applied sciences

MDPI

Article

Fingerprint Classification Based on Multilayer Extreme Learning Machines

Axel Quinteros * and David Zabala-Blanco

Department of Computer Science and Industry, Faculty of Engineering Science, Universidad Católica del Maule, Talca 3480112, Chile; dzabala@ucm.cl

* Correspondence: aquinteros@ucm.cl; Tel.: +56-953396386

Abstract: Fingerprint recognition is one of the most effective and widely adopted methods for person identification. However, the computational time required for the querying of large databases is excessive. To address this, preprocessing steps such as classification are necessary to speed up the response time to a query. Fingerprints are typically categorized into five classes, though this classification is unbalanced. While advanced classification algorithms, including support vector machines (SVMs), multilayer perceptrons (MLPs), and convolutional neural networks (CNNs), have demonstrated near-perfect accuracy (approaching 100%), their high training times limit their widespread applicability across institutions. In this study, we introduce, for the first time, the use of a multilayer extreme learning machine (M-ELM) for fingerprint classification, aiming to improve training efficiency. A comparative analysis is conducted with CNNs and unbalanced extreme learning machines (W-ELMs), as these represent the most influential methodologies in the literature. The tests utilize a database generated by SFINGE software, which simulates realistic fingerprint distributions, with datasets comprising hundreds of thousands of samples. To optimize and simplify the M-ELM, widely recognized descriptors in the field—Capelli02, Liu10, and Hong08—are used as input features. This effectively reduces dimensionality while preserving the representativeness of the fingerprint information. A brute-force heuristic optimization approach is applied to determine the hyperparameters that maximize classification accuracy across different M-ELM configurations while avoiding excessive training times. A comparison is made with the aforementioned approaches in terms of accuracy, penetration rate, and computational cost. The results demonstrate that a two-layer hidden ELM achieves superior classification of both majority and minority fingerprint classes with remarkable computational efficiency.

Keywords: feature descriptors; fingerprint classification; identification systems; biometry; multilayer extreme learning machines

Academic Editors: Elias N. Zois and Dimitrios Kalivas

Received: 13 December 2024
Revised: 10 February 2025
Accepted: 11 February 2025
Published: 5 March 2025

Citation: Quinteros, A.; Zabala-Blanco, D. Fingerprint Classification Based on Multilayer Extreme Learning Machines. *Appl. Sci.* **2025**, *15*, 2793. https://doi.org/10.3390/app15052793

1. Introduction

The use of fingerprints as a biometric identifier for person recognition is considered robust and reliable due to their unique, unalterable, and enduring characteristics over time [1]. While other biometric techniques share similar attributes—such as facial and iris recognition for identification of individuals or palm print identification [2] for classification of age and gender—fingerprints stand out as more accessible and cost-effective, particularly regarding scanning technology [3]. Two primary applications of person recognition can be identified. The first straightforward and expedient method involves validating whether an individual is who they claim to be, a process commonly referred to as identity verification.

In this scenario, a fingerprint is compared to the corresponding record stored in a database (DB) using a one-to-one matching procedure. The second application, which is the primary focus of this study, addresses situations where an individual's identity is unknown. In such cases, the fingerprint must be matched against the entire DB, requiring a one-to-many comparison process. Recent advances in fingerprint localization have focused on improving accuracy through the use of novel similarity metrics [4].

Due to the constant growth of the human population, large fingerprint databases are continuously generated, making it increasingly complex for identification systems to provide instant and precise results [5]. To improve the response times of these identification systems, a preprocessing step—fingerprint classification—is employed to reduce the global search space. Addressing this challenge, convolutional neural networks (CNNs) achieve 100% accuracy but at the expense of excessive computational costs [6,7]. Techniques such as the use of an orientation field (OF), support vector machines (SVMs), and random forests (RFs) have been reported to yield satisfactory results. In [6], a threshold above 1 resulted in 100% accuracy, while a threshold of 0.6 achieved over 93% accuracy using the National Institute of Standards and Technology Special Database 4 (NIST-4), which contains 4000 samples. Nevertheless, this excellent performance was impacted by the high structural similarity of ridges in certain samples. As the threshold increases, classification confusion also rises [6]. Despite the high accuracy achieved by CNNs in fingerprint classification, their application in real-world scenarios is often impractical due to their high computational demands and extensive training requirements [8]. In situations where decisions must be made in real time or within strict time constraints, such as in medical service provision or border control systems, the inference time of CNNs can become a critical bottleneck. These applications require rapid processing and reliable identification without depending on specialized high-performance hardware. Consequently, alternative models that balance accuracy and computational efficiency are necessary to ensure practical implementation in large-scale fingerprint identification systems [9]. The authors of [7] evaluated SVM and RF performance using several databases, including DB-HLG, the fingerprint verification competition (FVC) datasets (2000, 2002, and 2004), and NIST-4. The SVM achieved an accuracy of $\geq$95.5% and a mean squared error (MSE) of $\leq$0.321, with computational times of 7, 9, and 5 h for each database, respectively. The RF obtained an accuracy of $\geq$96.75%; an MSE of $\leq$0.274; and computational times of 8, 12, and 18 h for the same databases. As evidenced, machine learning offers significant advantages in fingerprint classification. However, these networks require high-performance computational architectures, which often render their application impractical. To address these limitations, this work proposes the use of a multilayer extreme learning machine (M-ELM) for efficient fingerprint classification, with the primary objective of improving training times and computational costs.

