changes in operations within organisations, due to the adoption of digital technologies that allow information to be extracted from data, analysed, transformed and for greater value to be extracted from it [2]. The biggest concerns and obstacles of the digitisation process in an organisation are the cost of equipment, staff time required for implementation and learning, creation of metadata and digital storage media [3]. Those costs are a challenge that make full digitisation still unjustified for some tasks where the costs would be greater than alternatives which are totally or partially manual.

Attendance validation is a fundamental task in school management, especially when attendance is mandatory because of funding or other requirements. In a classroom with dozens or hundreds of students, the challenge of signature validation is even greater because it is not feasible to ask for an identification document proving that the students are who they claim to be. Additionally, in many cases such as short courses, seminars or similar
events, the cost of using a full digital process is often higher than the cost of a manual or semi-manual process.

The present case study is a higher education institution, where attendance in intermediate courses is verified by signing a sheet in class, as evidence of their presence. Afterwards, the signatures are officially and manually recorded in a software platform that manages all academic activities. In this context, teachers and staff must record the attendance for each class and each student based on the signatures on the sheets. While the method is deemed safe, as signature falsifications are a serious issue which is not expected to happen, there are still possibilities of fraud or errors by the students. That means there may still be miscounting, due to fraud or errors in the process. The present work aims to improve the quality and speed of the process by proposing a method to automate the verification, counting and validation of signatures of class attendants. Digitization improves the speed of the process, frees staff from time-consuming tasks and reduces the likelihood of human error in data entry. The academic institution intends to implement a program that quickly verifies students’ attendance through recognition of their signatures in the attendance sheets. This is achieved through a computer program that is intended to automatically read the attendance sheet of a lesson, with many signatures. Then, individual signatures are retrieved and classified using a deep neural network.

When approaching a real problem such as the above, a number of problems arise that make it difficult to design a good solution. One of the problems with attendance sheets is that the signature fields are very small. The typical signature field in the dataset used is 215 × 30 pixels. The small amount of information per signature makes it difficult to develop a model that is able to extract the most important features of each student. A literature review shows that intra-class variability is a major problem in signature recognition. The same subject can have a large variability between signature samples. This variability is particularly sensitive in “writer-dependent” models, where the model is trained for each of the subjects [4]. Analysis of some of the data from the dataset also shows that a small minority of students use more than one type of signature, so the classical template matching approach is not appropriate to address the problem. Given the preceding challenges, we propose a solution that uses machine learning techniques to make the most of the data available.

The dataset consists of attendance sheets created by the campus academic management platform. For each lesson, a sheet is printed and passed by the attending students, so that they sign in the proper rectangle. On the attendance sheet, there is a header with a unique barcode and information regarding the lesson that is taking place. Underneath, there is a table listing the students enrolled in class. Each line of the table has three fields: (i) the student ID; (ii) the student name; and (iii) a blank rectangular space that corresponds to the place where the student must sign.

The first stage of the present work consisted of developing a script in Python using optical mark recognition (OMR) tools, namely, the OpenCV library [5], which includes functions for image manipulation, extraction of elements, and their identification from digitized attendance sheets. The script performs the following actions: (1) corrects the orientation of the attendance sheet; (2) identifies the class by its unique barcode; (3) uses morphological operations to identify the tables in which the enrolled students are displayed and finds each student by the horizontal lines separating them; (4) identifies the student ID and checks if there is a signature in the field provided for this purpose; and (5) exports the list of all the students assumed to be present to a file.

No signature verification is actually performed. Written marks, for example, if aligned on the same table row for a given student ID, are assumed to actually represent a signature.

In a second stage, signature validation is performed. To overcome the challenges and problems of signature validation, it was necessary to develop a robust system that is not too rigid to generalize the unique complex geometry of a signature with high accuracy. After verifying the presence of a signature, however, a trained model is used to verify that the signature belongs to the corresponding student in the grid. If the model has low confidence
that the signature belongs to the student, it is considered a possible forgery and marked for manual verification. Both steps will be detailed in the following sections. Signature validation is performed using a modified version of the deep learning model AlexNet, implemented using the Tensorflow library.

