inventions

MDPI

Article

A Fingerprint Matching Algorithm Using the Combination of Edge Features and Convolution Neural Networks

Andreea-Monica Dincă Lăzărescu [1,2], Simona Moldovanu [1,3] and Luminita Moraru [1,4,*]

[1] The Modelling & Simulation Laboratory, Dunarea de Jos University of Galati, 47 Domneasca Str., 800008 Galati, Romania; andreea.dinca@ugal.ro (A.-M.D.L.); simona.moldovanu@ugal.ro (S.M.)
[2] Mihail Kogălniceanu High School, 161B Brăilei St., 800320 Galați, Romania
[3] Department of Computer Science and Information Technology, Faculty of Automation, Computers, Electrical Engineering and Electronics, Dunarea de Jos University of Galati, 47 Domneasca Str., 800008 Galati, Romania
[4] Department of Chemistry, Physics & Environment, Faculty of Sciences and Environment, Dunarea de Jos University of Galati, 47 Domneasca Str., 800008 Galati, Romania
* Correspondence: luminita.moraru@ugal.ro

Abstract: This study presents an algorithm for fingerprint classification using a CNN (convolutional neural network) model and making use of full images belonging to four digital databases. The main challenge that we face in fingerprint classification is dealing with the low quality of fingerprints, which can impede the identification process. To overcome these restrictions, the proposed model consists of the following steps: a preprocessing stage which deals with edge enhancement operations, data resizing, data augmentation, and finally a post-processing stage devoted to classification tasks. Primarily, the fingerprint images are enhanced using Prewitt and Laplacian of Gaussian filters. This investigation used the fingerprint verification competition with four databases (FVC2004, DB1, DB2, DB3, and DB4) which contain 240 real fingerprint images and 80 synthetic fingerprint images. The real images were collected using various sensors. The innovation of the model is in the manner in which the number of epochs is selected, which improves the performance of the classification. The number of epochs is defined as a hyper-parameter which can influence the performance of the deep learning model. The number of epochs was set to 10, 20, 30, and 50 in order to keep the training time at an acceptable value of 1.8 s/epoch, on average. Our results indicate the overfitting of the model starting with the seventh epoch. The accuracy varies from 67.6% to 98.7% for the validation set, and between 70.2% and 75.6% for the test set. The proposed method achieved a very good performance compared to the traditional hand-crafted features despite the fact that it used raw data and it does not perform any handcrafted feature extraction operations.

Keywords: convolutional neural networks; fingerprint images; data augmentation; fingerprint classification

check for updates

Citation: Dincă Lăzărescu, A.-M.; Moldovanu, S.; Moraru, L. A Fingerprint Matching Algorithm Using the Combination of Edge Features and Convolution Neural Networks. *Inventions* **2022**, *7*, 39. https://doi.org/10.3390/inventions7020039

Academic Editor: Anastasios Doulamis

Received: 10 May 2022
Accepted: 26 May 2022
Published: 27 May 2022

1. Introduction

Various approaches to automatically authenticate fingerprints for personal identification and verification have found important applications in ensuring public security and criminal investigations. A fingerprint represents a graphical pattern on the surface of a human finger expressed by ridges and valleys. Ridges are the upper skin surface parts of the finger that touch a surface, and valleys are the lower parts. In a fingerprint image, ridge lines are the dark areas, and valleys are the bright areas which represent the inter-ridge spaces. Fingerprints are unique and are the most reliable human feature which can be utilized for personal identification [1].

Automatic fingerprint identification uses fingerprint features such as ridge flow, ridge period, ridge ending, and the delta or core points for fingerprint enrollment and verification steps [2]. Matching performance is strongly affected by fingertip surface conditions such as fingerprint deformations or distortions, fingerprint collection conditions, variations in the pressure between the finger and the acquisition sensor, scars, age, race, sex, etc. Additionally, the minutiae extraction can be affected by noise, rotation, and the scale of the images or fingerprint alignment information [1–3]. There is a sinusoidal-shaped wave of ridges with some slow changes in their orientation. This characteristic defines a fingerprint pattern. However, fingerprint images are prone to structural imperfections. In order to create an accurate identification system, an effective enhancement algorithm is necessary. This algorithm can be coupled with a performance classification method [4,5]. A major limitation of fingerprint recognition algorithms is that only small-area fingerprint images are usually available to the algorithm for differential matching. This calls for a model which can solve the restoration of the whole fingerprint image to make the process of fingerprint recognition and matching more effective [6–8]. The enhancement step is based on the obvious directional behavior manifested in a fingerprint image. Some effective enhancement techniques are based on the Prewitt and Laplacian of Gaussian filters [9,10]. Additionally, a robust feature extractor and classifier must be able to deal with augmentation operations such as translation, rotation, or skin distortion.

The process of feature extraction and matching demands some preprocessing operations such as ridge enhancement (for a fingerprint structure clarity), followed by feature extraction using artificial neural networks. Recently, deep convolutional networks have been heavily used in image recognition. Most of these are devoted to a single-frame recognition with an improved classification performance [11–15]. The main advantage of CNN-based classifiers is that they are fully independent of any human actions devoted to feature extraction and classification. For large databases, the computational cost of searching for a fingerprint image is huge, but CNNs drastically reduce this burden.

In the present study, starting from the fact that the existing fingerprint recognition algorithms rely too heavily on the details of the fingerprint, a software solution was proposed to evaluate the quality of fingerprint identification by using a convolutional neural network (CNN) architecture and by calling full images belonging to four digital databases. We did not perform any handcrafted feature extraction operations. The main challenge that we face is the low quality of fingerprints, which can impede or make the identification process difficult.

