

Article

Fingerprint Classification Based on Deep Learning Approaches: Experimental Findings and Comparisons

Carmelo Militello ^{1,*}, Leonardo Rundo ², Salvatore Vitabile ³ and Vincenzo Conti ⁴

- ¹ Institute of Molecular Bioimaging and Physiology, Italian National Research Council (IBFM-CNR), 90015 Cefalù, Italy
- ² Department of Radiology, University of Cambridge, Cambridge CB2 0QQ, UK; lr495@cam.ac.uk
- ³ Department of Biomedicine, Neuroscience and Advanced Diagnostics (BiND), University of Palermo, 90127 Palermo, Italy; salvatore.vitabile@unipa.it
- ⁴ Faculty of Engineering and Architecture, University of Enna KORE, 94100 Enna, Italy; vincenzo.conti@unikore.it
- * Correspondence: carmelo.militello@ibfm.cnr.it

Abstract: Biometric classification plays a key role in fingerprint characterization, especially in the identification process. In fact, reducing the number of comparisons in biometric recognition systems is essential when dealing with large-scale databases. The classification of fingerprints aims to achieve this target by splitting fingerprints into different categories. The general approach of fingerprint classification requires pre-processing techniques that are usually computationally expensive. Deep Learning is emerging as the leading field that has been successfully applied to many areas, such as image processing. This work shows the performance of pre-trained Convolutional Neural Networks (CNNs), tested on two fingerprint databases—namely, PolyU and NIST—and comparisons to other results presented in the literature in order to establish the type of classification that allows us to obtain the best performance in terms of precision and model efficiency, among approaches under examination, namely: AlexNet, GoogLeNet, and ResNet. We present the first study that extensively compares the most used CNN architectures by classifying the fingerprints into four, five, and eight classes. From the experimental results, the best performance was obtained in the classification of the PolyU database by all the tested CNN architectures due to the higher quality of its samples. To confirm the reliability of our study and the results obtained, a statistical analysis based on the McNemar test was performed.



Citation: Militello, C.; Rundo, L.; Vitabile, S.; Conti, V. Fingerprint Classification Based on Deep Learning Approaches: Experimental Findings and Comparisons. *Symmetry* **2021**, *13*, 750. <https://doi.org/10.3390/sym13050750>

Academic Editor: Jan Awrejcewicz

Received: 22 March 2021

Accepted: 19 April 2021

Published: 26 April 2021

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



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: fingerprint classification; deep learning; convolutional neural networks; fingerprint features

1. Introduction

Traditionally, in an Automated Fingerprint Identification Systems (AFIS), each fingerprint (or the derived biometric characteristics) must be compared with all the others contained in the database. If the database is large, a high number of comparisons is required, thus resulting in long processing and latency times. On the contrary, the availability of a system capable of classifying the fingerprints, before the identification phase, allows for reducing the 'search set', as well as the response times of any biometric identification system in real-time applications. Especially, when the identification process is performed on partial fingerprint images [1], it is especially important to rely on a division into classes to reduce the computational cost of searching for a fingerprint image.

Biometric systems, like all security systems, have vulnerabilities. A vulnerability in biometric security results in incorrect recognition or failure to correctly recognize individuals. The vulnerability concept includes methods to wrongly accept an individual (i.e., spoofing) affect the overall system performance (i.e., denial of service), or to attack another system via leaked data (i.e., identity theft) [2]. Exactly in this context and in order to

guarantee better performance in fingerprint recognition, the classification phase is crucial and important to reduce the search space and, at the same time, also the computational cost.

In recent years, an unprecedented number of achievements in the field of Artificial Intelligence and, especially in the field of Deep Learning, have been obtained. In image processing and computer vision, the main advantage of Deep Learning over traditional approaches is that it does not impose the need for hand-crafted feature extraction and pre-processing pipelines. The classification task is indeed performed directly on the raw data (i.e., input images). In multiple domains, including biometrics, Deep Learning techniques have shown a great success in supervised learning tasks [3,4]. As a matter of fact, biometrics, which deals with reliably identifying people in critical application domains, can leverage the unique potential offered by Deep Learning [5]. In this context, symmetry plays a crucial role in biomedical image analysis [6] and biometry [7], especially in terms of linear symmetry features in fingerprint classification [8,9].

In the proposed work, a software solution was developed to classify fingerprints into 4, 5 and 8 categories, by using Convolutional Neural Networks (CNNs) on three digital databases. This work aims to highlight the potential of CNN architectures, by effectively reducing the data set to be processed during the search and identification of a fingerprint, thus allowing for a classification based on certain features.

With more details, two publicly available fingerprint databases were employed in this work—namely, Hong Kong Polytechnic University (PolyU) [10] and National Institute of Standards and Technology (NIST) [11]—which together constitute a total database of 7800 images. We implemented and tested three well-established CNNs: AlexNet [12], GoogLeNet [13], and ResNet [14].

The main goal of this study is to understand the best performing approach among the three CNN architectures considered, considering both the type of database used and the type of classification. Such a development allowed us to implement a user-friendly application, equipped with a Graphical User Interface (GUI), which provides the user with the performance of each CNN associated with the various databases. In a very simple manner, it is possible to: (i) calculate the performance of each network, using a database and a different classification task; (ii) load an image from any database and proceed with its classification by choosing the neural network and number of classes; (iii) identify the best performing CNNs for any fingerprint image. These properties make the GUI an easy, yet important, analysis and testing tool. In particular, the application uses the CNNs to analyse the characteristics of the fingerprints and exploits a transfer learning process, which allows us to refine an existing model, previously trained, and adapt it to the classification problem at hand.

The main contributions of this work are:

- thorough analysis using three different pre-existing CNN-based architectures on fingerprint classification (into four, five, and eight classes), which can improve fingerprint recognition when integrated into an AFIS;
- considerable number of tests performed, using two distinct large-scale fingerprint databases, aimed to acquire further insights into performance achieved on different databases with heterogeneous characteristics;
- a user-friendly tool offering an immediate interaction and performance assessment via a GUI;
- statistical validation of the obtained findings and comparisons by means of the McNemar test.

In order to confirm the goodness of the approaches and related obtained results, the p -values of the McNemar statistical test—on the pairwise comparisons of the classification results achieved by the investigated CNNs—considering the experiments described have been calculated and discussed.

This paper is structured as follows. Section 2 summarizes the relevant background and principles on fingerprint classification. Section 3 outlines the main literature approaches for fingerprint classification. Section 4 illustrates the used databases and the Deep Learning

methods exploited to perform the comparison among CNN architectures. Section 5 shows the experiments and the obtained classification performance. Finally, Section 6 provides discussion and draws conclusions.

2. Fingerprint Classification

The identification process of a person—for instance, in an AFIS—requires the comparison of a fingerprint with all the others stored in a specific archive. These archives can grow to a substantial size. Therefore, millions of fingerprint comparisons may be required.

Typically, the first phase for any large-scale fingerprint recognition system (for both software and hardware systems) is fingerprint classification, aimed to provide an indexing mechanism for reducing the number of comparisons. In fact, for efficient one-to-many fingerprint recognition, fingerprint classification is an important step because assessing the target fingerprint only with the same type of fingerprints helps save searching time and the total time of one-to-one recognition. Therefore, fingerprint classification can reduce the execution time of the whole recognition procedure without affecting the performance [15].

Using a fingerprint classification system, it is possible to considerably reduce the number of comparisons—by processing only the fingerprints belonging to the same class as the fingerprint to be identified—and efficiently improve the AFIS performance.

2.1. Types of Fingerprint Classification

Since the 19th century, several people have been interested in the classification of fingerprints into different types: in 1823 Jan Evangelista Purkinje [16] discussed a thesis where the fingerprints were classified into nine different types.

A significant turning point in the classification of fingerprints was given by Francis Galton [17,18] at the end of the 19th century, who described for the first time the so-called “deltas” (i.e., points where three different areas of the fingerprint converge) by determining the conformation of a ridge. Galton proposed a fingerprint classification into three categories.

A few years later, Edward Henry proposed a classification system into five classes [19], which is still frequently applied. Inspired by Henry’s classification model, the Federal Bureau of Investigation (FBI) presented in 1985 the book “The Science of Fingerprints” [20], where eight fingerprint types were described.

2.2. Fundamental Elements for Fingerprint Classification

Three types of points can be identified in a fingerprint (Figure 1) and starting from their symmetric properties used to classify fingerprint images [8,9]:

- *Core*: approximately coincides with the center of the ridge pattern. Each fingerprint has only a core point. In practice, it consists of the point with the greatest curvature of the innermost ridge that forms the spiral.
- *Delta*: represents a divergence point and identifies a stretch where two ridges, which draw an almost parallel path, are divided. The delta is also defined as the point in the centre of a triangular region, which is usually found in the lower right or left corner, wherein the ridges converge from different directions.
- *Loop*: is characterized by several ridges that cross the imaginary line between the core point and the delta point, thus forming a “U” pattern; the loops return approximately in the direction from which they originated, since they are indeed characterized by having exactly one loop point and one core.

Based on these three characteristics (core, delta, and loop), the various types of fingerprints can be distinguished.