The standard Extreme Learning Machine (ELM) is characterized by its rapid processing and low computational cost while achieving accuracy levels comparable to those of CNNs [10–12]. An ELM with a single hidden layer was demonstrated in [13], where a standard ELM was compared to two types of CNNs (a CaffeNet variant and the CNN proposed in [10]) for fingerprint classification. The study found that when using Hong08 [14] as a descriptor and the ELM as a classifier, an accuracy of 94% and a penetration rate of 0.0332 were achieved, compared to the 99% accuracy obtained by the CNNs.

ELM networks were proposed by Huang for the training of single-hidden-layer feedforward neural networks (SLFNs). In ELM, the hidden nodes are initialized randomly, then trained without iterative methods. The weights and hidden neuron biases are randomly assigned, while the output weights are determined using the Moore–Penrose pseudoinverse under the least squares criterion [15]. In ELM, the hidden nodes (or neurons) may or may

not be of the same size. The only parameters that need to be learned are the connections (or weights) between the hidden layer and the output layer. Consequently, ELM is formulated as a linear model. In comparison with traditional learning methods such as SVM, CNNs, etc., ELM is notably efficient and tends to achieve a globally optimal solution. Field studies have proven that even with randomly generated hidden nodes, ELM retains the universal approximation capability of SLFNs [16,17].

M-ELMs are networks formed by extreme learning machine autoencoders (ELM-AEs), proposed as a new multilayer perceptron (MLP) training scheme aimed at addressing the deficiencies of ELMs. Notably, a standard ELM does not perform well when processing natural signals such as sounds and images. In [18], imaging tests were conducted, including car detection, gesture recognition, and incremental online tracking. For car recognition, the UrbanaChampaign dataset from the University of Illinois [19,20] was used in both the training and testing phases. Images with dimensions of 100 × 40 pixels achieved an accuracy of 95.5% with a training time of 46.7 s. In the second experiment, the Cambridge gestures dataset was used [21], comprising 900 image sequences of novel hand gesture types, which are divided into three hand shapes and three movements. Each type contains 100 image sequences with dimensions of 60 × 80 × 10 pixels. In the first phase, static gestures were analyzed, while the second phase included movements. This experiment achieved an average test accuracy of 99.4%, with a training time of 57.7 s.

The M-ELM training architecture is structurally divided into two phases: a hierarchical representation of unsupervised features and a supervised feature classification phase. ELM-AEs are stacked to enable learning across multiple hidden layers (unsupervised), except for the final layer, which consists of a standard ELM (supervised) that performs classification [22]. The M-ELM model was proposed to improve generalization capacity, which directly depends on the characteristics of the training dataset. By addressing the two primary issues of the original ELM—network stability and computational complexity—the M-ELM enhances generalization capacity and simplifies computations. Specifically, the output weights are calculated using the generalized inverse of the hidden-layer output and the system's actual outputs [15]. As mentioned earlier, this study introduces, for the first time, the use of an M-ELM as a fingerprint classifier. To achieve this, we optimize the hyperparameters using brute force and determine the optimal number of neurons for each hidden layer. The results are validated using five-fold cross-validation, employing metrics such as training time, accuracy, and penetration rate [5,23]. These metrics enable a fair comparison with the most recent studies, highlighting their relevance in the literature [10,13]. The main contributions of this research are the introduction of novel alternatives for fingerprint classification, combining the best descriptors presented in the literature with the M-ELM approach. This method demonstrates a 4% improvement in performance compared to approaches using commercial computers [13], although it is accompanied by a slight decrease in the penetration rate (PR) of 0.0003. Additionally, the training time is significantly reduced by approximately 17 s compared to ELM-based approaches. These results emphasize the scalability of the proposed method, as it evaluates the impact of varying the number of hidden neurons across different classifier configurations, including one hidden layer for the ELM and two or three hidden layers.