The main contributions of this work are as follows:

- Despite intense development efforts, there is still one open research problem devoted to the restoration of whole fingerprint images to make the process of fingerprint recognition and matching more effective. We discuss the investigation of whole fingerprint images.
- We aim to validate the edge enhancement operations, data augmentation, and the network structure regarding the potential of a CNN architecture to accurately identify the fingerprints for a further classification task.
- The Prewitt and Laplacian of Gaussian filters are used to enhance the edges that separate the ridges and valleys in the fingerprint images. Moreover, we do not use any skeletonization operations to convert gray-scale fingerprint images to black-and-white images.
- To decrease the training time, we reduce the dimensionality of the fingerprint images from [256 × 256] to [80 × 80] pixels.
- To improve the performance of the proposed model, we use the rotation as a data augmentation technique.
- As CNNs can learn discriminative features from whole fingerprint images and they do not require explicit feature extraction to do so, the deep learning approach is an attractive option in fingerprint identification. Thus, the performance of the proposed CNN model is evaluated based on the accuracies for the training and validation tests with attention paid to the number of epochs, which is considered the hyper-parameter of the CNN that could influence the performance of the deep learning model.

The rest of this paper is organized as follows: Section 2 reviews the related literature and emphasizes the most important sources of our motivation. Also, in Section 2 the adopted methodology, the design of our CNN method, and the used databases are presented. Section 3 shows the results obtained from several experiments, discusses and evaluates the accuracy values provided by the individual systems. Finally, Section 4 provides a summary of our research and sets out our future work intentions.

2. Materials and Methods

2.1. Related Work

Over the last thirty years, there have been important developments in deep learning methods. These developments had a significant impact on a wide range of applications dealing with computer vision and pattern recognition. The research field of automatic fingerprint recognition is among the most interesting topics, due to the requirement to increase the recognition accuracy rate. Additionally, deep learning methods avoid the focus on methods devoted to minutiae extraction as handcrafted features, shifting the interest to the analysis of the whole image. The latest investigations devoted to the field of fingerprint image organization are reviewed in this section.

The most sensitive step in the fingerprint recognition scheme is image improvement. Wang et al. [16] proposed an algorithm for fingerprint image quality enhancement, i.e., to improve the clarity and continuity of ridges, based on the wavelet transform and a mechanism of compensation coefficients for each sub-band based on a Gaussian template. Yang et al. [17] presented an enhancement technique which approached both spatial and frequency domains. A spatial ridge-compensation filter was employed to enhance the fingerprint image in the spatial domain. Then, a frequency bandpass filter performed sharp attenuation of both the radial and angular-frequency domains. Shrein et al. [13] used a convolutional neural network that performed the classification tasks for fingerprints in the IAFIS (integrated automated fingerprint identification system) database with 95.9% accuracy. He has shown that precise image preprocessing, aiming to reduce the dimensionality of the feature vector, greatly decreased the training times, even in networks of moderate depth. Mohamed [14] thoroughly investigated all the factors which may affect fingerprint classification using CNNs. His proposed system consists of a preprocessing stage dealing with increasing the fingerprint quality, and a post-processing step devoted to training and classification. A resized image (its dimensions reduced from 512 × 512 pixels to 200 × 200 pixels) was created in order to reduce the training time. A classification accuracy of 99.2% with a zero rejection rate was reported. Militello et al. [15] used a pre-trained convolutional neural network and two fingerprint databases with heterogeneous characteristics (PolyU and NIST) for classification purposes. The used features were the arch, left loop, right loop, whorl, and three nets: AlexNet, GoogLeNet, and ResNet. The comparative analysis allowed the system to determine the type of classification that should be used for the best performance in terms of precision and model efficiency. Borra et al. [18] reported a method based on a denoising procedure (the wave atom transform technique), image augmentation (based on morphological operations), and an adaptive genetic neural network in order to evaluate the performance of the approach. The networks used the feature values that were extracted from each image. The experiments were performed on the FVC2000 databases. The authors reported better performance values compared to some neural networks and machine learning approaches. Listyalina et al. [19] sought to classify raw fingerprint images. They proposed a deep learning method (i.e., transfer learning GoogLeNet) which transferred the classification steps, such as pre-processing, feature extraction, and classification rather than training a deep CNN architecture. They used fingerprint images from the NIST-4 database and reported performance accuracy measures as follows: 94.7% and 96.2% for the five-class and four-class classification problems, respectively. Tertychnyi et al. [20] proposed an efficient deep neural network algorithm to recognize low-quality fingerprint images. These images are affected by physical damage, dryness, wetness, and/or blurriness. A VGG16 convolutional network model was employed based on transfer learning for training. In addition, both image dimension reduction and data augmentation were performed to improve the computing cost. They reported an average accuracy of 89.4%; this is almost the same accuracy provided by regular CNNs. Finally, Pandya et al. [21] proposed a model which encompasses a pre-processing stage (histogram equalization, enhancement based on a Gabor filter, and ridge thinning) and a classification stage using a CNN architecture. The proposed algorithm achieved a 98.21% classification accuracy with a 0.9 loss for 560 samples (56 users providing 10 images each). Overfitting was avoided by Nur-A-Alam et al. [22] using a combination of the Gabor filtering technique coupled with deep learning techniques and principal component analysis (PCA). The meaningful features that can support an automatic authentication process for the fingerprint for personal identification and verification were extracted using the fusion of CNNs and Gabor filters; PCA reduces the dimensionality of statistical features. The proposed approach reached an accuracy of 99.87%. An efficient unimodal and multimodal biometric system based on CNNs and feature selection for fast palmprint recognition was recently proposed by Trabelsi et al. [23]. Simplified Gabor–PCA convolutional networks, an enhanced feature selection method, and a "reduction of the dimensions" approach were used to achieve a high recognition rate, i.e., 0% equal error rate (meaning the best trade-off between false rejections and false acceptances) and 100% rank-one recognition (meaning the percentage of samples recognized by the system). Oleiwi et al. [24] introduced a fingerprint classification method based on gender techniques, which integrates the Wiener filter and multi-level histogram techniques with three CNNs. They used CNNs to extract the fingerprint features followed by Softmax as a classifier.