The remainder of the paper is as follows. In Section 2, related work is reviewed. Section 3 describes the methods. Section 4 describes the experiments and results. Section 6 draws some conclusions and notes future research directions.

2. Related Work

Offline handwritten signature recognition has been the subject of research since the early 1990s. Since then, many different solutions to this problem have been developed. Figure 1 shows a timeline where important developments are marked.

Miguel A. Ferrer et al. [6] propose the use of a template matching algorithm based on the fact that the envelope of a signature contains features that can uniquely identify the signature. To compute the envelope of a signature, it is proposed to first apply the morphological dilation operation to ensure that the outline of the signature is unified and the variability of the signature is reduced. Then, a filling operation is used to simplify the process of extracting the outline. After those operations, the Euclidean distance can be used as a measurement function between the envelopes of the signatures.

Hafemann et al. [7] propose a feature extraction approach that uses Convolutional Neural Networks (CNN) to distinguish between genuine signatures and forgeries. The proposed CNN architecture includes five convolutional layers, three max-pooling layers, and four fully connected layers. Batch normalization is suggested as a crucial step in the implementation of the model. After the CNN extraction, the authors propose a transfer learning approach to use the extracted features to train a support vector machine (SVM) with the radial basis function (RBF) kernel.

Other authors have proposed the use of traditional artificial neural networks to classify signatures. Al-Shoshan et al. [8] propose the use of Fourier descriptors and the contour of the signature as input to a multilayer perceptron (MLP). Ali Karouni et al. [9] use simple geometric features of the signature such as area, centroid, eccentricity, kurtosis, and skewness to train an artificial neural network (ANN).

Several approaches have been proposed in the literature using concepts from graph theory. Tomislav Fotak et al. [10] propose the use of graph connectivity and the existence of Eulerian and Hamiltonian paths as features to classify a signature, and also propose the computation of the minimum spanning tree of the signature to build the graph of the signature, using the concept of avoiding sets as a condition to measure the distance and similarity between signatures. That is a combinatorial necessary and sufficient condition for cluster consensus [11]. The method achieves 94.25% accuracy.
Abhay Bansal et al. [12] also propose the use of graph theory concepts to compare signatures by first extracting the critical points on the signature contour and using a graph matching algorithm to compute the similarities between signatures.

The use of Hidden Markov Models (HMM) has been widely popular in the literature and is proving to be a good statistical model for solving the signature recognition problem. An example of this is the work of J. Coetzer et al. [13], who propose the use of discrete Radon transforms and a Hidden Markov Model with ring topology trained with the Viterbi algorithm to detect forgeries. Other authors [14,15] have proposed the use of Hidden Markov Models, making them a well-known method in the field of signature recognition.

In one particular study, an MLP network was used to classify the features, in the last layer of a CNN model used to extract the signature features. Similar to the present study, the final result is binary, but it allows to check for forgery. The results of the model showed high accuracy [16].

AlexNet is a CNN originally developed by Alex Krizhevsky et al. [17] and is one of the most influential works on image recognition. The original paper, published in 2012, demonstrated low error in the CIFAR-10 dataset, which consists of over 1000 classes with many similarities. Therefore, it is a suitable candidate for solving the signature recognition problem, namely, to classify marks as a signature of a particular student.

Convolutional networks are often used in offline signature recognition and with good results, as shown in a systematic literature review paper [18]. In particular, feature extraction using AlexNet [19] shows an accuracy as high as 100%, but for a limited dataset of 600 signatures. A genetic algorithm searching for a rational set of features, such as checking the curvature of signature features, achieves better results in terms of equal error rate (EER) [20]. Kao and Wen [21] demonstrate recognition with single reference sample to check how forgery local feature extraction of signatures reduces the number of samples to train and it is even possible to obtain good results with only one sample.