The remainder of this article is structured as follows: Section 2 provides a review of the state of the art, summarizing the most significant contributions from the past decade that are directly relevant to this study, focusing on the two dominant trends: CNNs for image-based classification and ELMs for descriptor-based classification. Section 3 discusses the descriptors, along with the theoretical and mathematical foundations employed for feature extraction and fingerprint classification using multilayer ELMs. Section 4 describes the validation process and performance metrics applied in this study, along with details

about the database, including the quality and quantity of fingerprints. Section 5 presents the heuristic optimization process, considering different hidden layers for the multilayer ELM algorithm (one, two, and three hidden layers) using various descriptors and databases. It also includes tables summarizing the results, offering a comparison of completeness and computational cost in terms of the general and specific performance of the proposed approach relative to the most recent and high-performing studies. Finally, Section 6 concludes the manuscript by summarizing the findings and outlining potential directions for future research.

2. State of the Art

This section is divided into three main subsections—CNN and Images, ELM with Descriptors, and Classical Methodologies—with the aim of improving understanding and distinguishing the main trends in addressing the problem of classifying individuals through fingerprint analysis.

2.1. CNN and Images

In previous studies, various techniques have been proposed for fingerprint classification using neural networks, with two primary approaches dominating the field: CNNs for image-based classification and ELMs for descriptor-based classification.

Regarding CNN-based methods, the work of Peralta [10] stands out, in which a convolutional neural network with softmax functions—a modification of AlexNet—was proposed for fingerprint classification. This approach eliminates the need for an explicit feature extraction process while addressing challenges related to image noise or disturbances in fingerprint imaging. For comparison, tests were conducted using three additional classifiers: SVM, decision trees (DTs), and K-Nearest Neighbors (K-NN). For feature extraction, three descriptors were used: Capelli02 [24], Hong08 [14], and Liu10 [25]. The databases were SFinGe and NIST-4 [3]. A five-fold cross-validation (5-FCV) scheme was applied, where each database was randomly divided into five subsets with identical class distributions, enabling the calculation of average performance across all tests. KEEL software (3.0) (Knowledge Extraction based on Evolutionary Learning) was used to implement the SVM, DT, and K-NN classifiers, while the CNNs were implemented using Caffe. The experiments were conducted on a computer with an Intel Core i7-3820 processor (3.60 GHz) and 27 GB of RAM (Santa Clara, CA, USA). The CNN was run with an Nvidia GeForce GTX TITAN GPU (2688 cores, 6144 MB GDDR5 RAM) (Santa Clara, CA, USA). This study aimed to evaluate classification methods from the perspectives of robustness and efficiency, analyzing both the accuracy and computational performance of different approaches. The proposed network achieved an accuracy of 99.60% for the HQNoPert database, corresponding to only 120 errors out of 30,000 fingerprint test samples. The CNN outperformed all combinations of feature extractors and state-of-the-art classifiers, achieving a penetration rate of 29.79%, compared to 39.99% for the original CaffeNet. Furthermore, the CNN demonstrated competitive computational performance, requiring an average of 960 s, which was significantly faster than the 230 s required, on average, by the combination of feature extraction and classification.

Zia [23] sought to improve the efficiency of fingerprint classification by using deep convolutional neural networks (DCNNs) in combination with a Bayesian model to address false positives. This approach, termed Bayesian DCNN (B-DCNN), highlights the advantages of incorporating Bayesian methods over traditional DCNNs. Two fingerprint databases were used in the experimentation: NIST-4 and 1-A (FVC DB1-A) from 2002 and 2004, respectively. The classification was based on the five-class system proposed by Henry [26] (arch, left loop, right loop, tented arch, and whorl). While some researchers

exclude these fingerprints from the database, others consider these labels valid. To ensure fair experiments, this study used two versions of the NIST-4 database: the complete database, denoted as NIST-4 (2000), and a version with 350 fingerprints, having multiple deleted labels, referred to as NIST-4 (1650). The results showed accuracies of 96.1% for NIST-4 and 95.3% for the FVC 2002 and 2004 databases. The findings demonstrate that the B-DCNN approach outperforms conventional DCNN models, yielding improvements in accuracy of 0.8% and 1.0% compared to the work presented by Gal et al. [27]. The training times for the National Institute of Standards and Technology (NIST) and Fingerprint Verification Competition (FVC) databases were 4.393 and 3.801 s, respectively.