2.2. Proposed Methodology

2.2.1. Mathematical Approaches

To improve the quality of fingerprint images, the first- and second-order derivative filters were used. An image is defined by an image function $A(x, y)$ that gives the intensity of the gray levels at pixel position (x, y). The gradient vector of the image function is defined as in [25]:

$$\nabla A(x,y) = \left[G_x G_y\right] = \left[\frac{\partial A(x,y)}{\partial x}\ \frac{\partial A(x,y)}{\partial y}\right] \tag{1}$$

- Prewitt Operator

The Prewitt filter detects the vertical and horizontal directions of the edges of an image by locating those pixel values defined by steep gray values [26]. The Prewitt operator consists of two 3 × 3 convolution masks [25–27]:

$$G_y = \begin{bmatrix} +1 & 0 & -1 \\ +1 & 0 & -1 \\ +1 & 0 & -1 \end{bmatrix} * A(x,y) \tag{2}$$

$$G_x = \begin{bmatrix} +1 & +1 & +1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} * A(x,y) \tag{3}$$

where A is the image source and * is the 2D convolution operation.

- The Laplacian operator

The Laplace operator is computed using the second-order derivative approximations of the image function $A(x, y)$. It is noise sensitive, so it is often combined with a Gaussian filter to decrease the sensitivity to noise [28]. The Laplacian filter searches the zero crossing points of the second-order derivatives of the image function and establishes the rapid changes in adjacent pixel values that belong to an edge [28,29].

$$\nabla^2 A(x,y) = \frac{\partial^2 A(x,y)}{\partial x^2} + \frac{\partial^2 A(x,y)}{\partial y^2} \tag{4}$$

A zero value indicates the areas of constant intensity, while values < 0 or > 0 are placed in the vicinity of an edge.

- The Laplacian of Gaussian (*LoG*) operator

For an image $A(x, y)$ with pixel intensity values (x, y), a combination of the Laplacian and Gaussian functions generates a new operator *LoG* [20], centered on zero and with a Gaussian standard deviation σ:

$$LoG(x,y) = -\frac{1}{\pi\sigma^4}\left[1 - \frac{x^2+y^2}{2\sigma^2}\right]e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (5)$$

The Gaussian operator suppresses the noise before using the Laplace operator for edge detection. The *LoG* operator detects areas where the intensity changes rapidly, namely the function's values are positive on the darker side (pixel values close to 0) and negative on the brighter side (pixel values close to 255) [30].

2.2.2. Dataset

The analyzed fingerprint images belong to the FVC2004 database, which is the property of the University of Bologna, Italy [31]. The image data are described, in detail, in Table 1.

Table 1. Dataset characteristics.

FVC2004 Datasets	Fingerprint Scanner	Image Size	Fingerprint Images	Total Image Number after Augmentation
BD1	Optical sensor "V300" by CrossMatch	640 × 480	80	720
BD2	Optical sensor "U.are.U 4000"	328 × 364	104	936
BD3	Thermal Sweeping Sensor "FingerChip FCD4B14CB" by Atmel	300 × 480	104	936
BD4	Synthetic fingerprint generator	288 × 384	104	936

In order to enhance the limitations of low-quality fingerprint images, both the Prewitt and *LoG* filters were used to enhance the edges that separate the ridges and valleys in the fingerprint images [9,10]. Figure 1 displays examples of image enhancement from each used database and filter.

2.2.3. Data Augmentation

The optimization of a CNN which uses small datasets means avoiding the network convergence to a local minimum. This issue is overcome using augmentation to extend the training dataset and to prevent the issue of overfitting during training. The number of images has been increased nine times. Data augmentation has been performed by executing ±30° rotations, from 0° to 360°. This provided a total of 3528 images, of which 2469 were used as training samples. Additionally, each image was resized from 256 × 256 pixels to 80 × 80 pixels to be more suitable to be fed into the network and to reduce the training time. Figure 2 shows an example of data augmentation.

2.2.4. Convolutional Neural Network

A CNN architecture aggregates convolutional modules that perform feature extraction, pooling, and fully connected layers [32]. CNNs perform well in recognition of images tasks. To optimize the performance of a CNN model, it has to be trained to extract the most important deep discriminatory features. As a general description of a CNN architecture and its learning strategy, we could mention that a gradual training process designed to solve complex concepts is employed. Thus, the early layers detect the general features of a given image. Furthermore, from layer to layer the convolution filters are trained to detect more and more complex patterns, such as object features (Figure 3). The model architecture is given in Table 2. The parameter settings and some hyperparameters selected for the proposed CNN model are also presented (Table 3). They were established before training. An epoch means the whole dataset passes once forward and backward through the neural network. Usually, the number of epochs is determined when the validation accuracy starts decreasing, even if the training accuracy is still increasing. In addition, one epoch is too large to be run through the model as a whole, so it is divided into several smaller subsamples called batches. A higher number of epochs increases the cost of computational complexity and the risk of overfitting, respectively. The variation in the number of epochs stops when the validation loss no longer improves.