3. Methodology

3.1. Workflow of the Method

The methods followed were based on the following steps:
1. Prepare datasets of signatures extracted from attendance sheets;
2. Develop a binary classifier (feed forward neural network) to distinguish signatures from non-signatures;
3. Develop a deep classifier (CNN) to classify signatures as legitimate or not.

Figure 2 shows a block diagram highlighting the most important steps of the present project, from collection of data to fine-tuning of the images and neural network architectures.

![Figure 2. Block diagram showing the steps of the method followed.](image_url)
3.2. Signature Image Extraction

The images of the signatures used are from various attendance sheets. These sheets came from different departments on campus and were scanned by different people using different scanning devices and different resolutions. When the sheets were read, those that the script could not read were excluded. The causes identified for rejected sheets are: (1) errors in sheet orientation, (2) poor scan quality/resolution, and (3) barcode reading errors. Those sheets were left out of the dataset and ideally those problems will be avoided in the future.

One of the first challenges is the validation of a mark as a signature, distinguishing it from other elements such as a stroke or a stain, which are not signatures. This problem can often be minimized through the delimitation of the region of interest. In the present case study, the regions of interest in the attendance sheet were delimited. A set containing sub-images of the regions where students were supposed to sign in was retrieved from a total of 1635 attendance sheets. From those sheets, after a manual validation of the image category and error elimination, 20,516 categorized images were available, of which a total of 15,847 correspond to the “presence” category and 4669 to the “absence” category.

The signature images were extracted and pre-processed using various OpenCV [5] functions. The attendance sheets contain a table of students enrolled in each lesson. After each student’s ID and name, there is a blank rectangle for the student to sign. This region is defined as the Region Of Interest (ROI) for each row in the table. The ROI images were extracted, based on the rectangles’ borders. They were also transformed into a single-channel grayscale image and normalized to the size of $215 \times 30$ pixels. Resizing preserved the aspect ratio of the images and all their distinctive features as much as possible.

To reduce noise, a Gaussian blur and threshold were applied. The parameter \texttt{THRESH_TRUNC} is used to transform the image so that the unwritten area is cleared from noisy pixels. Applying the threshold function with this parameter allows to change the colouring up to a certain value and keep the properties of the pixels. In this case, the area that is supposed to be unwritten is clearer when original pixels are between 240 and 254, thus turning into white (255) and the degree of darkness of the area where the signature is located is kept [22]. The transformation is applied to all images extracted from the attendance sheet to preserve uniformity in the dataset.

3.3. Signature Dataset for Classification

Each signature image is a $215 \times 30$ matrix. For easier classification through machine learning models, the matrix is represented by a set of 245 values, which correspond to the sum of the values in each pixel for each image row and column. Hence, an image $I$ with $m$ rows and $n$ columns is represented by vector $\hat{I}$, where $R_i$ is the sum of pixels of row $i$, for $i \in [0, m - 1]$ and $C_j$ is the sum of the pixels for column $j$, for $j \in [0, n - 1]$:

$$\hat{I} = \{R_1, R_2, ... , R_m, C_1, C_2, ..., C_n\}$$

(1)

In the present case, each row of the dataset has 246 columns: 245 values for each signature image, represented as $\hat{I}$, plus one binary value $S$, which states whether it is a valid signature or not.

The option to use a representation of a signature’s image as the value of the sum of the rows and columns is that it is a simpler way to represent the image, while still retaining properties important for classification, reducing the processing time and condensing the dataset. This method is popular in the literature, especially for image classification using machine learning models such as neural networks [23].

Since all the images that corresponded to non-signatures were used (4669), the same number of images were randomly selected from the pool of signature corresponding images, so that the dataset is balanced, comprising 4669 non-signatures and 4669 signatures. To comply with the 70% training and 30% testing rule, both subsets were divided so that 6538 images fell into the training pool and 2800 images fell into the testing pool.
The dataset created this way was used for the binary classification of images as signature/non-signature, as detailed in Section 4.2.