Jian [28] proposed a lightweight CNN architecture focused on the singularity region of interest (ROI). The principal dataset was the NIST-4, which contains 4000 grayscale fingerprint images. To enhance the dataset, four data augmentation techniques were applied—vertical flip, horizontal rotation, small-angle rotation, and Gaussian noise—generating an additional 16,000 images. Thus, the final experimental database comprised 20,000 fingerprint images. In the first experiment, the performance of five popular non-neural network classifiers was compared. The second experiment aimed to explore and identify an optimal lightweight neural network architecture for fingerprint classification. The third experiment compared the proposed CNN architecture with three other recently published neural network architectures. The experimental results demonstrate that the proposed architecture achieved an accuracy of 93%, significantly outperforming non-neural network classifiers, including RF, K()NN, logistic regression (LR), SVM, and radial basis function (RBF). More importantly, when compared to other recently published CNN architectures, the proposed structure achieved similar or even better performance, with a parameter scale ranging from 1 = 12 to 1 = 38, thereby accelerating both training and testing. Moreover, the proposed CNN model, with fewer neurons, exhibited better overfitting suppression and greater noise robustness.

Militello [29] presented a study on the performance of pre-trained CNNs—specifically, AlexNet, GoogLeNet, and ResNet—tested on two fingerprint databases: the Hong Kong Polytechnic University (PolyU) and NIST databases. The study also compared these results with other works in the literature to determine which classification approach yielded superior performance in terms of accuracy and efficiency. This research represents the first comprehensive analysis comparing widely used CNN architectures for fingerprint classification across four, five, and eight classes. The experimental results revealed that the tested CNN architectures achieved the best performance on the PolyU dataset, likely due to the larger size and higher quality of the sample. To validate the reliability of the findings, statistical analysis was performed using the McNemar test. Together, the databases used in the study—the PolyU and NIST datasets—comprise a total of 7800 fingerprint images. The three established CNN architectures tested in the study were AlexNet [30], GoogLeNet [31], and ResNet [32]. The primary objective of this study was to determine the approach that leads to the best performance among three CNN architectures when used with different datasets, considering the type of dataset as part of the classification task. A Graphic User Interface (GUI) was provided to display the performance of each CNN in relation to the different datasets. The performance of each network can be easily calculated by: (1) using a different dataset and classification task; (2) uploading an image from any database to classify it, selecting the neural network and the number of classes; and (3) identifying the CNN with the best performance for any given fingerprint image. These features make the GUI both a powerful analysis tool and a simple test implementation.

In [33], a new fingerprint classification method based on the Histogram of Oriented Gradients (HOG) descriptor was proposed. This method enhances the calculation of ridge patterns within the HOG descriptor, improving its ability to robustly represent a fingerprint, as the resulting orientation clearly delineates the ridges forming the fingerprint image.

An ELM with an RBF kernel is employed as the classifier. Experiments were conducted using the FVC-2004 fingerprint database. The proposed method achieved an average accuracy of 98.7%, outperforming the latest fingerprint classification methods based on CNNs [6].

2.2. ELM with Descriptors

In [13], a novel fingerprint classification approach based on feature extraction models and both basic and modified ELMs was proposed, marking the first time this approach was adopted. In general, ELMs produce good results when dealing with imbalanced data, as in this case, where fingerprint types were distributed across five imbalanced classes. The descriptors used in [10], Capelli02 [24], Hong08 [14], and Liu10 [25], are based on the most relevant visual characteristics of fingerprints. Given the imbalanced classes in fingerprint identification schemes, the ELM was optimized (standard, weighted, and decay-weighted) in terms of geometric mean, estimating its hyperparameters (regularization parameter, number of hidden neurons, and disintegration parameter). SFinGe software (v.4.1.1746) was used to generate the dataset, creating synthetic fingerprints with realistic appearances, varying quality levels (translations, rotations, and geometric deformations), and true class labels. To emulate different scenarios, three quality profiles were used to generate fingerprints: High Quality No Perturbations (HQNoPert), Default, and Varying Quality and Perturbations (VQandPert). The HQNoPert database consists of high-quality fingerprints without perturbations. The default database contains medium-quality fingerprints with slight localization and rotational disturbances. The VQAndPert database includes fingerprints of varying quality with location, rotation, and geometric disturbances. The quality of the images is the only difference among the databases. A total of 30,000 fingerprints were generated, with 10,000 for each quality level. Classic metrics such as precision and rate of penetration were calculated for comparison with superior CNN-based methods reported in the literature. Experimental results show that the weighted ELM 2 (W-ELM2) with the golden ratio in the weighted matrix generally outperforms other ELMs (with 0.95% accuracy and a 0.0332% penetration rate). The combination of the Hong08 extractor and W-ELM2 competes with CNNs in terms of fingerprint classification efficacy, but the ELM-based methods demonstrate faster training speeds in all contexts.