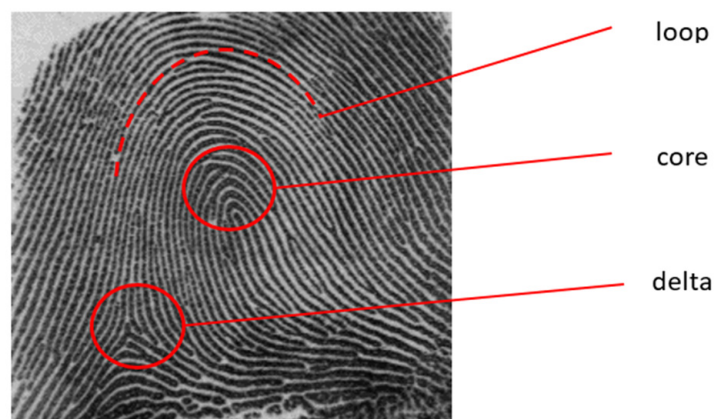


Figure 1. Example of delta and core points within a fingerprint that define symmetric characteristics among similar images. Moreover, the “U” pattern of the loop is highlighted with a dashed line.

2.3. Fingerprint Classes

Relying upon a careful analysis of the three elements described above (i.e., core, delta and loop) and the related their symmetric characteristics, it is possible to characterize the fingerprint into several classes:

- *Arch*: they are distinguished by the fact that the ridges enter from one side, rise forming a small protuberance and exit from the opposite side. They do not have loops and deltas.
- *Tented Arch*: if the arch has at least one ridge showing a high curvature, and there is the presence of a loop and a delta, they can be classified as Tented Arch.
- *Left Loop*: the ridges that form it come and return towards the left direction. They have a loop and a delta (core positioned to the left of the delta).
- *Right Loop*: characterized by one or more ridges that enter from the right side, curve and exit from the same side. They have a loop and a delta (core positioned to the right of the delta).
- *Whorl*: it is characterized by the presence of at least two delta points. The pattern is defined by a part of ridges that tend to form a circular pattern. The fingerprints of this class are characterized by at least one ridge that makes a complete 360° turn around the fingerprint center.

Each fingerprint class has a different incidence, in terms of frequency; the most common are those of the Whorl class, while the Arch class that has a lower number of occurrences. Figure 2 shows an example for each of the five classes according to Henry’s classification.

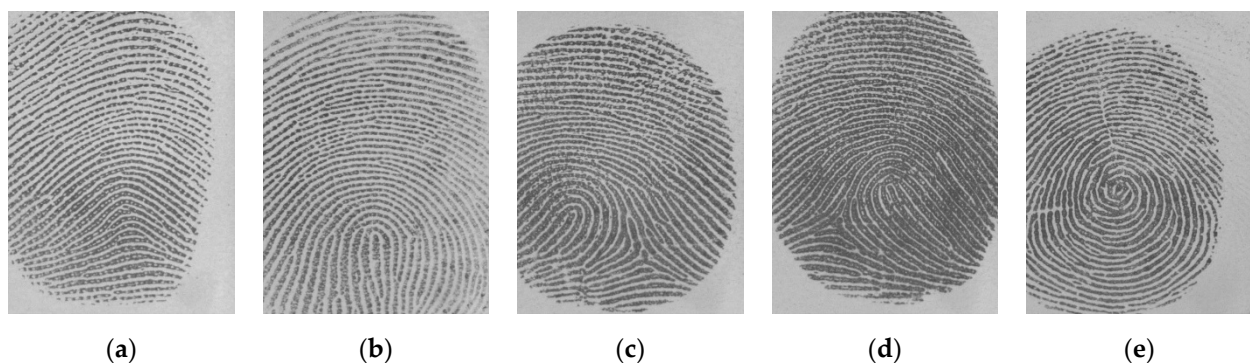


Figure 2. Classification into 5 classes and the corresponding frequency (in percentage) [21]: (a) Arch (3.7%); (b) Tented Arch (2.9%); (c) Left Loop (33.8%); (d) Right Loop (31.7%); (e) Whorl (27.9%).

Fingerprints belonging to the Whorl class can be divided into further subclasses:

- *Plain Whorl*: they can be distinguished by drawing an ideal line that joins the two delta points, if this does not intersect the area of the ridges that form a circle then it is a Plain subclass fingerprint.
- *Central Pocket Whorl*: also in this case, it is necessary to draw a line joining the two delta points identified, if it intersects the area of the ridges that form a circle then it can be said that the fingerprint belongs to the Central Pocket subclass.
- *Double Loop Whorl*: when it is possible to distinguish two loops, which meet, the fingerprint is defined as a Double Loop subclass.
- *Accidental Loop Whorl*: this category includes all the fingerprints that contain more than two deltas and all those that do not clearly belong to any of the three previous subclasses.

Figure 3 shows some examples of the four subclasses of the whorl class.

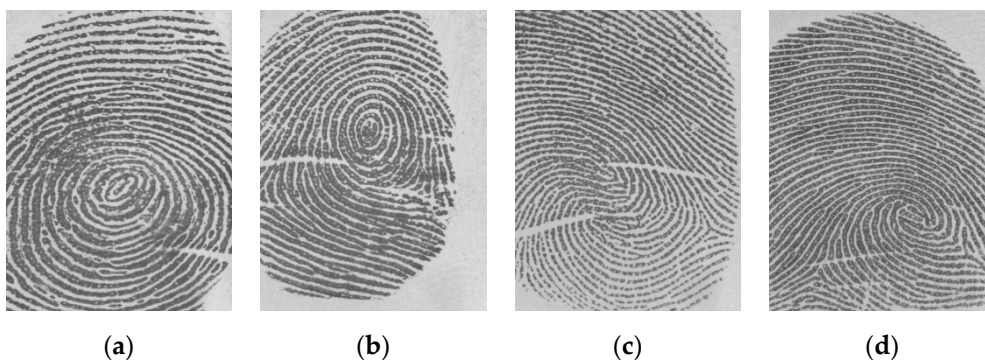


Figure 3. Subclasses of the whorl class: (a) Plain Whorl; (b) Central Pocket Whorl; (c) Double Loop Whorl; (d) Accidental Loop Whorl.

2.4. Fingerprint Classification Using Deep Learning Approaches

Traditional methods for fingerprint classification rely on morphological and symmetric characteristics, based on local fingerprint substructures [22]. Given the difficulty in detecting them, and since some are very similar to each other, most systems use only the two most prominent structures [23]: *Ridge Ending* and *Ridge Bifurcation* (i.e., minutiae) which correspond to the terminations and bifurcations of the ridges, respectively. To achieve this, it is necessary to implement an algorithm, which can detect the minutiae, by taking into account various factors, such as the fingerprint acquisition method or the type of sensor used. These factors entail the need to adapt the identification algorithm to different circumstances, considerably affecting management costs and processing times.

CNN-based classifiers do not require any human intervention in the feature extraction and classification steps, introducing a drastic reduction in design times, cost reduction and ensuring greater flexibility.

The goal of this work is to classify fingerprints into four, five, and eight classes, by making use of CNN-based architectures, comparing them and providing guidelines on the best architecture according to the fingerprint database and the classification type.

3. Related Work

Generally, the first phase for any large-scale fingerprint recognition system (for both software and hardware systems) [24,25] is fingerprint classification [26]. Classifying a fingerprint image is crucial and represents a very difficult morphological structure recognition problem, due to variability of inter-class and intra-class features and for the presence of noise. Over the last thirty years, many techniques have been devised to overcome these problems, such as genetic algorithms [27,28], support vector machines [29], Wavelet and Fourier transforms [30], neural networks [31,32], statistical information [33]. However, the

main proposed approaches can be classified as follows: (i) based on heuristics or singularity points, (ii) structure- or morphology-based, and (iii) Neural- and CNN-based. The main approaches are outlined in what follows.

3.1. Approaches Based on Heuristics/Singularity Points

Solutions based on heuristics are generally focused on fingerprint ridge structures and singularity points. In [34], the authors proposed a classification approach exploiting singular points (extracted by means of a modified Poincare index) and the orientation field for a reliable and fast fingerprint classification. The algorithm was tested on two databases: (i) the Fingerprint Verification Competition (FVC) 2002 [35] and the FVC2004 [36] from the Biometric System Laboratory, University of Bologna, Italy, and (ii) the VeriFinger_Sample_DB from the Neurotechnologija website [37]. The whole set of 730 fingerprint images was classified into five different classes. The classification results showed an overall classification rate of 96.1%. Among all images, the algorithm failed six times during the core point detections, therefore all the remaining images were correctly classified. Unfortunately, singularity points are not always present in a fingerprint image due to an incorrect acquisition process (e.g., the acquired fingerprint might be partial) or the fingerprint belongs to particular classes (e.g., arch). For this reason, a new set of points—called pseudo-singularity-points—were defined and used to enable the classification as the first phase in an identification system [38]. As a result, fingerprint processing and fingerprint matching involve few steps to compare features obtaining performance comparable to the conventional minutiae-based systems. The experiments were performed on several official FVC databases. The achieved results show an identification system with a False Acceptance Rate (FAR) = 1.22% and a False Rejection Rate (FRR) = 9.23% on the FVC2002 DB2-A database; thus, a satisfactory effectiveness of the proposed approach was obtained [35]. The best results were obtained on the FVC2000 DB1-B [39] database, with a FAR = 0.26% and a FRR = 7.36%. The authors of [40] developed an efficient estimation approach to increase the accuracy of the extracted directional field. According to the improved directional field and for generating the input features of the fingerprint classifier, singular points and the related macro-features were extracted. After encoding the input features, to perform a five-class fingerprint classification, a fuzzy Wavelet neural network (FWNN)-based classifier was applied. The experiments were performed using the FWNN-based classifier on the NIST-4 database obtaining an overall accuracy of 92.4%.