### 3.4. Proposed MLP Architecture for Mark Recognition

An MLP is a network that maps sets of input data onto a set of outputs in a feedforward manner [24]. There is a layer of input nodes, a layer of output nodes, and one or more intermediate layers called “hidden layers” because they are not directly observable [25]. The nodes in each layer are fully connected to both the preceding layer and the succeeding layer.

The MLP network features one input layer with 215 nodes, one hidden layer with 100 nodes, and one output layer with 2 output variable. Figure 3 presents a schematic diagram for this MLP neural network.

![Figure 3. Proposed MLP schematic model.](image)

### 3.5. Data Augmentation

The available dataset contains thousands of images, so it is expected to be enough for artificial neural networks, even deep convolutional neural networks. Nonetheless, it is often possible to improve the results benefiting from data augmentation to increase the generalization capabilities of the models. That is especially important when training a model to tell a real signature from a forgery. In that case, the number of examples of valid signatures may be small for each person. Therefore, the goal of the model is to achieve good generalization with a small data set of valid signature examples. Therefore, it is important to use data augmentation to capture possible variations in new data.

The geometry of handwritten signatures is very unique, and therefore, it is bound to choose operations that do not destroy the original information of the signature. Horizontal and vertical flipping are poor choices, since signatures are not expected to be symmetrical relative to any axis. The most obvious valid operations are rotations and translations. They represent real variations that are expected to happen, since it is quite common for a signature to be written at a different angle or location within the signature field. Figure 4 shows an example of a signature which has been modified through rotation and translation.
Figure 4. Example of signature modified using data augmentation operations. The yellow area is later filled with the same colour as the background of the signature so as not to introduce noise into the dataset. The image is in its true resolution.

To generate the training datasets with data augmentation, constraints on the augmentation operations were introduced. Horizontal and vertical displacements were limited to 15 and 3 pixels in each direction, respectively. The rotation operation was limited to the $[-0.1, 0.1]$ radian interval. Increasing the ranges for those operations resulted in very frequent partial losses of the signatures, which is neither ideal for training, nor realistic.

3.6. Hyperparameter Tuning

The CNN model is meant to recognize different signatures, which are grouped according to the different classes of students. Therefore, it needs to be trained every time we need to evaluate a new class. Additionally, the number of students may vary from class to class, as well as the number of training examples. Hence, some of the hyperparameters are automatically adjusted to maintain the model consistency. The number of training samples varies linearly with the number of students in the group.

After analysing the learning behaviour of the model during training, the number of epochs was set to 25. Figure 5 shows that after 25 epochs and a certain batch size, the validation loss started to stagnate, indicating that the convolutional neural network was not able to improve the features extracted from each signature.

To determine the batch size that best fits a given dataset, the number of steps per epoch was set to a constant value, which was 450. Then, Equation (2) was used to determine which batch size $B$ yields this value, for $S$ training samples.
\[ B = \max(\text{int}(S^{450}), 2) \]  

The max function is used to prevent having a batch size equal to 1 or 0 in extreme cases, where the number of training examples is very small, as this could potentially lead to errors in the code or poor results from the model.

3.7. Proposed CNN Architecture

A multiclass convolutional neural network, inspired by the AlexNet architecture, is proposed. This convolutional neural network consists of 5 convolutional layers, 3 max-pooling layers, and 3 fully connected layers, paired with dropout layers, yielding a total of 31.2 million parameters. Table 1 shows a summary of the architecture of the neural network.