In [34], a fingerprint classification system utilizing the ELM algorithm was presented, employing a heuristic approach to maximize classification precision. This optimization was achieved by determining the optimal parameters—the activation mapping function and number of hidden neurons—that yield the best results. The optimal configuration attained a classification accuracy of 95.5%, using the bounded sign mapping function, the SFinGe ACSP quality database, the Hong08 [14] descriptor, and 1832 hidden nodes. The evaluation was performed using the NIST-4 and SFinGe databases under the following conditions. (1) NIST-4: This dataset comprises 4000 fingerprint images, each with dimensions of 512 × 480 pixels. Each fingerprint was scanned twice at a resolution of 500 DPI. The fingerprints are distributed naturally across five classes, but due to class similarities, 350 fingerprints are labeled with dual-class identifiers. Consistent with prior studies [3,5,10,23,35], fingerprints with dual-class labels were excluded, resulting in a final dataset of 1650 fingerprints (3300 images in total). (2) SFinGe: Following the methodology outlined in previous works [3,10], SFinGe software was used to generate three datasets of synthetic fingerprints with varying image quality and natural class distribution. The software parameters included an acquisition area of 14.6 × 19.6 mm^2, a resolution of 500 DPI, and an image size of 288 × 384 pixels. The three generated quality levels were: high quality undisturbed (HQU), default (D), and quality with variant disturbance (QVD). Each of these datasets contains 120,000 samples. The obtained optimal classifiers were compared against the most recent

fingerprint classification methods reported in the state of the art [3,10,23,35,36] in terms of classification accuracy, robustness, and computational complexity. Results indicate that the ELM classifier, using the Hong08 [14] descriptor, achieved a high accuracy of 95.55% with a low penetration rate (0.299) for the NIST-4 database. Moreover, the training and testing times were significantly lower compared to CNN-based approaches, with 0.3811 s for training and 0.0466 s for testing.

2.3. Classical Methodologies

Beyond the two trends mentioned above, other studies have explored statistical machine learning algorithms without employing neural networks. For instance, Rajanna [37] conducted a comparative analysis of four feature extraction methods for fingerprint classification—orientation map (OM), Gabor feature, minutiae map (MM), and orientation collinearity (OC)—which were combined using a rank-based fusion scheme to improve performance. Using the NIST-4 dataset, the combination of OM with OC surpassed the performance of classifiers employing Gabor filters, achieving a classification accuracy of 97.761%. However, the training time was not reported.

Cao [38] proposed a hierarchical classifier consisting of five steps, utilizing SVM and K-NN. The process includes (1) distinguishing as many arch fingerprints as possible using complex filters, (2) identifying the largest subset of whorl fingerprints using central points and a ridge-line flow classifier, (3) applying K-NN to classify the two main categories based on OC features and responses from complex filters, (4) distinguishing loops from non-whorl classes using ridge-line flow, and (5) using an SVM for final classification. Tests conducted on the NIST-DB4 dataset reported an accuracy of 95.9%, with a classification time of 4.31 s.

Guo [1] proposed a rule-based fingerprint classification method using decision trees, relying on singular points and their counts for analysis. For illegible fingerprints, additional features such as Center-to-Delta Flow (CDF) and Balance Arm Flow (BAF) were introduced. The method achieved accuracies of 92.74% and 97.2%, respectively, in classifying four classes without rejection, using three databases from FVC versions (2000, 2002, and 2004). However, processing times were not reported.

Luo [39] introduced an algorithm for fingerprint classification that combines Curvelet Transform (CT) with Grey-Level Co-occurrence Matrices (GLCMs). First, the original image is divided into five scales to reduce noise. Subsequently, the curvelet transform and GLCMs are calculated at a larger scale, generating 16 texture features based on four GLCMs. A total of 49 features are then derived across the remaining four scales. Finally, the classification process is carried out using the K-NN algorithm, employing the NIST-4 dataset consisting of 4000 fingerprint images. This approach achieves an accuracy of 94.6% in 1.47 s.