3.2. Structure/Morphology-Based Approaches

Approaches based on morphological information are generally focused on fingerprint structural and information shape. Nain et al. in [41] used the Sobel operator and Gabor filter to extract the high ridge curvature region. Subsequently, the ridges within the region were traced in both the directions, starting from any point on the ridge. Vectors were drawn at the end points of the ridge in order to determine the class to which a fingerprint image belongs. The algorithm was very efficient by avoiding the core point detection, and was based on simple ridge-flow connectivity. The overall classification accuracy without rejection was 98.75%. The authors of [42], for enhancing the fingerprint image quality, proposed a three-stage method. The fingerprint image was first processed by a denoising procedure based on the wave atom transform. Afterwards, the image augmentation (based on morphological operations) was exploited for improving the classification performance. The dilation and the area opening were the morphological operators considered for refinements. To evaluate the performance of the approach, the authors used an adaptive genetic neural network and the FVC2000 databases [39]. A fingerprint classification approach, based on histograms of oriented gradient descriptors, was proposed in [43]. The computed orientation field, which was adapted to the ridge patterns and incorporated into the proposed descriptor, better represented a fingerprint in a robust manner compared to the non-adapted field. An Extreme Learning Machine (ELM) with a Radial Basis Function (RBF) kernel was used as a classifier. The experimental tests were conducted on the FVC2004 [36] database, obtaining

a mean accuracy of 98.70%. In [44], the authors proposed a neural network model using as input the local ridge orientation information and the corresponding locations of the fingerprint singular points. The local ridge orientation was obtained by filtering the raw fingerprint to decrease errors due to noisy of the scanned fingerprint. From the resulting image, singular points were extracted and their positions were used as the input to the Sim-Net unsupervised neural network model. The number of the singular points and the distance between these points can adjust the number of classes.

3.3. Neural- and CNN-Based Approaches

In recent years, Deep Learning has been achieving outstanding results in many application fields, such as image processing and computer vision. An attempt on the fingerprint classification problem by using a CNN-based method was performed in [45]. For the four-class problem on the NIST-DB4 database, only choosing the orientation field as the classification feature, they achieved 91.4% accuracy, by using the stacked sparse auto-encoders with three hidden layers. Afterwards, two classification probabilities were considered for fuzzy classification, which can affect classification performance. By only adjusting the probability threshold, they obtained a classification accuracy of 96.1% (setting a threshold of 0.85), 97.2% (setting a threshold of 0.90) and 98.0% (setting a threshold of 0.95) with a single-layer architecture. The authors of [46] proposed a fingerprint classification algorithm, which can classify raw fingerprint images. The low computational complexity of the proposed algorithm was obtained by using transfer learning rather than training a deep CNN architecture. The NIST-4 database has been used to test the performance of the proposed algorithm. In the experimental part, the proposed algorithm achieved 94.7% and 96.2% for the five-class and four-class classification problems, respectively. Hamdi et al. in [47] investigated the use of the conic Radon transform as a feature extractor and a Deep Learning technique to solve fingerprint classification tasks. The used Radon technique (which represents an extension of classical Radon transform over conic sections) enabled the extraction of fingerprint global characteristics, which are invariant to geometrical transformations, such as translations and rotations. This approach was tested on the NIST-SD4 achieving a recognition rate of 96.5%. In [47,48], the authors proposed Res-FingerNet, a deep CNN to tackle the classification task. Moreover, to reduce the intra-class variance and increase the inter-class variance of the fingerprints, they utilized a center loss in the network training phase so that the learned deep features were more discriminative for fingerprint classification tasks. By using the center loss the classification accuracy increased about 1.5%. The performance of the approach was evaluated on NIST-DB4 database, achieving a classification accuracy of 97.9%. In [49], the authors assessed the performance of two pre-trained CNN architectures, namely VGG-F and VGG-S Nets, fine-tuned on the NIST SD4 database. The results showed that this approach obtained an accuracy of 94.4% using VGG-F with a testing time of 39 ms per image, while an accuracy of 95.05% using VGG-S with a testing time of 77 ms per image. The work in [50] proposed an architecture that comprised a pre-processing stage performed by using histogram equalization, Gabor filter enhancement and ridge thinning. Subsequently, these fingerprints were fed to a CNN-based classifier. This approach, tested on a proprietary database collected using a Futronics FS88 scanner device, achieved 98.21% classification accuracy with 0.9 loss. The database contained 10 images acquired from each of 56 users: 560 samples, where 280 (56 users \times 5 images per user) samples considered for this research. Finally, Conti et al. in [51] proposed an efficient embedded fingerprint classifier node based on the fusion of a Weightless Neural Network architecture and the Virtual Neuron technique. The classifier leveraged devices with limited number of resources, allowing for resource-efficient hardware implementations (e.g., FPGA devices). Experimental results, based on a 10-fold cross-validation strategy, showed an average classification rate of 90.08% using the official FVC2002DB2 database.

In our work, the most used CNNs were investigated to extensively compare the results existing in the literature with respect to ones calculated in this work, and pro-

vide some guidelines to choose the best CNN for classification in terms of precision and execution time.

4. Materials and Methods

This section describes the details of the fingerprint databases and the CNN architectures considered. Moreover, the development environment, the CNN training and the implemented tool are provided.

The analysed fingerprint images belong to two distinct public databases: PolyU [10] and NIST [11]. By means of transfer learning, the three investigated CNNs architectures—AlexNet, GoogLeNet and ResNet—were trained to classify fingerprints into four, five, and eight classes. Subsequently, in order to make the tool usable and easy to use, a GUI was implemented. Using this interface, it is possible to carry out an accurate analysis of the classification results obtained following the training, as well as a comparison of the CNN performance, in combination with the different databases.

4.1. Fingerprint Databases

The CNN training was conducted using two databases of fingerprints, publicly available for research and non-commercial purposes, for a total data set of 7800 images, divided as follows:

- *PolyU* [10]: fingerprint database acquired from September 2014 to February 2016 at “The Hong Kong Polytechnic University”. The fingerprints come from 300 different individuals; contains 1800 images in JPG format, in 8-bit grayscale and with an image size of 320×240 pixels. The images have a very high quality, as they are acquired with an innovative technique that does not require finger contact with the sensor.
- *NIST* [11]: the NIST database containing 2000 pairs of 512×512 8-bit grayscale fingerprint images. The fingerprints were classified into five classes, in fact each image is associated with a text file that provides the corresponding class (along with other information, such as the gender and date of acquisition). The database consists of 400 pairs of fingerprints for each class, stored in PNG format.

Table 1 shows the characteristics of the PolyU, and NIST fingerprint databases.

Table 1. Characteristics of the PolyU, and NIST fingerprint databases.

Database	Fingerprint Images	People	Image Size (Pixels)	Image Format
PolyU	1800	300	320×240	BMP (8 bit)
NIST	4000	N.A.	512×512	PNG (8 bit)

4.2. Investigated Deep Learning Architectures

CNNs are well-suited for the recognition of images or more generally with data that have a spatial correlation. To optimize the performance of a specific CNN architecture in a specific application scenario, we need to effectively train it. In particular, an approach called transfer learning was used in our work, which consists in fine-tuning the deepest CNN layers on a new data set. Therefore, starting from an existing and pre-trained network, new data are fed, containing previously unknown classes. Once the network is in place, a new task can be carried out, such as fingerprint classification in our case.

4.2.1. AlexNet

AlexNet is a CNN proposed by Krizhevsky et al. [12] and ranked second in the 2012 contest on image recognition of the ImageNet database. Compared to the LeNet to which it refers, proposed a few years earlier by Yann LeCun [52], AlexNet presents a deeper neural structure, composed of a higher number of pooling and convolution layers. AlexNet uses Rectified Linear Units (ReLUs) as activation functions.

4.2.2. GoogLeNet

GoogLeNet is a CNN [13], released by Google Inc. in 2014, the year in which it participated in the ImageNet Large Scale Visual Recognition Competition (ILSVRC) [53], reaching the first place with an error rate of 6.66%, which is close to the human error rate.

The network is composed of an architecture inspired by LeNet from which it draws an important element: the inception-module. Inception is a technique originally designed by Yann LeCun and consists of the use of multi-dimensional convolutions that are grouped into a layer. This introduces an improvement in the results, due to the fact that hierarchical information, with variable dimensions, can be identified at the same level.

Unlike other networks that apply momentum to accelerate the stochastic descent of the gradient, GoogLeNet uses Root Mean Square Propagation (RMSProp), a method with an adaptive learning rate.

GoogLeNet is based on very small convolutions to reduce the number of parameters. In fact, it has 22 layers, and a reduced number of parameters (four million versus 60 million of AlexNet).

4.2.3. ResNet

The Residual Network (ResNet) was released in 2015 by Microsoft Research, employing 152 layers [14]. GoogLeNet and AlexNet follow the archetypal scheme of classic CNNs: a series of convolutions, pooling, activation layer and finally fully-connected-layers. ResNet has a distinctly different architecture: the developers investigated why by increasing the number of layers, CNNs do not improve the performance. This problem has always been motivated by the hypothesis that adjacent feature-maps lead to unsatisfactory results, for the reasons discussed in the previous paragraph.

To address this problem, ResNet uses a global average-pooling (i.e., average calculated on the entire feature-maps), applied to the result of the last convolution, followed by a single fully-connected-layer. In practice, the difference between the feature maps of the layers is learned, or the “residual” and subsequently, correct the input based on this.

Table 2 summarizes the most relevant characteristics of AlexNet, GoogLeNet, and ResNet CNN-based architectures.

Table 2. Characteristics of used CNNs architecture.