<table>
<thead>
<tr>
<th>Layer Type</th>
<th>Size</th>
<th>Other Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>215 x 90 x 1</td>
<td></td>
</tr>
<tr>
<td>Convolution</td>
<td>96 x 11 x 11</td>
<td>strides = 4</td>
</tr>
<tr>
<td>MaxPooling2D</td>
<td>96 x 3 x 3</td>
<td>strides = 2</td>
</tr>
<tr>
<td>Convolution</td>
<td>256 x 5 x 5</td>
<td>strides = 1</td>
</tr>
<tr>
<td>MaxPooling2D</td>
<td>256 x 3 x 3</td>
<td>strides = 2</td>
</tr>
<tr>
<td>Convolution</td>
<td>384 x 3 x 3</td>
<td>strides = 1</td>
</tr>
<tr>
<td>Convolution</td>
<td>384 x 3 x 3</td>
<td>strides = 1</td>
</tr>
<tr>
<td>MaxPooling2D</td>
<td>256 x 3 x 3</td>
<td>strides = 2</td>
</tr>
<tr>
<td>Fully connected</td>
<td>4096</td>
<td></td>
</tr>
<tr>
<td>Dropout</td>
<td>4096</td>
<td>dropout rate = 0.5</td>
</tr>
<tr>
<td>Fully connected</td>
<td>4096</td>
<td></td>
</tr>
<tr>
<td>Dropout</td>
<td>4096</td>
<td>dropout rate = 0.5</td>
</tr>
<tr>
<td>Fully connected + Softmax</td>
<td>N</td>
<td></td>
</tr>
</tbody>
</table>

Several changes were made to the original AlexNet architecture, including the use of batch normalization between convolutional layers instead of the original local response normalization. This change allowed the convolutional neural network to better converge to a solution. It is also important to note that the original AlexNet has an input layer consisting of 3 colour channels, which is not the case in our proposed architecture, as shown in Table 1.

Before training or inferring with the convolutional neural network, the input images must be resized to a reasonable size, such as 215 x 90, due to the heavy downsampling caused by the max-pooling layers. This process does not affect the main features of each signature, as long as it is applied equally to each example. The output layer consists of N units that indicate the number of students. The output of the last fully connected layer is fed to a N-way softmax, which produces a distribution over the N class labels. The padding and stride values used in each layer are used to effectively adjust the dimensionality of the input data.

The stacked convolutional layers allow the neural network to decompose the signature features hierarchically. This makes the feature extraction process much more efficient and accurate, as the lower-level features are captured in the final convolutional layers.

3.8. Optimization Algorithm

During the model optimization phase, different optimization algorithms were used to select the most appropriate one for the problem being solved. In 2015, Diederik P. Kingma et al. [26] proposed ADAM, an algorithm for first-order gradient-based optimization of stochastic objective functions based on adaptive estimates of lower-order moments. After comparing ADAM with other optimization algorithms, we concluded that the best option is the stochastic gradient descent algorithm (SGD), which provides better generalization than
ADAM and other algorithms. In a study published by Pan Zhou et al. [27], a theoretical analysis is performed to understand the convergence behaviour of both algorithms and why the SGD provides better generalization in some cases.

3.9. Exponential Learning Rate Schedule

To improve the optimization of the gradient descent algorithm, we decided to test the exponentially decaying learning rate technique. By default, the stochastic gradient descent algorithm is assumed to use a constant value for the learning rate, which can be manually adjusted to achieve better results. This could lead to problems when trying to converge to a local minima, since the value of the learning rate does not change between training steps. To solve this problem, the value of the learning rate is iteratively decreased, in order to facilitate the process of convergence in later training epochs. The learning rate $L R_i$ for step $i$ is calculated as in Equation (3), where $\lambda$ is the decay rate and $L R_0$ is the initial learning rate [28].

$$L R_i = L R_0 \times \lambda^{(\frac{\text{steps}}{2500})}$$

This expression yields a staircase function that decreases the value of the learning rate in an exponential fashion. During training, we used an initial learning rate of 0.001 with a decay rate of 0.9 and 2500 steps.

3.10. Dropouts and Batch Normalization

After some initial testing, the CNN still had problems generalizing the signature features, as the validation loss did not decrease during the training phase. To solve this problem, it was decided to test different combinations of batch normalization and dropout to study the behaviour of the CNN. In 2014, Srivastava et al. [29] published a technique called dropout, a machine learning technique that aims to reduce overfitting by dropping units in the neural network at a certain predetermined rate.