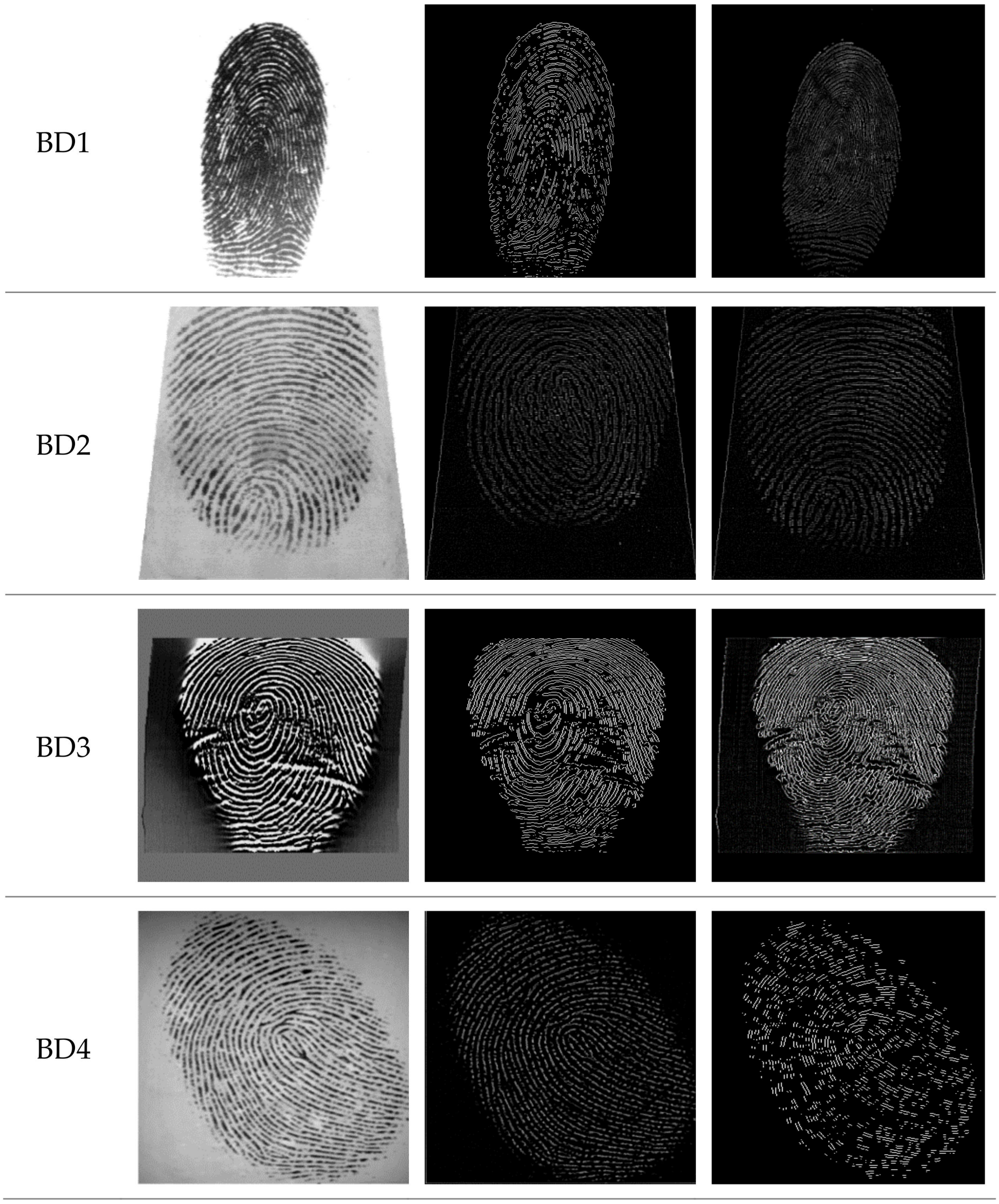

Figure 1. Samples of fingerprints from our evaluation datasets and examples of enhanced fingerprint images. Columns: **left**—raw grayscale images; **middle**—edge enhancement using the Prewitt filter; **right**—edge enhancement using the *LoG* filter.

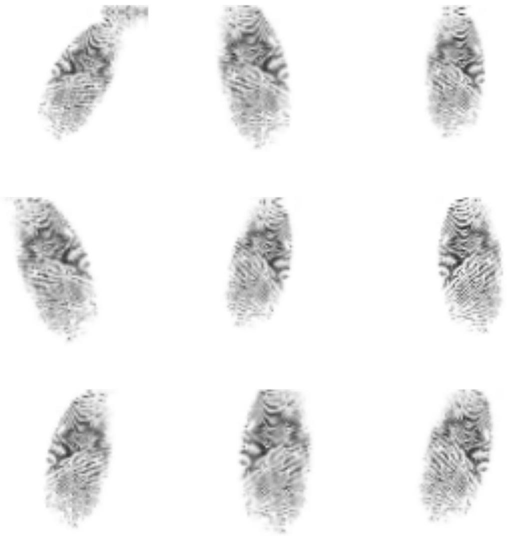

Figure 2. Data augmentation. Rotated fingerprint on ±30°.

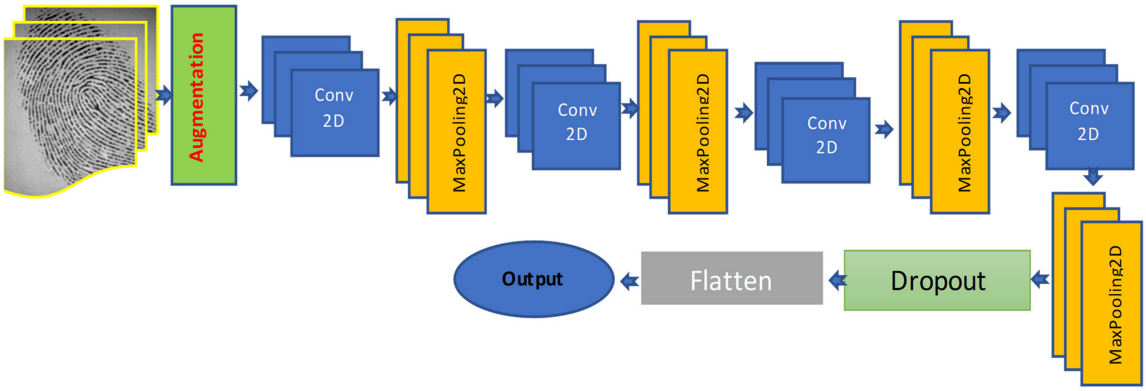

Figure 3. CNN model architecture.

Table 2. Model architecture and parameter settings.

Layer (Type)	Output Shape	Param
sequential_11 (Sequential)	(None, 80, 80, 3)	0
rescaling_7 (Rescaling)	(None, 80, 80, 3)	0
conv2d_27 (Conv2D)	(None, 80, 0, 8)	224
max_pooling2d_27 (MaxPooling2D)	(None, 40, 40, 8)	0
conv2d_28 (Conv2D)	(None, 40, 40, 16)	1168
max_pooling2d_28 (MaxPooling2D)	(None, 20, 20, 16)	0
conv2d_29 (Conv2D)	(None, 20, 20, 32)	4640
max_pooling2d_29 (MaxPooling2D)	(None, 10, 10, 32)	0
conv2d_30 (Conv2D)	(None, 10, 10, 64)	18,496
max_pooling2d_30 (MaxPooling2D)	(None, 5, 5, 64)	0
dropout_7 (Dropout)	(None, 5, 5, 64)	0
flatten_7 (Flatten)	(None, 1600)	0
dense_14 (Dense)	(None, 128)	204,928
dense_15 (Dense)	(None, 3)	387

Total parameters: 229,843; Trainable parameters: 229,843; 'None' indicates that any positive integer may be expected so that the model is able to process batches of any size.