3. Theoretical Foundations

This section presents the theoretical and mathematical foundations employed for fingerprint feature extraction and classification. Three high-performance feature extraction methods reported in the literature [14,24,25] are considered. These methods are referenced by the first author and the year of publication: Capelli02, Liu10, and Hong08. Furthermore, this study introduces, for the first time, a multilayer perceptron network based on an ELM-AE and the original ELM as a classification algorithm.

3.1. Feature Descriptors

Fingerprint classification is the most commonly employed technique to reduce the penetration rate in fingerprint identification datasets [40]. Most researchers adopt the five-class system proposed by Henry [26], which categorizes fingerprints based on distinct

visual patterns. These classes are unevenly distributed as follows: arch (3.7%), left loop (33.8%), right loop (31.7%), tented arch (2.9%), and whorl (27.9%).

Fingerprint classification algorithms typically rely on global fingerprint features, such as OM, ridge structures, and singular points (SPs). Singular points are commonly detected using the Poincaré method. While OMs are generally extracted through gradient-based methods, alternative approaches, such as cleft-based or skeleton-tracking techniques, have also been proposed. Based on several analyses, Peralta already compared various fingerprint extraction (FE) methods (SVM, 5NN, 10NN, and Kappa) across multiple databases. These evaluations consistently show that the Capelli02, Hong08, and Liu10 methods deliver the best performance [3,41]. Hong08 maintains an accuracy of 93.55% and a Kappa of 0.9083 using K-NN (k = 5, Euclidean distance), suggesting that K-NN may be more suitable for low-quality fingerprint classification than SVM. The Jain method, also using K-NN, shows similar performance, with an accuracy of 93.47% and a Kappa of 0.9072. As the database difficulty increases, the extraction of the new features introduced by Hong08 becomes more challenging. Nevertheless, Hong08 remains the best option compared to other methods, demonstrating a strong ability to handle fingerprints of varying quality. Capelli02 [24] is a method based on the fingerprint OM; the OM is aligned using a central point detected by the Poincaré method. A dynamic mask is applied to each class, generating a five-element vector. The orientations are also recorded as part of the feature set. The authors combined two distinct approaches for feature extraction and classification. Additionally, features from the earlier Capelli99a [42] method were incorporated into the feature vector. Once the OM is registered, it is enhanced following the procedure described in Capelli99a. Finally, orientations are recorded in radians within the feature vector, whose size depends on the original image dimensions. These features are then processed using the Multi-space KL (MKL) classification method [43].

The Hong08 [14] method extends the FingerCode feature vector (derived from Gabor filters) by incorporating pseudo-ridges traced from the fingerprint center, along with the number of singular points (nucleus and deltas), and their distance and location. This approach builds upon the FingerCode method proposed by Jain, adding ten new features derived from the SPs and pseudo-ridges. SPs are extracted using the Karu method [44], while the pseudo-ridge is based on the Zhang04 method [45]. In this case, pseudo-ridges are traced from the center of the fingerprint, which is defined as the closest central point detected or remains unchanged if no central point is identified. Once the SP and pseudo-ridge are calculated, ten additional features are added to the FingerCode feature vector. These features include the number of nuclei and deltas, as well as the relative location and distance of the deltas and the endpoints of pseudo-ridges in relation to the fingerprint center. It should be noted that these features are categorical, as distances and relative locations are discretized into four and six values, respectively.

The Liu10 approach [25] differs significantly from other methods by exclusively considering relative measures of SPs as features. A total of 16 features are extracted at four different scales to enhance robustness against noise. All features are related to the primary central point, including its position, orientation, and degree of freedom. Additional features are defined as relative measures between the central point and the other SPs. The Liu10 method can be summarized in four main steps. First, the OM is extracted using a gradient method, incorporating an additional smoothing step based on the coherence concept [46]. Subsequently, the fingerprint is segmented following the method proposed by Bazen and Gerez [47]. The key phase involves SP extraction, which is performed using complex filters and a multiscale model [48]. This extraction process is repeated across four different OM scales to improve the reliability of SP detection. Finally, the feature vector is formed based on the extracted SP measurements.

3.2. Extreme Learning Machines

3.2.1. Original ELM

The concept of an ELM was proposed for the training of single-layer feedforward neural networks (SLFNs) in 2006 [17]. ELMs have gained significant popularity due to their ease of implementation. Huang et al. [17] demonstrated that ELMs outperform gradient-based artificial neural networks and SVMs in terms of prediction performance for both classification and regression tasks, offering the unquestionable advantage of negligible computation time, even when executed on standard commercial computers [13].