Recent research suggests that using dropout prior to batch normalization (BN) leads to overall poorer performance [30]. After testing various possible placements of both techniques, the best performance can be achieved when batch normalization is applied after each convolutional layer and the dropout technique is applied later in the fully connected layers at a rate of 50%. This value was chosen according to the the original AlexNet study [17].

3.11. Weight Decay

When tuning the neural network with dropout and batch normalization, the generalization capabilities of the current model increased dramatically compared to previous tests. Although the overall results improved with these techniques, there was still a noticeable problem with training the neural network, as the loss values occasionally increased exponentially, causing the model to lose all previous progress. This type of behaviour is usually caused by uncontrolled changes in the neural network parameters, which can cause the exploding gradient problem.

To solve this problem, different weight decay norms were tested to penalize this kind of behaviour. After some tests, we concluded that the L2 norm gave the best results, because it completely prevents major changes to the neural network parameters [31,32].

$$C(w, b) = \frac{1}{n} \sum_{i=1}^{n} L(\hat{y}_i, y_i) + \frac{\lambda}{2n} \sum_{i=1}^{n} w_i^2$$

In Equation (4), $L$ represents the sparse categorical cross-entropy loss function which takes two arguments, $\hat{y}_i$ and $y_i$, which represent the network prediction to some input and the true input label value, respectively. The value of $\lambda$ represents the penalty factor, chosen to be 0.005. By applying this norm to the sum of the $n$ examples, the neural network is penalized for deviating weight values $w$, as the cost function increases with the square of
each weight. The values of the biases $b$ are not changed with this operation. By setting $\lambda$ to a smaller value, we give the neural network enough freedom to explore the parameter space, but constrain it when the deviations of the weights become too obvious and potentially harm the model performance.

3.12. Upsampling

One way that was also deemed important to further improve the results was to resize the input images, in order to facilitate feature extraction. The original input images were small, and max-pooling layers further reduce the already small image. Given a factor value, the columns and rows of pixels that make up the image should increase, without changing the appearance of the image. To fill the new space, interpolation algorithms are used to produce intermediate values that fill the new pixels of the image.

4. Experiments and Results

The results of both experiments, the binary classification and the signature recognition, are described in the following subsections.

4.1. Evaluation Metrics

Two metrics were used to evaluate the results of the model, namely, precision ($P$) and recall ($R$). Both metrics can be calculated using the confusion matrix (true value $\times$ predicted value), created by using the model to classify the test set, since the expressions use false-positive ($FP$), false-negative ($FN$), true-positive ($TP$), and true-negative ($TN$).

$$P = \frac{TP}{TP + FP} \quad R = \frac{TP}{TP + FN}$$  \hspace{1cm} (5)

Since the proposed architecture is a multi-class model, the computations of these expressions are slightly different. Let $M$ be the confusion matrix generated by the model. Then, we can compute the precision and recall of class $i$ in the following way, for $n$ classes.

$$P_i = \frac{M_{ii}}{\sum_{j=1}^{n} M_{ij}} \quad R_i = \frac{M_{ii}}{\sum_{j=1}^{n} M_{ji}}$$  \hspace{1cm} (6)

We can then calculate the average of the metrics of all classes to obtain an estimate of the model performance.

4.2. Binary Classification Model Test

The results of the feed-forward neural network trained as a binary classifier of signature/non-signature are presented in Figure 6. The activation function responsible for transforming the summed weighted input from the node into the activation of the node or output for that input is the rectified linear activation function (ReLU) [33]. Adam optimizer algorithm [26] was used for adjusting network weights. In the confusion matrix shown, the success rate for each of the labels and the errors made by the classifier are displayed, both false positives and false negatives. The original script for this Seaborn confusion matrix scheme is available on GitHub [34].
On the test data, the MLP classifier performed well. The accuracy is 98.4% and the F1 score is 98.3%. The test proves the suitability of the model for binary evaluation of the presence or absence of a signature proving the presence of a particular student. However, this model was trained and tested with images that clearly show signatures or their absence. Uncertain cases, such as erasures, scribbles or students not signing in the exact place that prevent attribution to the specific student ID, affect the reliability of the classification method. This method also does not make it possible to verify whether the signature actually comes from a specific student—additional validation is performed by the CNN.