Table 3. The hyper-parameters of the CNN model.

Hyper-Parameters	Name/Dimension
Epochs	10, 20, 30, 50
Batch size	20
Activation function	ReLU
Image size	80 × 80
Time consuming/20 epochs	23 s
Learning rate	0.01

The convolutional layers extract features from the input images. The size of the input image is reduced by the pooling layers. In addition, these pooling layers collect features, and their basic task is to reduce the feature dimensions or to reduce the feature map size by using the ReLU (rectified linear unit) activation function. The ReLU function does not activate all the neurons at the same time. If the output of the linear transformation is negative, the neurons are deactivated. The fully connected or dense layers exploit the learned high-level features and act as classifiers. Additionally, to reduce overfitting and enhance the CNN performance, a large amount of data is required. This issue is overcome by augmentation. To evaluate the performance of the proposed method, the accuracy of fingerprint identification is calculated [33].

3. Results and Discussion

The experiment was carried out in MATLAB R2018a (The MathWorks, Natick, MA, USA), using the proposed approach and the image processing toolbox. The CNN was implemented using Python (Jupyter Notebook) and the open-source platform Keras for the TensorFlow machine learning toolbox. It is run in Google Colaboratory (Colab).

The image datasets were stored in Google Drive and the workspaces were connected. Of the total dataset, 60%, 20%, and 20% were used as the training set, validation set, and test set, respectively. The performance of the CNN model training and validation classification accuracy rate over the 10, 20, 30, and 50 epochs are shown in Figures 4–7. The classification accuracy is defined as the ratio between the correct predictions and the total number of predictions in the training or validation data.

As shown in Figures 4–7, during the training of the CNN, the accuracy of the training set (blue line) continued to increase and the network was learning constantly. The validation set (orange line) first increased, then overfitting occurred and the accuracy showed an unstable variation. The same behavior was observed for loss curves. Consequently, we investigated the number of epochs and selected the best number to solve the overfitting problem.

The number of epochs was set to 10, 20, 30, and 50 in order to keep the training time at an acceptable value of 1.8 s/epoch, on average. Prior to each training epoch, the training data was randomly shuffled. The performance of the proposed model is summarized in Table 4.

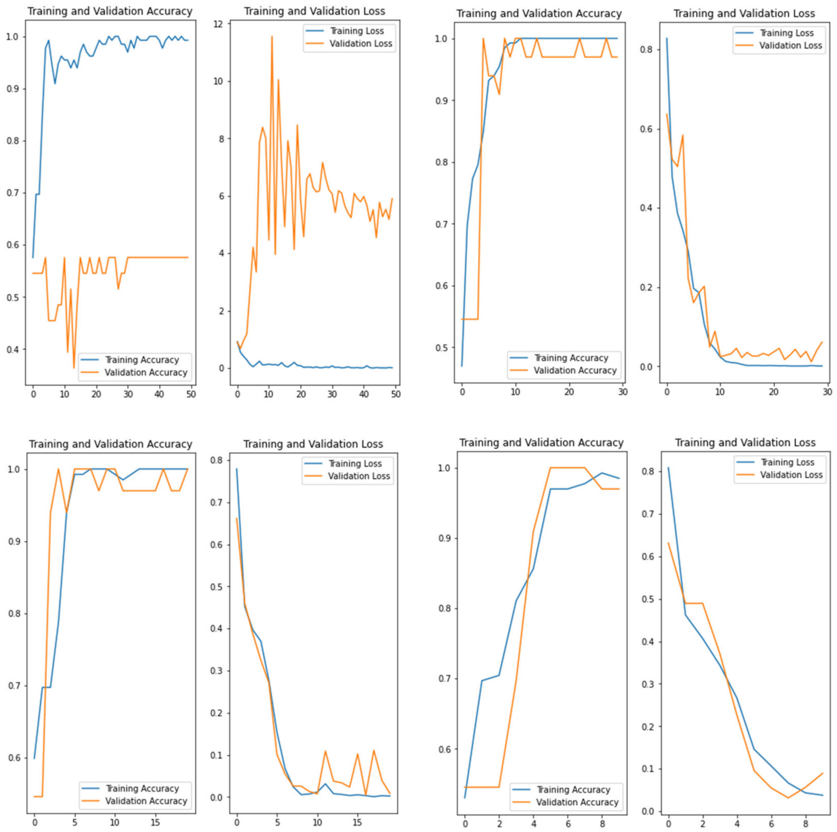

Figure 4. Illustration of model accuracy rate for 50, 30, 20, and 10 epochs for DB1 dataset acquired using an optical sensor "V300" by CrossMatch.

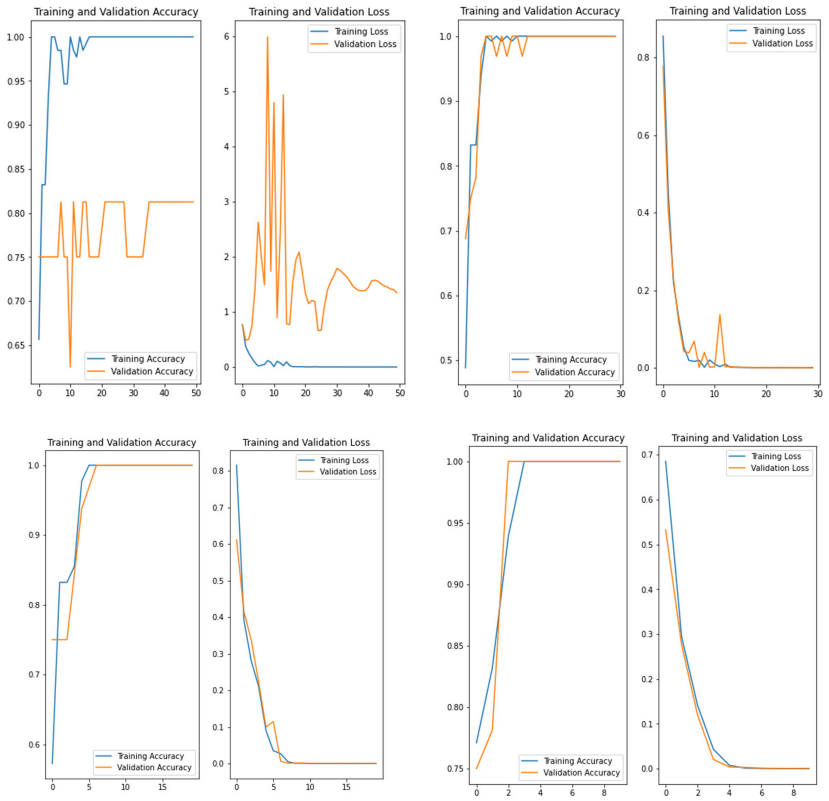

Figure 5. Illustration of model accuracy rate for 50, 30, 20, and 10 epochs for DB2 dataset acquired using an optical sensor "U.are.U 4000".