Given a training dataset containing M samples, a basic ELM maps inputs (data samples) to outputs (labels) using a single hidden layer comprising N nodes. The ELM can be expressed as follows [13]:

$$H\gamma = T, \tag{1}$$

$$H = \begin{bmatrix} f(c_1 \cdot x_1 + b_1) & \cdots & f(c_M \cdot x_1 + b_N) \\ \vdots & f(c_j \cdot x_i + b_j) & \vdots \\ f(c_1 \cdot x_M + b_1) & \cdots & f(c_M \cdot x_M + b_N) \end{bmatrix},$$

$$\gamma = \left[\gamma_1^T, \quad \cdots, \quad \gamma_N^T\right]^t,$$

$$T = \left[t_1^T, \quad \cdots, \quad t_M^T\right]^t,$$

where H represents the output matrix of the hidden layer, γ denotes the matrix of output weights between the hidden layer and the output layer, and T corresponds to the label matrix. The function $f(\cdot)$ is the activation function, which is a nonlinear piecewise continuous function, such as a sigmoid function. The c_j vector is the input weight connecting the input node and the j-th hidden node, and $x_i \in \mathbb{R}^n$ represents the i-th input data point, where n is the dimensionality of the input layer. The b_j parameter is the bias of the j-th hidden node, and γ_j denotes the output weight vector connecting the j-th hidden neuron to the output nodes. Finally, $t_i \in \mathbb{R}^m$ is the m-dimensional target vector corresponding to x_i.

Additionally, c_j and b_j are derived from any continuous probability distribution, such as the uniform distribution, reducing the need for manual intervention. This simplification affects the $c_j \cdot x_i$ term, which is calculated as part of the mapping process. The original structure of the ELM is illustrated in Figure 1, where all parameters, including inputs and outputs, are detailed for clarity and comprehension.

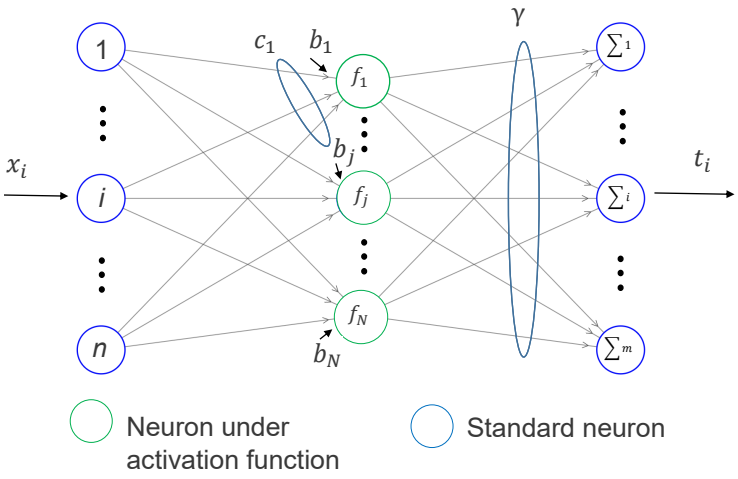

Figure 1. General architecture of an original ELM.

Moreover, the least squares solution with a minimum norm can be calculated analytically through the generalized Moore–Penrose inverse of H as follows [13]:

$$\gamma = \begin{cases} \left(H^T H + I/C\right)^{-1} H^T T & M > N \\ H^T \left(H^T H + I/C\right)^{-1} T & M \leq N' \end{cases} \quad (2)$$

where I is an identity matrix and C is a regularization parameter ($\in \mathbb{R}^+$). The dimensions of I depend on the ratio between N and M, and C is added to balance the training error and the norm of the output weights, avoiding overfitting. To simplify the analysis, the second term, I/C, is omitted from the equation. This removes the regularization factor, allowing the use of a standard ELM and thereby reducing the model's complexity. To summarize the ELM training process, its algorithm is presented below (Algorithm 1) [13].

Algorithm 1 ELM Training Algorithm

Given the training set $\phi = \{(x_i, t_i) \mid i = 1, \ldots, M\}$, set the hyperparameters, including the activation function $f(\cdot)$, and the number of hidden neurons N.

1: Set the input weights c_i and biases b_i randomly.
2: Compute the output of the hidden layer matrix H.
3: Set the output weights γ.

3.2.2. Multilayer ELM

The features of the dataset influence the learnability during both the training and the testing phases for the ELM algorithm. Consequently, feature design becomes essential, as it allows for a more effective representation of the data's inherent structure. However, this task requires expertise in the area, as well as human ingenuity, to develop suitable features. To address this challenge, AEs can be employed to train multilayer perceptron networks. An AE can be defined as an SLFN with multiple hidden layers [29]—in this sense, an unsupervised training paradigm, where the outputs and inputs are the identical. The goal of the AE is to reproduce the input signal as accurately as possible through the connections between its neurons and the biases in its hidden layers.