4.3. Cnn Results

In order to determine the performance of the CNN model to distinguish valid and not valid signatures, a smaller set was selected to measure the performance of the model without any bias. A random sample of 50 students was selected to be part of the dataset. During training, a varying number of signatures with data augmentation were used to observe the effects of data augmentation on the generalization of the model and overall performance. In addition to using data augmentation, two different sets with different numbers of genuine signatures were selected to understand the effectiveness of genuine data compared to augmented data during training.

After training each model for each combination of genuine and augmented signatures, a test set was used to accurately evaluate the performance of the model. For this purpose, 20 genuine signatures were selected from the same group of 50 students in such a way that no genuine signature from the training dataset was present in the test set. After conducting the tests, data augmentation proved to be a critical step to achieve better model results, as shown in Table 2. Using 5 genuine signatures and 0 augmented signatures, the results show that the model achieves 52.8% accuracy and a recall of 48.7%. However, when we increase the number of augmented signatures, these two metrics increase significantly to 72.3% and 71.5%, respectively. This confirms the original hypothesis, as there is a strong correlation between the number of augmented signatures and the performance of the model. The same is true for the genuine signatures, as the results show that the use of 10 genuine signatures further improves the results. In the case where 10 genuine signatures and 160 augmented signatures were used, the model achieved a precision of 82.2% and a recall of 82.0%.

Figure 6. Confusion matrix and metrics for test data with MLP Classifier.
Table 2. Test set results using different number of genuine and augmented signatures in the training set.

<table>
<thead>
<tr>
<th>Genuine Signatures</th>
<th>Augmented Signatures</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0</td>
<td>52.8%</td>
<td>48.7%</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>61.5%</td>
<td>59.7%</td>
</tr>
<tr>
<td>5</td>
<td>40</td>
<td>66.2%</td>
<td>64.7%</td>
</tr>
<tr>
<td>5</td>
<td>80</td>
<td>69.3%</td>
<td>68.2%</td>
</tr>
<tr>
<td>5</td>
<td>160</td>
<td>72.3%</td>
<td>71.5%</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>67.8%</td>
<td>66.4%</td>
</tr>
<tr>
<td>10</td>
<td>20</td>
<td>74.5%</td>
<td>73.8%</td>
</tr>
<tr>
<td>10</td>
<td>40</td>
<td>79.5%</td>
<td>80.1%</td>
</tr>
<tr>
<td>10</td>
<td>80</td>
<td>81.0%</td>
<td>81.5%</td>
</tr>
<tr>
<td>10</td>
<td>160</td>
<td>82.2%</td>
<td>82.0%</td>
</tr>
</tbody>
</table>

Training the model with different datasets shows that increasing the size of the training data has a great impact on the generalization of the model, as shown the confusion matrices in Figure 7. Additionally, they show that a small increase of genuine signatures leads to large improvements in the results.

Figure 7. Confusion matrices generated by evaluating the test set with different training sets composed of different amounts of genuine and augmented signatures per student.

Figure 7 also excludes the possible existence of outliers that could contribute to an increase in error. The uniform noise outside the main diagonal of the matrix proves this hypothesis.
4.4. Upsampling Application

Since we want to increase the size of the images, linear interpolation was used as the resizing algorithm. To measure the performance of the model, different resizing factors were used.

Table 3 shows the results obtained using upsampled images, with the model trained with 10 genuine signatures and 160 augmented signatures per user. It can be seen that this operation provides a slight improvement in the model results. The drawback of this technique is that the computational load required to train the model scales quadratically with the dimensions of the image, adding a fair number of parameters to the convolutional neural network. By applying a resizing factor of 3, the total number of parameters with the proposed architecture is about 100 million.

Table 3. Model results with upsampled images.