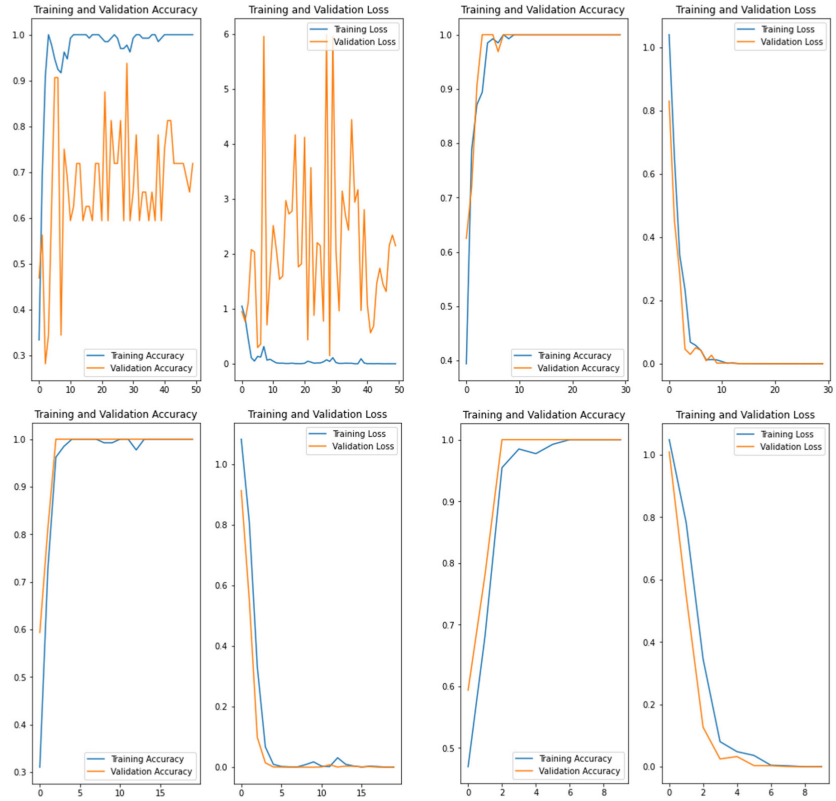

Figure 6. Illustration of model accuracy rate for 50, 30, 20, and 10 epochs for DB3 dataset acquired using a thermal sweeping sensor "FingerChip FCD4B14CB" by Atmel.

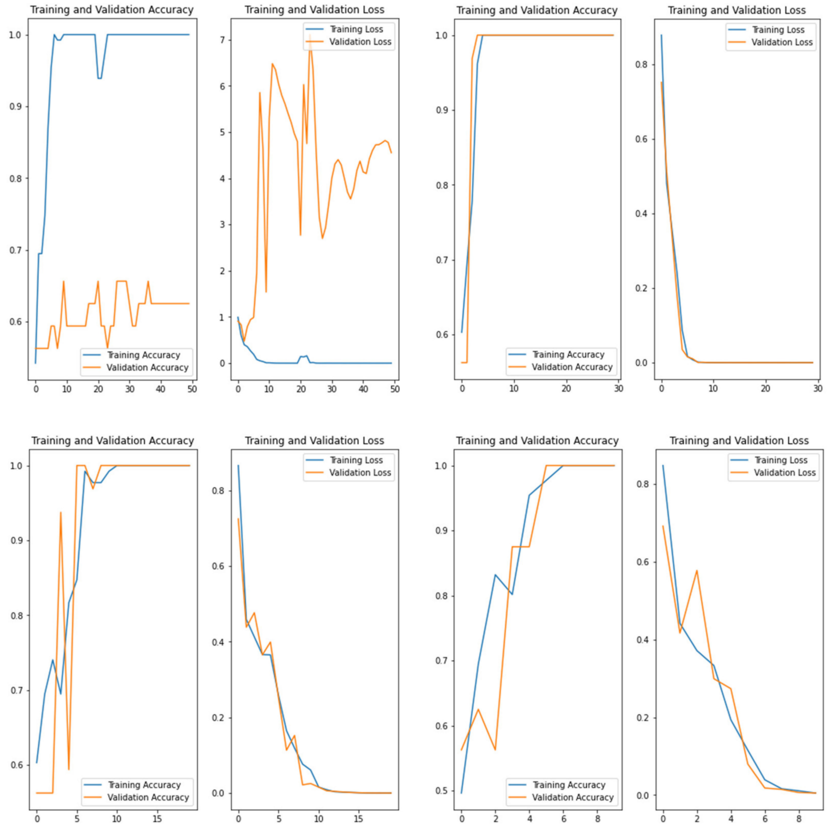

Figure 7. Illustration of model accuracy rate for 50, 30, 20, and 10 epochs for DB4 dataset generated as synthetic fingerprints.

Table 4. The performance of the proposed model.