CNN	Year	Input Size	Layers
AlexNet	2012	$227 \times 227 \times 3$	25
GoogLeNet	2014	$224 \times 224 \times 3$	144
ResNet	2015	$224 \times 224 \times 3$	177

4.3. Implementation Details: Fine-Tuning

Transfer learning makes it possible to exploit a neural network previously trained on a specific data set, to solve problems associated with “similar” contexts. This happens by reprocessing only some specific layers of the model: analysing the final layers of AlexNet, it is clear that the model provides the possibility of distinguishing 1000 different categories; if we want to classify the fingerprints in a number of classes not exceeding eight, it is clear that by making the appropriate changes to the network, the impact on computational costs is drastically reduced.

The benefits that can be drawn from this type of approach reside therefore in an effective reduction in the resources used. For example, during the training of GoogLeNet, a database of over one million images was used; as a result, the use of high-performance processors and graphics cards become essential. Through transfer learning, on the other hand, the network can be retrained to carry out new classification tasks, by requiring a much smaller number of images, as well as a computational platform with more limited performance.

4.3.1. Fingerprint Image Pre-Processing

In order to maximize the classification performance, some data pre-processing steps are needed. The fingerprint images were resized considering the input size of each CNNs (Table 2) and converted from grayscale to RGB (the pixel matrix is cloned times). Furthermore, the images underwent a data-augmentation procedure, using a scale factor between 0.9 and 1.1 (on both the x and y axes), in order to have greater variability of the input data.

4.3.2. CNN Architecture Adaptations

To enable result repeatability, we decided to keep the pre-existing CNN architectures unchanged for the training phase. By acting on the layers of the CNNs, it is possible to adapt the network to the type of classification at hand and reduce processing times accordingly. This is possible by freezing-layers, through which the weights of the frozen layers are not modified during training (basically, the learning rate on these layers is zero). This means that the gradients of the frozen layers are not calculated, thus significantly accelerating the network training process.

From a preliminary analysis it was obtained that by not freezing any layers and freezing all the layers (except the input-output ones), the performance remained overall unaffected, but with much longer training times in the case of complete training (i.e., without freezing some layers). For this reason, we chose to apply only a few small changes to the architectures to properly configure the input and output layers, by modifying and adapting them (i) to the input image size and (ii) to the output classes number, respectively, (four, five, or eight). This allowed us to keep the classification performance consistent by also reducing the training times. Table 3 provides the details of the layers that were modified.

Table 3. CNNs architecture adaptation details.

CNN	Original Layers	Adapted Layers
AlexNet (25 layers) <i>no layer is frozen</i>	#23: Fully Connected 1000 fully connected layer #25: Classification Output crossentropyex with "tench" and 999 other classes	#23: Fully Connected 4 fully connected layer #25: Classification Output crossentropyex
GoogLeNet (144 layers) <i>layers 1–10 are frozen</i>	#142 "loss3-classifier" Fully Connected 1000 fully connected layer #143 "prob" Softmax #144 "output" Classification Output crossentropyex with "tench" and 999 other classes	#142 "fc" Fully Connected 4 fully connected layer #143 "softmax" Softmax #144 "classoutput" Classification Output crossentropyex
ResNet (177 layers) <i>layers 1–110 are frozen</i>	#175 "fc1000" Fully Connected 1000 fully connected layer #176 "fc1000_softmax" Softmax #177 "ClassificationLayer_fc1000" Classification Output crossentropyex with "tench" and 999 other classes	#175 "fc" Fully Connected 4 fully connected layer #176 "softmax" Softmax #177 "classoutput" Classification Output crossentropyex

Each fingerprint database was randomly partitioned to obtain a (i) training set and a (ii) test set: in particular, 70% was used for the training of the CNNs, the remaining 30% was used for the performance test phase of the network. Table 4 shows the relevant parameters used during CNNs training phase.

Table 4. CNNs architecture training details.

Learning Parameter	AlexNet	GoogLeNet	ResNet
Mini Batch Size	64	36	64
Max Epochs	26	20	16
Initial Learn Rate	0.001	0.001	0.001
Validation Frequency	3	3	3

4.3.3. Development Environment

The development environment used to train and test the CNNs was MATLAB R2019a (The MathWorks, Natick, MA, USA), installed on a Linux Ubuntu 18.10 platform, equipped with an Intel Core i7-980 CPU@3.2 GHz, 16 GB of RAM and a solid-state drive.

MATLAB allows us to use the Graphics Processing Unit (GPU) to support processing. For this reason, an NVIDIA GeForce GT710 GPU with 2GB of DDR5 RAM, which used the CUDA 3.0 computing technology, was leveraged. The use of the GPU allowed us to reduce the training time. Moreover, we aimed at drawing out the MATLAB potential for Deep Learning by also leveraging the Graphical User Interface Development Environment (GUIDE). From this practical point of view, the use of MATLAB can facilitate the implementation of user-friendly tools, providing also a GUI, but at the same time allowing developers to deal with the classifier models based on the underlying Deep Learning architectures.

4.4. Experimental Setup

Before training the three CNN architectures, for each database, it was necessary to classify the fingerprints into four, five, and eight classes. The prepared databases were randomly split into two partitions: (i) training set (70% of the whole database); (ii) test set (30% of the whole database). Of course, depending on the specific trained/tested network, the size of the database images was adapted to meet the requirements in Table 2 (i.e., input size). These data allowed us to train each CNN to classify each database according to the three taxonomies illustrated in Section 2.3. In total, 18 different training sessions were carried out (3 CNN architectures \times 2 fingerprint database \times 3 classification tasks) corresponding to about 61 h of training (Figures 4 and 5).

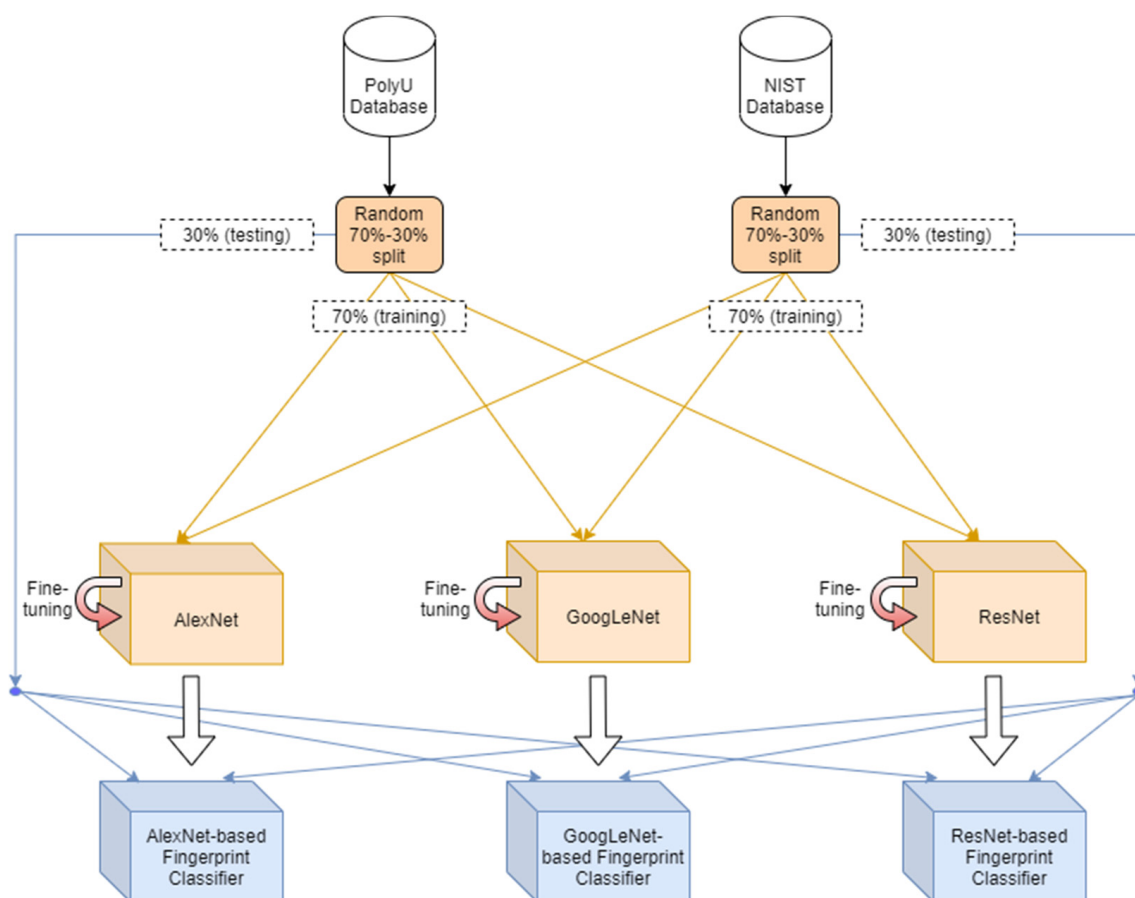


Figure 4. Flow diagram illustrating the approach used to train/test the CNNs. This training scheme was repeated three times, depending on the number of classes.