<table>
<thead>
<tr>
<th>Size</th>
<th>Resize Factor</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>430 × 180</td>
<td>2</td>
<td>83.2%</td>
<td>84.8%</td>
</tr>
<tr>
<td>537 × 225</td>
<td>2.5</td>
<td>85.0%</td>
<td>85.2%</td>
</tr>
</tbody>
</table>

5. Discussion

The present work uses true signatures from real attendance sheets. The results were obtained with images which do not have high resolution and often have noise.

In light of more recent studies with signature recognition and forgery, it is possible to compare the results obtained in the present work with the state of the art. A comparison is made in Table 4.

Table 4. Comparison of similar work.

<table>
<thead>
<tr>
<th>Author, Year and Reference</th>
<th>Accuracy Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ghanim &amp; Nabil, 2018 [35]</td>
<td>Signature Recognition</td>
</tr>
<tr>
<td></td>
<td>Bagging Trees, Random Forest &amp; SVM</td>
</tr>
<tr>
<td>Jagtap et al., 2020 [36]</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Siamese Neural Network (SNN)</td>
</tr>
<tr>
<td>Mshir &amp; Kaya, 2020 [37]</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>SNN</td>
</tr>
<tr>
<td>Poddar et al., 2020 [38]</td>
<td>94%</td>
</tr>
<tr>
<td></td>
<td>CNN, SURF algorithm &amp; Harris corner detection algorithm</td>
</tr>
<tr>
<td>Proposed</td>
<td>85.0</td>
</tr>
</tbody>
</table>

The proposed method has some advantages when compared to other state-of-the-art methods. One of them is the possibility of high-precision classification with a small number of genuine samples, which is crucial in some scenarios where little is known about the real-world data. This feature is enabled by the feature extraction capability of convolutional neural networks. The main disadvantage of this method is the initial time required to train the new models. A new model is required to train for every set of new users and the model has about 100 million parameters, therefore, the training process may take some time. Nonetheless, this shall still require less time than manual verification of the signatures, and require less energy and money than when digital equipment are installed in place for occasional verification.

It is important to emphasize that it is difficult to draw a comparison between the proposed method and some of the state-of-the-art results because different datasets were used. As mentioned earlier, the datasets used to evaluate the model proposed consist only of low-resolution images. Additionally, the number of genuine samples per user is reduced. The problem becomes easier to resolve using very high resolution images, which may be difficult to acquire in the context of attendance sheet verification.
6. Conclusions

Handwritten signatures in attendance sheets are an important attendance verification method that is still used nowadays.

The present work investigates the use of two different automatic verification methods: (1) the MLP classifier validates the absence or presence of marks for handwritten content; and (2) the CNN model allows individual signature recognition and validates similarity to detect possible forgeries.

The optical mark classification model has high accuracy, and a single model is adequate for all signatures in a rectangle of the given size. However, it does not provide information about the author of the signature.

The proposed convolutional neural network shows positive results in a scenario where the number of genuine signatures is limited and the region of interest contains very little information. The properties of this deep neural network allow the model to be robust to a reasonable amount of variability. The results show that data augmentation is an essential step to achieve better results. The model was improved using machine learning techniques such as dropout layers and batch normalization. The best precision obtained was 85%, with just 10 genuine signatures, which is very good for the problem being addressed and the number of samples given to the model.

A limitation of this method is that it requires training examples of the signatures that need to be recognized.

The inclusion of local feature extraction may increase the effectiveness of the model described. Additional experiments may also be performed with other neural network architectures or datasets. As far as putting this project into practice in the real world, future work includes development of a Graphical User Interface so that campus staff can use it without difficulty.

Author Contributions: Data curation, B.B.; Investigation, J.A.P.L., B.B. and N.L.; Supervision, N.L. and M.M.; Writing—original draft, J.A.P.L. and B.B.; Writing—review & editing, N.L. and M.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work received financial support from the Polytechnic Institute of Coimbra within the scope of Regulamento de Apoio à Publicação Científica dos Professores e Investigadores do IPC (Despacho n.° 12598/2020).

Data Availability Statement: Not applicable.

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

References