Database	Number of Test Samples	Validation Accuracy (%)	Validation Loss	Test Accuracy (%)	
				Prewitt Filter	*LoG* Filter
BD1	144	98.7	0.0586	69.8	75.6
BD2	187	67.6	3.1061	62.5	70.2
BD3	187	94.7	0.1931	71.6	73.4
BD4	187	98.7	0.0344	69.8	75.6

The results in Figures 4–7 indicate that, while the accuracy and loss of the training data have very good values, the accuracy and loss of the validation dataset is influenced by the epoch number, indicating the existence of overfitting or underfitting. Our data indicates the overfitting of the model starting with the seventh epoch. In this case, it is necessary to stop the model early by tuning the hyperparameter. The CNN performance is strongly influenced by the quality of a fingerprint image and by its local and global structures. The accuracy of the proposed CNN model depends on the amount and quality of the training images, which in our case show an important variability from dataset to dataset. The performance on the test data (20% of each dataset) is lower than the accuracy provided by the training data, with the amendment that the number of samples is reduced for the test set. However, the LoG filter increased the accuracy compared to the Prewitt filter, and it is a better solution to enhance the edges in the fingerprint images. Additionally, the raw images in the DB2 dataset that were acquired using the optical sensor "U.are.U 4000" had a low quality that affected the performance of the classification.

According to the data in Table 4, the accuracy values determined in the training and validation sets are in line with the reported results in the literature. In [3], an accuracy of 85% was reported for images belonging to the FVC2004 database which were processed using a CNN-based automatic latent fingerprint matching system which uses the local minutiae features. Mohamed et al. [14] reported a 99.2% classification accuracy for the training set in an experiment which used the NIST DB4 dataset and 4000 fingerprint images. Militello et al. [15] reported an accuracy value of 91.67% for a pre-trained CNN, used together with the PolyU and NIST fingerprint databases.

The accuracy values determined for the test set were slightly worse, thus indicating a smaller drop in performance. However, we have mentioned that our proposed method used the whole fingerprint images and the computation time is small compared to the other reported method. As an example, in [34], an accuracy of 94.4% and a testing time of 39 ms/image were reported for a pre-trained CNN architecture of the VGG-F network type, and an accuracy of 95.05% and a testing time of 77 ms/image for the VGG-S network.

In addition, CNN architectures have some drawbacks, such as a poor generalization capacity, a requirement for a huge training dataset, and a low stability to geometrical deformation and rotation. In the proposed study, the low generalization capacity was overcome by increasing the training data size to allow the network to train on as many samples as possible. The obtained results indicate that the CNN performance is greatly influenced by the quality of the fingerprint images. The low stability of the network is due to the diversity of finger scanners which were used to acquire the fingerprints, such as optical and thermal sweeping sensors, and synthetic fingerprints as well. Finding a solution to provide a good performance of classification for this variety of data was a big challenge for our method. Our approach integrates the monitoring of training and validation by setting the number of epochs as a form of regularization, and learning curve graphs to decide on the model convergence.

4. Conclusions

The work conducted in this paper is mainly devoted to fingerprint identification using a CNN network that can perform fingerprint classification by considering whole fingerprint images. The proposed algorithm uses poor-quality original raw fingerprint images. These were processed using Prewitt and Laplace filters to enhance the edges and, in order to reduce the expensive training cost, data resizing was applied. Hyper-parameter tuning, using various epoch numbers, was considered to improve the performance of classification. Our results indicate the overfitting of the model starting with the seventh epoch. The classification accuracy varied from 67.6% to 98.7% for the validation set, and from 70.2% to 75.6% for the test set. Following these considerations, we would argue that the proposed method can achieve a very good performance compared to the traditional hand-crafted features method, despite the fact that it uses raw data and does not perform any handcrafted feature extraction operations.

For future developments, we are interested in improving the performance of classification by using other pre-processing techniques correlated to extensive hyper-parameter tuning. Additionally, other fingerprint databases will be used to assess the generalization capabilities of CNN architectures.

Author Contributions: Conceptualization, S.M. and L.M.; methodology, S.M. and L.M.; software, S.M., A.-M.D.L. and L.M.; validation, S.M. and A.-M.D.L.; formal analysis, A.-M.D.L.; investigation, S.M. and A.-M.D.L.; writing—original draft preparation, S.M., A.-M.D.L. and L.M.; writing—review and editing, S.M. and L.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors thank the anonymous referees whose comments helped to improve the paper.

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

References

1. Jain, A.K. An Introduction to Biometric Recognition. *IEEE Trans. Circuits Syst. Video Technol.* **2004**, *14*, 4–20. [CrossRef]
2. Maltoni, D.; Maio, M.; Jain, A.K.; Prabhakar, S. Fingerprint analysis and representation. In *Handbook of Fingerprint Recognition*; Springer Professional Computing; Springer: New York, NY, USA, 2003; pp. 83–130.
3. Deshpande, U.U.; Malemath, V.S.; Patil Shivanand, M.; Chaugule Sushma, V. A Convolution Neural Network-based Latent Fingerprint Matching using the combination of Nearest Neighbor Arrangement Indexing. *Front. Robot. AI* **2020**, *7*, 113. [CrossRef] [PubMed]
4. Militello, C.; Conti, V.; Sorbello, F.; Vitabile, S. A novel embedded fingerprints authentication system based on singularity points. In Proceedings of the Second International Conference on Complex, Intelligent and Software Intensive Systems (CISIS 2008), Technical University of Catalonia, IEEE Computer Society, Barcelona, Spain, 4–7 March 2008; pp. 72–78.
5. Conti, V.; Militello, C.; Sorbello, F.; Vitabile, S. Introducing pseudo-singularity points for efficient fingerprints classification and recognition. In Proceedings of the 4th International Conference on Complex, Intelligent and Software Intensive Systems (CISIS-2010), Krakow, Poland, 15–18 February 2010; pp. 368–375.
6. Saponara, S.; Elhanashi, A.; Zheng, Q. Recreating Fingerprint Images by Convolutional Neural Network Autoencoder Architecture. *IEEE Access* **2021**, *9*, 147888–147899. [CrossRef]
7. Deshpande, U.U.; Malemath, V.S.; Chaugule, S.V. Automatic latent fingerprint identification system using scale and rotation invariant minutiae features. *Int. J. Inf. Tecnol.* **2022**, *14*, 1025–1039. [CrossRef]
8. Wang, T.; Zheng, Z.; Bashir, A.K.; Jolfaei, A.; Xu, Y. FinPrivacy. A privacy-preserving mechanism for fingerprint identification. *ACM Trans. Int. Technol.* **2021**, *21*, 56. [CrossRef]
9. Dhar, R.; Gupta, R.; Baishnab, K.L. An analysis of Canny and Laplacian of Gaussian image filters in regard to evaluating retinal image. In Proceedings of the International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE), Coimbatore, India, 6–8 March 2014.
10. Kumar, S.N.; Fred, L.; Haridhas, A.K.; Varghese, S. Medical image edge detection using gauss gradient operator. *J. Pharm. Sci. Res.* **2017**, *5*, 695–704.
11. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A. Inception-v4, Inception-Resnet and the impact of residual connections on learning. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.