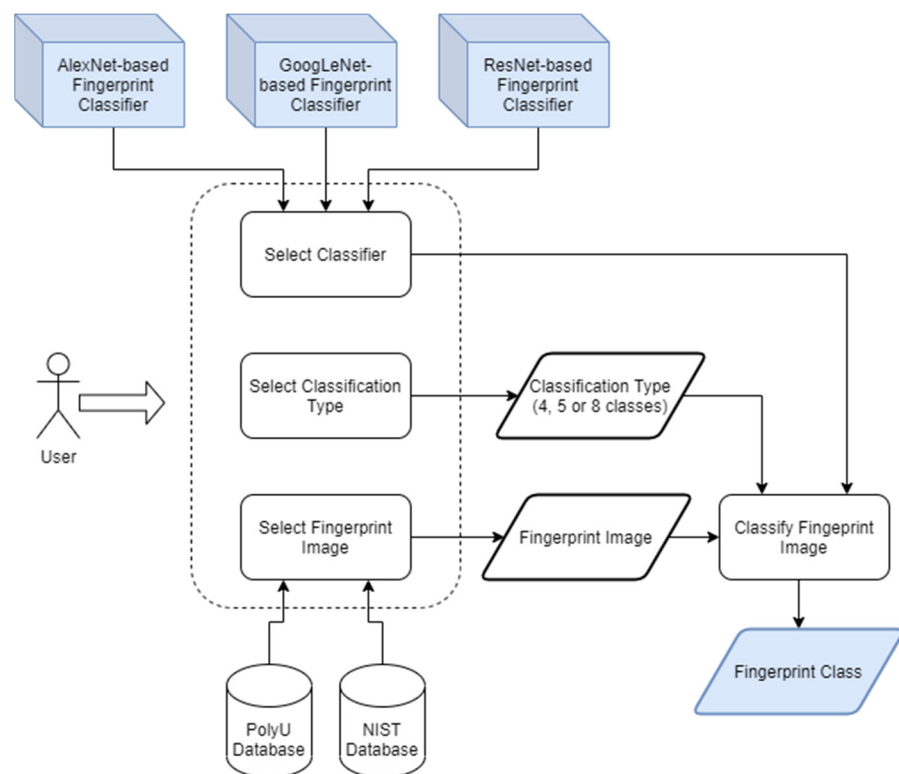


Figure 5. Flow diagram illustrating the functioning of the implemented tool and the provided processing operations. Three CNNs and two different databases can be used, while three classification types can be performed (namely, 4, 5 or 8 classes).

Considering that for a biometric system it is more important not to include false positives, *Precision* was used as a metric to assess the performance of the classification, obtained by dividing the number of fingerprints correctly classified as class *C*, by the number of all the fingerprints classified as class *C*. Let TP and FP be the True Positives and the False Positives, respectively, the *Precision* can be defined according to Equation (1):

$$Precision = \frac{TP}{TP + FP}. \quad (1)$$

Finally, to confirm and corroborate the comparisons among the investigated CNNs and databases, the McNemar test on paired authentication result was performed [54]. This non-parametric test on paired nominal data assesses whether the precision achieved by two classification models is statistically different. This test uses the following null hypothesis: the predicted class labels from the two compared models have equal precision for predicting the ground-truth class labels. Since this statistical analysis involves multiple comparison tests, we adjusted the *p*-values using the Bonferroni–Holm method [55]. In all the tests, a significance level of $\alpha = 0.05$ was used.

4.5. Fingerprint Classification Tool

The development and implementation work led to a user-friendly application, equipped with a GUI, which allows us to classify the fingerprints by using the three CNNs. This software application facilitated various operations for the network performance analysis, comparison of the results based on the database used and evaluations of the performance of the CNNs on individual fingerprints. In particular, the implementation of the GUI allows for easily using the classifiers implemented and analyzing the results obtained in the classification for each of the 18 combinations that can be obtained by selecting the CNN

architecture, the fingerprint database and the number of classes. The GUI is displayed in Figure 6. It offers three panels on which it is possible to carry out the planned activities.

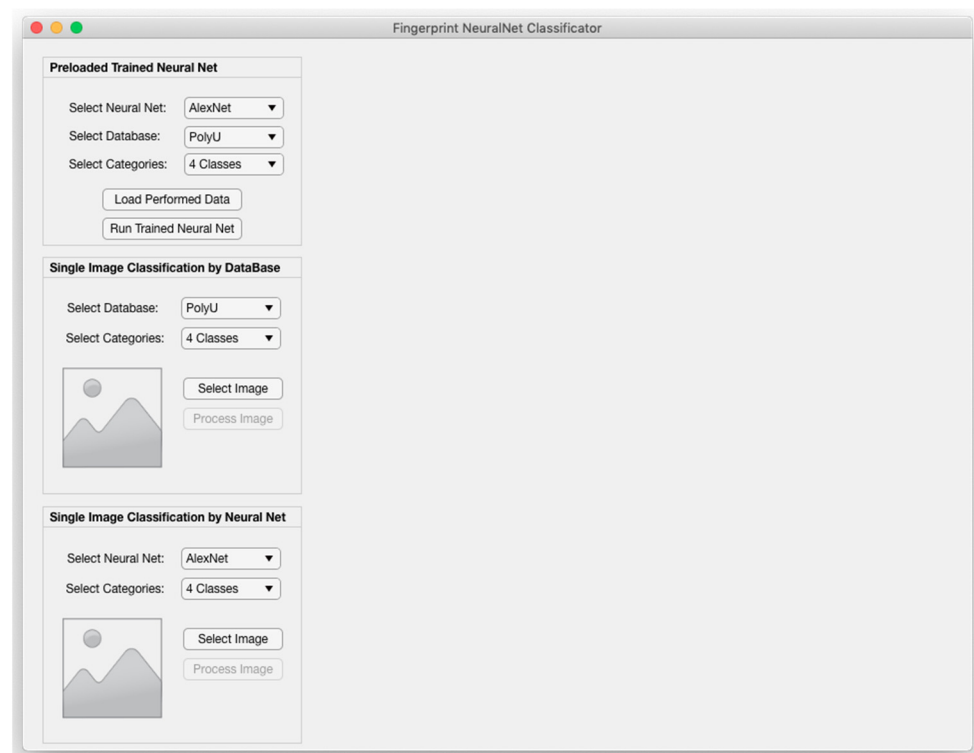


Figure 6. GUI of the implemented fingerprint classification tool.

On the left side of the GUI there are three panels, each of which groups a specific set of functionalities: (i) *Preloaded Trained Neural Net*, (ii) *Single Image Classification by Database*, and (iii) *Single Image Classification by Neural Net*.

The first panel “Preloaded Neural Net” refers to the training results that are stored at the end of each training. The GUI enables a complete outline of the performance for each CNN (for each of the three fingerprint databases and for the three different classification types). The tool generates a confusion matrix that returns a representation of the statistical classification performance (Figure 7).

The confusion matrix allows us to assess the performance of the network and how the network responds. The abscissa axis (“Actual Classes”) indicates the real class of the fingerprint, while the ordinate axis (“Predicted Classes”) denotes the classifications obtained from the CNNs. The elements that have been correctly classified are placed in the main diagonal of the matrix and can be identified in the green cells. For each specific class, each cell displays the number of correctly classified images.

The “*Single Image Classification by Database*” and “*Single Image Classification by Neural Net*” panels allow us to classify an image, leaving the user the right to choose the database, the CNN, and the classification type (Figure 8). In the first case (database selection), the results are: (i) the performance that each neural network obtained for the loaded image, (ii) the neural network that obtained the highest accuracy value, (iii) the type of classification selected, and (iv) the class of membership along with the chosen database. In the second case (selection of the neural network and the type of classification), the application autonomously chooses the best CNN and shows: (i) the performance that have been achieved in the three databases, (ii) the database that has obtained a higher performance, and (iii) the class assigned to the fingerprint.

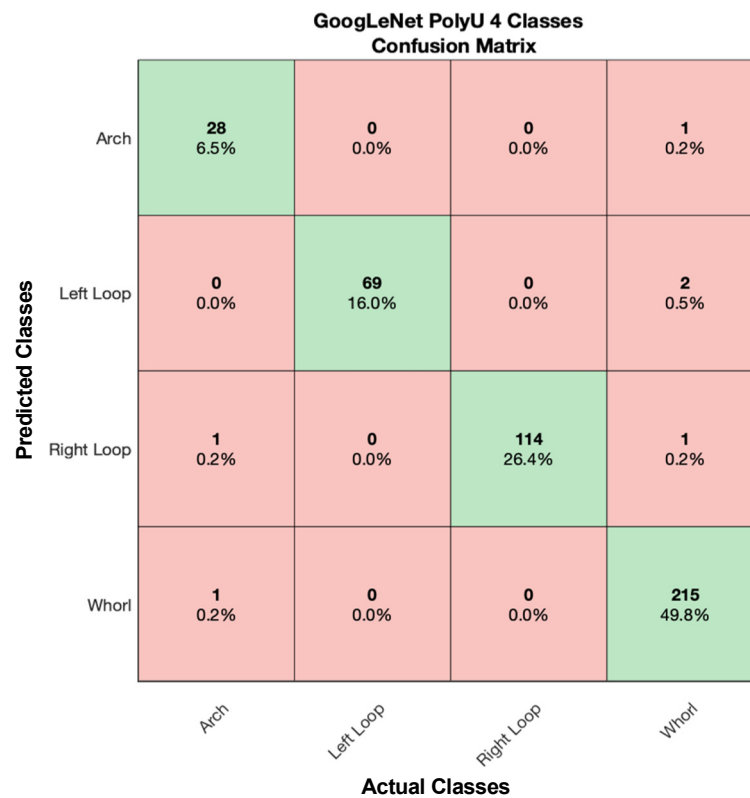


Figure 7. Example of a Confusion Matrix yielded by the implemented GUI-powered tool.

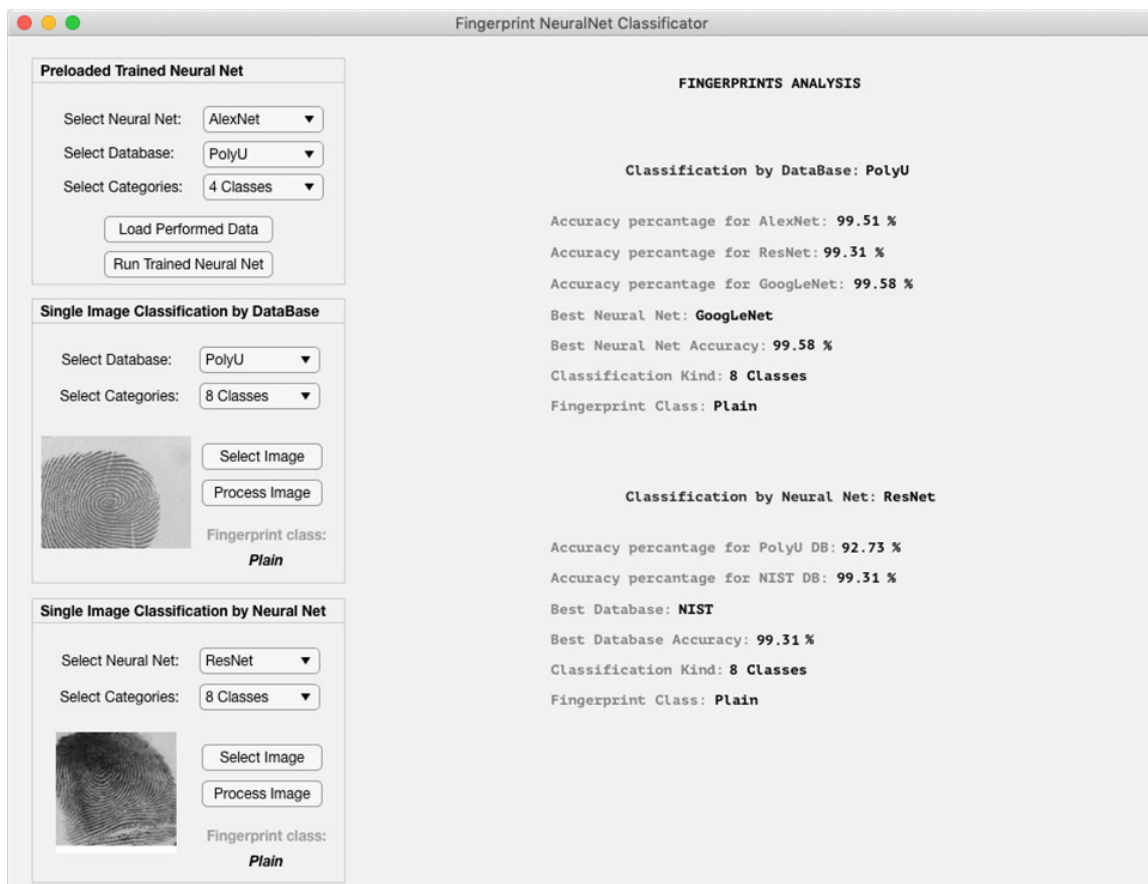


Figure 8. Single fingerprint classification example.

5. Experimental Results

This section provides a detailed comparison of the results obtained and corroborated by the statistical analysis based on McNemar tests. The performance of each CNN classifier is shown as the database and the number of classes vary.

5.1. Classification Evaluation

Each CNN was tested on different classification types. Table 5 shows the performance on four-, five- and eight-class classification.

Table 5. Precision achieved on 4-, 5- and 8-class classification for all the experiments. Boldface denotes the best performing CNN for each experiment.

Database	Classification Type	AlexNet	GoogLeNet	ResNet
NIST	four-class	96.85	93.90	96.57
	five-class	96.05	91.55	95.37
	eight-class	93.75	92.07	92.71
PolyU	four-class	99.51	99.51	99.65
	five-class	99.79	99.79	99.51
	eight-class	99.51	99.58	99.31

In four-class classification, the highest precision was achieved by AlexNet with the NIST database, with 96.85%. Considering the PolyU database, the CNN that achieved the best results was ResNet with 99.6528%. On five-class classification, it is worth noting that, similar to the four-class classification for NIST, the best network is AlexNet with 96.0500%. Conversely, for the PolyU database, the networks with the highest precision index were AlexNet and GoogLeNet that reached 99.7917%. Finally, in eight-class classification, it is observed that AlexNet predominated on the NIST database obtaining 93.75%, while GoogLeNet obtained the best result with 99.5833% on the PolyU database.

Table 6 shows the p -values of the McNemar test on the pairwise comparisons of the classification results achieved by the investigated CNNs, considering the experiments described in Section 4.4. Although AlexNet consistently achieved the best performance on the NIST database, statistical significance over GoogLeNet and ResNet was observed only on the eight-class classification. On four-class and five-class classification, AlexNet and GoogLeNet showed comparable performance and significantly outperformed ResNet. Considering the PolyU database, no statistical difference was observed in any of the experiments performed.

Table 6. McNemar test p -values for the pairwise comparison of the classification results achieved by the investigated CNNs. A significance level of $\alpha = 0.05$ with the Bonferroni-Holm correction method for multiple comparisons was used. Boldface denotes that the null hypothesis can be rejected.

Database	Classification Type	AlexNet vs. GoogLeNet	AlexNet vs. ResNet	GoogLeNet vs. ResNet
NIST	four-class	5.512×10^{-14}	0.3659	2.254×10^{-11}
	five-class	2.545×10^{-20}	0.0672	4.123×10^{-15}
	eight-class	9.099×10^{-4}	0.0254	0.2020
PolyU	four-class	1.0	1.0	1.0
	five-class	1.0	0.3750	0.3750
	eight-class	0.9090	0.9090	0.9053

The average precision values of the networks were also determined: In Table 7 the precision data are determined by calculating the average of the precision that each network obtains in the three classifications. It has been observed that, considering a single data set composed of NIST and PolyU databases, AlexNet is the most accurate on all three classification types.

Table 7. Average precision of each CNN for each classification type (regardless of the database).

Classification Type	AlexNet	GoogLeNet	ResNet
four-class	98.18	96.71	98.11
five-class	97.92	95.67	97.44
eight-class	96.63	95.83	96.00

Table 8 reports the average precision values obtained by the CNNs on each database, averaged on all three classification types. Analysing the accuracies, it is worth to note that AlexNet and GoogLeNet both reach the higher precision in NIST and PolyU, respectively.

Table 8. Average precision of each CNN for each specific database (regardless of the classification type).

Database	AlexNet	GoogLeNet	ResNet
NIST	95.55	92.51	94.88
PolyU	99.61	99.63	99.49

The variations of the performance achieved by the three investigated architectures were evaluated as the classification type and database varied.

In particular, we evaluate the loss of average precision on both databases, when switching from one classification type to another. Going from a four-class to a five-class classification, the network that loses the least percentage points is ResNet, with a loss of 0.26%. Taking into consideration the variation in precision between from five-class to eight-class classification, it is always ResNet that shows a lower difference which is equal to 1.29%. The transition from four-class to eight-class classification introduces a loss factor of 0.88% with GoogleNet, which represents the lowest of the three networks.

Similarly, we can evaluate the loss of average precision on all three types of classification, as we move from one database to another. Considering the transition from the PolyU database to the NIST database, it is ResNet that prevails with a lower loss factor, corresponding to 4.06% followed by AlexNet with 4.61%.

5.2. Training Times

For the comparison of the performance of the CNNs, training times were evaluated.

AlexNet has a simplified architecture that results in shorter training times. GoogLeNet and ResNet having a higher number of levels, require more time to process the data; to give consistency to processing times and performance, some levels were “freezing” (as discussed in Section 4.3) and the training options were tuned.

Table 9 reports the training times necessary for each neural network, associated with each database, making classifications in four-, five- and eight-class, respectively. As expected, AlexNet is the network that requires the lowest training times in any type of classification. GoogLeNet and ResNet achieve different results which vary according to the type of classification and the database: in the case of the classification into four-class, ResNet is always preceded by GoogLeNet, regardless of the database being processed. This also happens by classifying CASIA fingerprints into five- and eight-class. For the fingerprints belonging to the PolyU and NIST, classified into five- and eight-class, GoogLeNet that follows ResNet during training times.

Table 9. Training times on 4-, 5- and 8-class classification.

Database	Classification Type	AlexNet	GoogLeNet	ResNet
NIST	four-class	02:17:50	07:57:01	07:58:11
	five-class	02:16:17	07:56:23	07:54:52
	eight-class	02:16:06	07:57:59	07:52:04
PolyU	four-class	00:29:16	01:23:25	01:23:42
	five-class	00:29:01	01:23:26	01:23:09
	eight-class	00:33:56	01:23:24	01:03:10

6. Discussion and Conclusions

The aim of this work was to extensively compare the performance of different CNN architectures for the fingerprint classification problem and provide indications regarding the most suitable network to classify the fingerprints coming from different databases and according to different well-established taxonomies (i.e., four-, five- and eight-class classification).

Considering the average performance in the different classification types (Tables 7 and 8), our experimental findings can be summarized as follows:

- AlexNet and ResNet achieved equivalent performance in four- and five-class classifications on the NIST database, by significantly outperforming GoogLeNet;
- AlexNet performed significantly the best in the eight-class classification;
- all the investigated CNN-based architectures achieved comparable performance in the case of high-quality images (such as the PolyU database).

From a computational perspective, AlexNet requires the lowest amount of training time, with a $\sim 3\times$ reduction of training times compared to GoogleNet and ResNet. This aspect is important if we consider that not always the latest generation GPUs are available to train the CNN architectures.