12. Zhu, Y.; Yin, X.; Jia, X.; Hu, J. Latent fingerprint segmentation based on convolutional neural networks. In Proceedings of the IEEE Workshop on Information Forensics and Security, Rennes, France, 4–7 December 2017; pp. 1–6.
13. Shrein, J.M. Fingerprint classification using convolutional neural networks and ridge orientation images. In Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI), Honolulu, HI, USA, 7 November–1 December 2017.
14. Mohamed, M.H. Fingerprint Classification Using Deep Convolutional Neural Network. *J. Electr. Electron. Eng.* **2021**, *9*, 147–152. [CrossRef]
15. Militello, C.; Rundo, L.; Vitabile, S.; Conti, V. Fingerprint Classification Based on Deep Learning Approaches: Experimental Findings and Comparisons. *Symmetry* **2021**, *13*, 750. [CrossRef]
16. Wang, J.-W.; Tuyen Le, N.; Wang, C.-C.; Lee, J.-S. Enhanced ridge structure for improving fingerprint image quality based on a wavelet domain. *IEEE Signal Process. Lett.* **2015**, *22*, 390–394. [CrossRef]
17. Yang, J.; Xiong, N.; Vasilakos, A.V. Two-stage enhancement scheme for low-quality fingerprint images by learning from the images. *IEEE Trans. Hum. Mach. Syst.* **2013**, *43*, 235–248. [CrossRef]
18. Borra, S.R.; Jagadeeswar Reddy, G.; Sreenivasa Reddy, E. Classification of fingerprint images with the aid of morphological operation and AGNN classifier. *Appl. Comput. Inform.* **2018**, *14*, 166–176. [CrossRef]
19. Listyalina, L.; Mustiadi, I. Accurate and low-cost fingerprint classification via transfer learning. In Proceedings of the 2019 5th International Conference on Science in Information Technology, Yogyakarta, Indonesia, 23–24 October 2019.
20. Tertychnyi, P.; Ozcinar, C.; Anbarjafari, G. Low-quality fingerprint classification using deep neural network. *IET Biom.* **2018**, *7*, 550–556. [CrossRef]
21. Pandya, B.; Cosma, G.; Alani, A.A.; Taherkhani, A.; Bharadi, V.; McGinnity, T.M. Fingerprint classification using a deep convolutional neural network. In Proceedings of the 2018 4th International Conference on Information Management, Oxford, UK, 25–27 May 2018.
22. Nur-A, A.; Ahsa, M.; Based, M.A.; Kowalski, M. An intelligent system for automatic fingerprint identification using feature fusion by Gabor filter and deep learning. *Comput. Electr. Eng.* **2021**, *95*, 107387. [CrossRef]
23. Trabelsi, S.; Samai, D.; Dornaika, F.; Benlamoudi, A.; Bensid, K.; Taleb-Ahmed, A. Efficient palmprint biometric identification systems using deep learning and feature selection methods. *Neural Comput. Appl.* **2022**, 1–23. [CrossRef]
24. Oleiwi, B.K.; Abood, L.H.; Farhan, A.K. Integrated different fingerprint identification and classification systems based deep learning. In Proceedings of the 2022 International Conference on Computer Science and Software Engineering (CSASE), Duhok, Iraq, 15–17 March 2022; pp. 188–193.
25. Kumar, S.; Singh, M.; Shaw, D.K. Comparative Analysis of Various Edge Detection Techniques in Biometric. *Int. J. Eng. Technol.* **2016**, *8*, 2452–2459. [CrossRef]
26. Moldovanu, S.; Moraru, L.; Stefanescu, D.; Bibicu, D. Edge-preserving filters in a boundary options context. *Ann. Dunarea Jos Univ. Galati Math. Phys. Theor. Mech.* **2017**, *1*, 51–57.
27. Sun, Q.; Hou, Y.; Tan, Q.; Li, C.; Liu, M. A robust edge detection method with sub-pixel accuracy. *Optik JLEO* **2014**, *125*, 3449–3453. [CrossRef]
28. Baareh, A.; Al-Jarrah, A.; Smadi, A.; Shakah, G. Performance Evaluation of Edge Detection Using Sobel, Homogeneity and Prewitt Algorithms. *J. Softw. Eng. Appl.* **2018**, *11*, 537–551. [CrossRef]
29. Cui, S.; Wang, Y.; Qian, X.; Deng, Z. Image Processing Techniques in Shockwave Detection and Modeling. *J. Signal Inf. Process.* **2013**, *4*, 109–113. [CrossRef]
30. Moraru, L.; Moldovanu, S.; Pană, L. Edges identification based on the derivative filters and fractal dimension. *Ann. Dunarea Jos Univ. Galati Math. Phys. Theor. Mech.* **2019**, *1*, 34–42. [CrossRef]
31. FVC2004: Third Fingerprint Verification Competition. Available online: http://bias.csr.unibo.it/fvc2004/databases.asp (accessed on 10 February 2022).
32. Canziani, A.; Paszke, A.; Culurciello, E. An analysis of deep neural network models for practical applications. *arXiv* **2016**. [CrossRef]
33. Damian, F.; Moldovanu, S.; Moraru, L. Color space influence on ANN skin lesion classification using statistics texture feature. *Ann. Dunarea Jos Univ. of Galati Math. Phys. Theor. Mech.* **2021**, *1*, 53–62. [CrossRef]
34. Michelsanti, D.; Ene, A.; Guichi, Y.; Stef, R.; Nasrollahi, K.; Moeslund, T.B. Fast fingerprint classification with deep neural networks. In *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISAPP, Porto, Portugal, 27 February–1 March 2017*; Scitepress: Setúbal, Portugal, 2017; Volume 5, pp. 202–209.