Considering the p -values obtained by the McNemar test (Table 6) for the pairwise comparison of the classification results achieved by the investigated CNNs, we can argue that AlexNet is overall the best performing network.

In [49], the performance of two versions of VGG Net was analysed in the classification into four classes of the fingerprints of the NIST database, thus reaching an accuracy of 95.05% with VGG-S Net. The accuracy value obtained in this work, in the same context, is equal to 91.67%. The difference of 3.38% is mainly attributable to the different training options. In particular, in the aforementioned publication, the number of epochs was set to 140, introducing a considerable increase of the training time (30 h compared to about 5 h on average with the networks chosen for this paper).

Extending the work carried out in [49], we decided to tackle the problem of fingerprint classification, using two distinct databases (PolyU and NIST) and three different neural networks (namely, AlexNet, GoogLeNet, and ResNet). The three analyzed CNNs reached excellent performance: average precision values higher than 98% were achieved on the PolyU database (Table 9).

Analysing the obtained results, it is clear that the CNN performance is strongly influenced by the quality of the images. For the PolyU database, an innovative technique was used that does not require contact of the finger with the sensor, obtaining images with a higher resolution, therefore the results with this database are to be considered decisive in determining whether it is convenient to perform a classification into eight, five, or rather four classes.

With more details, regarding the PolyU database and AlexNet, the classification into four and eight classes (Table 5) showed an identical percentage of precision, equal to 99.61%. With the other two CNNs, the results followed almost the same trend. Considering the totality of the data set (PolyU and NIST databases), processed with AlexNet, the passage from a classification into four classes to one in five classes, introduced a decrease in performance of 0.10%, while a classification into eight classes reduced the precision by 5.5%.

Considering ResNet and GoogLeNet, the results are once again comparable to those just described, with minor variations, which are attributable to the two networks are characterized by a more complicated architecture. These architectures have been simplified, eliminating some of the least significant layers to make training and classification times as homogeneous as possible compared to AlexNet, which is composed of fewer levels.

Following these considerations, we could argue that AlexNet is the most suitable for solving eight-class fingerprint classification problems. In fact, a classification of this type does not significantly reduce the precision, compared to a classification into a smaller number of classes. In the event that a fingerprint recognition should be performed—for example, in an application that works in real-time—considering eight-class classification, rather than four or five, will introduce a higher reduction in the number of comparisons to be put into practice. Given that the extension from four classes to eight classes is mainly performed through a sub-classification of Whorl fingerprints, which occur with a frequency of about 28%, the time of recognition of the fingerprint would be considerably reduced.

For future developments, it would be interesting to train the CNNs from scratch, without using the freezing of the layers and investigate how the “simplified” transfer learning approach has affected the classification capacity of the architectures. In this way, the CNN-based classification could produce even higher performance. However, this might be possible only replying upon large-scale fingerprint classification databases. The availability of a high-performance computer, equipped with a GPU that supports at least CUDA 7.5, would allow for greater computing resources and leave the structures of the CNNs unchanged to carry out a new retraining of the networks in a short time.

In this work, we aimed at showing how off-the-shelf CNNs can perform fingerprint classification by considering a high number of experiments. Hyper-parameter tuning, by using a grid-search or metaheuristics on the various settings (e.g., filter size, strides, activation functions), would be interesting for future work [56]. For this reason, we preferred to keep the CNN hyper-parameters and settings as in the original implementations; indeed, we focused on conducting a considerable number of experiments by considering all the combinations among three CNNs, two datasets, and three distinct classification tasks. Therefore, an exhaustive hyper-parameter optimization would affect the validity across the different experimental configurations. With reference to filter size, obtaining empirical evidence involves an expensive training cost, especially for particularly deep networks [57]. Due to the fine-grained details of fingerprints (e.g., minutiae [58]), we did not consider strided convolutions to preserve the information conveyed by the original input images. Finally, it could be interesting to use other fingerprint databases to further assess the classification and generalization capabilities of CNN architectures.

As a further improvement of this study, it could consider other data-augmentation techniques using also other geometric transformations (e.g., cropping, rotation, stretching) and histogram-based operations. Moreover, Generative Adversarial Network (GAN)-based data augmentation might be investigated, in terms of both image enhancement [59] and geometrical transformations [60], for classification and detection tasks [61]. The preparation of adversarial examples—by applying small modifications to the original images that are close to the decision boundaries learned by a classifier [62]—might affect Deep Learning based fingerprint recognition systems [63] and authentication [64].

Author Contributions: Conceptualization, C.M., L.R., S.V. and V.C.; methodology, C.M.; software, C.M. and S.V.; validation, C.M. and L.R.; formal analysis, C.M., L.R. and S.V.; investigation, C.M., L.R. and S.V.; resources, S.V. and V.C.; data curation, C.M. and S.V.; writing—original draft preparation, C.M. and L.R.; writing—review and editing, S.V. and V.C.; visualization, C.M. and L.R.; supervision, S.V. and V.C. 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: The analysed datasets are publicly available. Related references are reported in the References section.

Acknowledgments: Authors would like to thank Davide Viscuso for his initial contribution to this research.

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

References

1. Conti, V.; Vitello, G.; Sorbello, F.; Vitabile, S. An Advanced Technique for User Identification Using Partial Fingerprint. In Proceedings of the 2013 Seventh International Conference on Complex, Intelligent, and Software Intensive Systems, Taichung, Taiwan, 3–5 July 2013; pp. 236–242.
2. Adler, A.; Schuckers, S. Biometric vulnerabilities, overview. In *Encyclopedia of Biometrics*; Li, S.Z., Jain, A., Eds.; Springer: Boston, MA, USA, 2009. [CrossRef]
3. Nguyen, H.T. Fingerprints Classification through Image Analysis and Machine Learning Method. *Algorithms* **2019**, *12*, 241. [CrossRef]
4. Peralta, D.; Triguero, I.; García, S.; Saeys, Y.; Benitez, J.M.; Herrera, F. On the use of convolutional neural networks for robust classification of multiple fingerprint captures. *Int. J. Intell. Syst.* **2018**, *33*, 213–230. [CrossRef]
5. Sundararajan, K.; Woodard, D.L. Deep Learning for Biometrics: A Survey. *ACM Comput. Surv.* **2018**, *51*, 1–34. [CrossRef]
6. Militello, C.; Rundo, L.; Minafra, L.; Cammarata, F.P.; Calvaruso, M.; Conti, V.; Russo, G. MF2C3: Multi-feature Fuzzy clustering to enhance cell colony detection in automated clonogenic assay evaluation. *Symmetry* **2020**, *12*, 773. [CrossRef]
7. Shadrina, E.; Vol’pert, Y. Functional asymmetry and fingerprint features of left-handed and right-handed young Yakuts (Mongoloid race, north-eastern Siberia). *Symmetry* **2018**, *10*, 728. [CrossRef]
8. Nilsson, K.; Bigun, J. Using linear symmetry features as a pre-processing step for fingerprint images. In *Lecture Notes in Computer Science*; Springer: Berlin, Germany, 2001; pp. 247–252. ISBN 9783540422167.
9. Le, T.H.; Van, H.T. Fingerprint reference point detection for image retrieval based on symmetry and variation. *Pattern Recognit.* **2012**, *45*, 3360–3372. [CrossRef]
10. The Hong Kong Polytechnic University—3D Fingerprint Images Database Version 2.0. Available online: <http://www4.comp.polyu.edu.hk/~{csajaykr}/fingerprint.htm> (accessed on 22 January 2021).
11. NIST Biometric Special Databases and Software. Available online: <https://www.nist.gov/itl/iad/image-group/resources/biometric-special-databases-and-software> (accessed on 22 January 2021).
12. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
13. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
14. 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, Seattle, WA, USA, 21–23 June 2016; pp. 770–778.
15. Jian, W.; Zhou, Y.; Liu, H. Lightweight convolutional neural network based on singularity ROI for fingerprint classification. *IEEE Access* **2020**, *8*, 54554–54563. [CrossRef]
16. Grzybowski, A.; Pietrzak, K. Jan Evangelista Purkyně (1787–1869): First to describe fingerprints. *Clin. Dermatol.* **2015**, *33*, 117–121. [CrossRef]
17. Galton, F. *Finger Prints*; Macmillan and Co.: London, UK, 1892.
18. Galton, F. Decipherment of Blurred Finger Prints. In *Finger Prints*; Macmillan and Co.: London, UK, 1893.
19. International Biometric Group—The Henry Classification System. Available online: <http://static.ibgweb.com/HenryFingerprintClassification.pdf> (accessed on 22 January 2021).
20. *The Science of Fingerprints: Classification and Uses*; United States Federal Bureau of Investigation: Washington, DC, USA, 1985.
21. Zabala-Blanco, D.; Mora, M.; Barrientos, R.J.; Hernández-García, R.; Naranjo-Torres, J. Fingerprint Classification through Standard and Weighted Extreme Learning Machines. *Appl. Sci.* **2020**, *10*, 4125. [CrossRef]
22. Bigun, J. Fingerprint Features. In *Encyclopedia of Biometrics*; Li, S.Z., Jain, A., Eds.; Springer: Boston, MA, USA, 2009; pp. 465–473. ISBN 9780387730035.
23. Wegstein, J.H. *An Automated Fingerprint Identification System*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 1982.
24. Vitabile, S.; Conti, V.; Lentini, G.; Sorbello, F. An Intelligent Sensor for Fingerprint Recognition. In *International Conference on Embedded and Ubiquitous Computing*; Springer: Berlin, Germany, 2005; Volume 3824, pp. 27–36.
25. Militello, C.; Conti, V.; Vitabile, S.; Sorbello, F. A novel embedded fingerprints authentication system based on singularity points. In Proceedings of the CISIS 2008: 2nd International Conference on Complex, Intelligent and Software Intensive Systems, Barcelona, Spain, 4–7 March 2008; pp. 72–78.
26. Win, K.N.; Li, K.; Chen, J.; Fournier-Viger, F.; Li, K. Fingerprint classification and identification algorithms for criminal investigation: A survey. *Future Gener. Comput. Syst.* **2020**, *110*, 758–771. [CrossRef]

27. Hu, J.; Xie, M. Fingerprint classification based on genetic programming. In Proceedings of the 2010 2nd International Conference on Computer Engineering and Technology, Chengdu, China, 16–18 April 2010; Volume 6, pp. 193–196.
28. Rundo, L.; Tangherloni, A.; Cazzaniga, P.; Nobile, M.S.; Russo, G.; Gilardi, M.C.; Vitabile, S.; Mauri, G.; Besozzi, D.; Militello, C. A novel framework for MR image segmentation and quantification by using MedGA. *Comput. Methods Programs Biomed.* **2019**, *176*, 159–172. [[CrossRef](#)] [[PubMed](#)]
29. Alias, N.A.; Radzi, N.H.M. Fingerprint Classification using Support Vector Machine. In Proceedings of the 5th ICT International Student Project Conference, Bangkok, Thailand, 27–28 May 2016.
30. Saini, M.K.; Saini, J.S.; Sharma, S. Moment based Wavelet Filter Design for Fingerprint Classification. In Proceedings of the 2013 International Conference on Signal Processing and Communication, Noida, India, 12–14 December 2013; pp. 267–270.
31. Li, X.; Wang, G.; Lu, X. Neural Network Based Automatic Fingerprints Classification Algorithm. In Proceedings of the 2010 International Conference of Information Science and Management Engineering, Xi'an, China, 7–8 August 2010; Volume 1, pp. 94–96.
32. Turkey, A.M.; Ahmad, M.S. The use of SOM for fingerprint classification. In Proceedings of the 2010 International Conference on Information Retrieval Knowledge Management, Shah Alam, Malaysia, 17–18 March 2010; pp. 287–290.
33. Saeed, F.; Hussain, M.; Aboalsamh, H.A. Classification of Live Scanned Fingerprints using Dense SIFT based Ridge Orientation Features. In Proceedings of the 2018 1st International Conference on Computer Applications Information Security, Riyadh, Saudi Arabia, 4–5 April 2018; pp. 1–4.
34. Wang, X.; Wang, F.; Fan, J.; Wang, J. Fingerprint classification based on continuous orientation field and singular points. In Proceedings of the 2009 IEEE International Conference on Intelligent Computing and Intelligent Systems, Shanghai, China, 20–22 November 2009; Volume 4, pp. 189–193.
35. Biometric Systems Lab—FVC2002: Fingerprint Verification Competition. Available online: <http://bias.csr.unibo.it/fvc2002/> (accessed on 22 January 2021).
36. Biometric Systems Lab—FVC2004: Fingerprint Verification Competition. Available online: <http://bias.csr.unibo.it/fvc2004/> (accessed on 22 January 2021).
37. Neurotechnology Company—Sample Fingerprint Databases. Available online: <http://www.neurotechnologija.com/download.html> (accessed on 22 January 2021).
38. Conti, V.; Militello, C.; Vitabile, S.; Sorbello, F. Introducing Pseudo-Singularity Points for Efficient Fingerprints Classification and Recognition. In Proceedings of the 2010 International Conference on Complex, Intelligent and Software Intensive Systems, Krakow, Poland, 15–18 February 2010; pp. 368–375.
39. Biometric Systems Lab—FVC2000: Fingerprint Verification Competition. Available online: <http://bias.csr.unibo.it/fvc2000/> (accessed on 22 January 2021).
40. Wang, W.; Li, J.; Chen, W. Fingerprint Classification Using Improved Directional Field and Fuzzy Wavelet Neural Network. In Proceedings of the 2006 6th World Congress on Intelligent Control and Automation, Dalian, China, 21–23 June 2006; Volume 2, pp. 9961–9964.
41. Nain, N.; Bhadviya, B.; Gautam, B.; Kumar, D.; Deepak, B.M. A Fast Fingerprint Classification Algorithm by Tracing Ridge-Flow Patterns. In Proceedings of the 2008 IEEE International Conference on Signal Image Technology and Internet Based Systems, Bali, Indonesia, 30 November–30 December 2008; pp. 235–238.
42. 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](#)]
43. Saeed, F.; Hussain, M.; Aboalsamh, H.A. Classification of Live Scanned Fingerprints using Histogram of Gradient Descriptor. In Proceedings of the 2018 21st Saudi Computer Society National Computer Conference, Riyadh, Saudi Arabia, 25–26 April 2018; pp. 1–5.
44. Özbayoğlu, A.M. Unsupervised Fingerprint Classification with Directional Flow Filtering. In Proceedings of the 2019 1st International Informatics and Software Engineering Conference, Ankara, Turkey, 6–7 November 2019; pp. 1–4.
45. Wang, R.; Han, C.; Guo, T. A novel fingerprint classification method based on deep learning. In Proceedings of the 2016 23rd International Conference on Pattern Recognition, Cancun, Mexico, 4–8 December 2016; pp. 931–936.
46. 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; pp. 27–32.
47. Hamdi, D.E.; Elouedi, I.; Fathallah, A.; Nguyuen, M.K.; Hamouda, A. Combining Fingerprints and their Radon Transform as Input to Deep Learning for a Fingerprint Classification Task. In Proceedings of the 2018 15th International Conference on Control, Automation, Robotics and Vision, Singapore, 18–21 November 2018; pp. 1448–1453.
48. Ge, S.; Bai, C.; Liu, Y.; Liu, Y.; Zhao, T. Deep and discriminative feature learning for fingerprint classification. In Proceedings of the 2017 3rd IEEE International Conference on Computer and Communications, Chengdu, China, 13–16 December 2017; pp. 1942–1946.
49. Michelsanti, D.; Ene, A.-D.; Guichi, Y.; Stef, R.; Nasrollahi, K.; Moeslund, T.B. *Fast Fingerprint Classification with Deep Neural Networks*; Scitepress: Setúbal, Portugal, 2017; pp. 202–209.
50. 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; pp. 86–91.

51. Conti, V.; Rundo, L.; Militello, C.; Mauri, G.; Vitabile, S. Resource-Efficient Hardware Implementation of a Neural-based Node for Automatic Fingerprint Classification. *J. Wirel. Mob. Netw. Ubiquitous Comput. Dependable Appl. ISYOU* **2017**, *8*, 19–36.
52. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural. Comput.* **1989**, *1*, 541–551. [[CrossRef](#)]
53. 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](#)]
54. McNemar, Q. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika* **1947**, *12*, 153–157. [[CrossRef](#)]
55. Westfall, P.H.; Troendle, J.F.; Pennello, G. Multiple McNemar tests. *Biometrics* **2010**, *66*, 1185–1191. [[CrossRef](#)]
56. Yamasaki, T.; Honma, T.; Aizawa, K. Efficient optimization of convolutional neural networks using particle swarm optimization. In Proceedings of the IEEE Third International Conference on Multimedia Big Data, Laguna Hills, CA, USA, 19–21 April 2017; pp. 70–73. [[CrossRef](#)]
57. Han, S.; Meng, Z.; Li, Z.; O'Reilly, J.; Cai, J.; Wang, X.; Tong, Y. Optimizing filter size in convolutional neural networks for facial action unit recognition. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5070–5078. [[CrossRef](#)]
58. Tang, Y.; Gao, F.; Feng, J. Latent fingerprint minutia extraction using fully convolutional network. In Proceedings of the 2017 IEEE International Joint Conference on Biometrics, Denver, CO, USA, 1–4 October 2017; pp. 117–123. [[CrossRef](#)]
59. Huang, X.; Qian, P.; Liu, M. Latent fingerprint image enhancement based on progressive generative adversarial network. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 3481–3489. [[CrossRef](#)]
60. Xu, H.; Li, C.; Rahaman, M.M.; Yao, Y.; Li, Z.; Zhang, J.; Kulwa, F.; Zhao, X.; Qi, S.; Teng, Y. An enhanced framework of generative adversarial networks (EF-GANs) for environmental microorganism image augmentation with limited rotation-invariant training data. *IEEE Access* **2020**, *8*, 187455–187469. [[CrossRef](#)]
61. Han, C.; Rundo, L.; Araki, R.; Nagano, Y.; Furukawa, Y.; Mauri, G.; Nakayama, H.; Hayashi, H. Combining noise-to-image and image-to-image GANs: Brain MR image augmentation for tumor detection. *IEEE Access* **2019**, *7*, 156966–156977. [[CrossRef](#)]
62. Goodfellow, I.; McDaniel, P.; Papernot, N. Making machine learning robust against adversarial inputs. *Commun. ACM* **2018**, *61*, 56–66. [[CrossRef](#)]
63. Biggio, B.; Fumera, G.; Russu, P.; Didaci, L.; Roli, F. Adversarial biometric recognition: A review on biometric system security from the adversarial machine-learning perspective. *IEEE Signal Process. Mag.* **2015**, *32*, 31–41. [[CrossRef](#)]
64. won Kwon, H.; Nam, J.W.; Kim, J.; Lee, Y.K. Generative adversarial attacks on fingerprint recognition systems. In Proceedings of the 2021 International Conference on Information Networking, Jeju Island, Korea, 13–16 January 2021; pp. 483–485. [[CrossRef](#)]