This paper presents an ELM based on AEs that learns singular feature values to learn feature representations and classify fingerprints while reducing the penetration rate in identification databases. An AE-based multilayer M-ELM is constructed with a training architecture divided into two distinct phases: an unsupervised hierarchical feature representation phase and a supervised label classification phase (see Figure 2). In the initial phase, the ELM-AE is designed to extract sparse multilayer features from the input data using unsupervised learning. In the subsequent phase, a standard ELM is employed for final decision making. The detailed processes for each phase are presented in Algorithm 2 (unsupervised feature extraction using ELM-AE) and Algorithm 3 (supervised classifier learning phase (standard ELM)). These algorithms outline the step-by-step procedures for both phases, ensuring clarity and reproducibility of the proposed approach. Building on the supervised learning process outlined in the previous subsection, the training procedure for the unsupervised AE block in the M-ELM architecture is systematically developed. This two-phase approach ensures robust feature extraction and efficient classification.

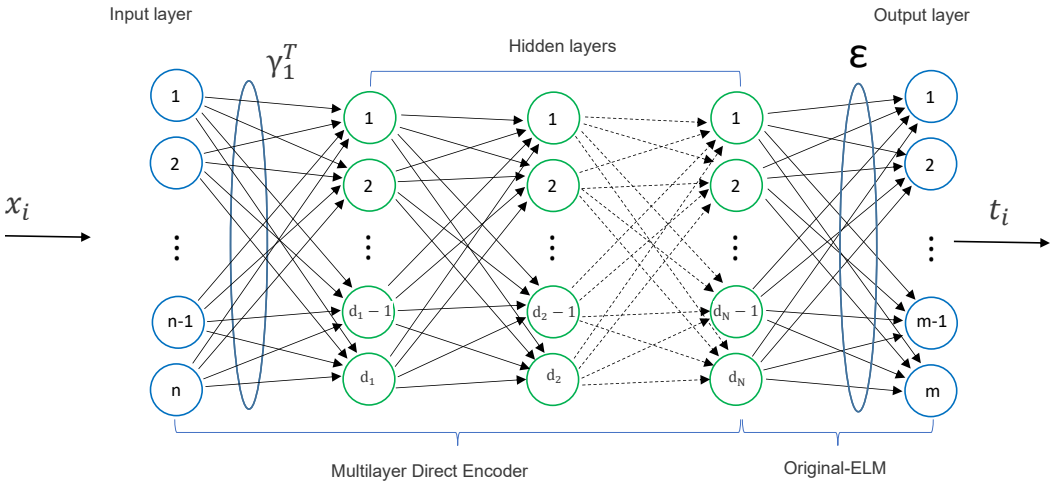

Figure 2. Representative structure of a multilayer ELM.

An ELM-AE can be conceptualized as a neural network comprising an input layer, a hidden layer, and an output layer (see Figure 3). It is characterized by having a number (n) of neurons in the input layer, d neurons in the hidden layer, and n neurons in the output layer and employing a nonlinear activation function ($f(\cdot)$) for the hidden neurons. In contrast, the neurons in the output layer are typically limited to performing summation operations. For a set of M distinct samples ($x_i \in \mathbb{R}^{M\times j}, i = 1, \ldots, M$), the weights ($c$) and biases ($b$) corresponding to the hidden-layer nodes are randomly generated based on a probability density function. These weights and biases are subject to the orthogonality condition, which enhances the training capacity of the neural network. The output of the hidden layer in the ELM-AE and the parameters between the input and output layers can be mathematically expressed as follows:

$$h = f(cx + b), \tag{3}$$

subject to

$$c^T c = I, \tag{4}$$

and

$$b^T b = 1, \tag{5}$$

where $c = [c_1, \ldots, c_M]$ denotes the orthogonal random weight between the input and hidden layer, $b = [b_1, \ldots, b_M]$ represents the orthogonal random biases between the input nodes and the hidden nodes, I corresponds to the identity matrix, and ϵ refers to the output of the connections between the last hidden layer and the output nodes. As demonstrated, the standard ELM is modified for unsupervised learning by using the input data as the output data. It is worth mentioning that the ELM-AE is designed to effectively extract input features through three distinct approaches: (1) a compressed approach, which represents entities from a higher dimensional input data space to a lower dimensional entity space; (2) a sparse approach, which is the opposite of the compressed approach; and (3) an equal-dimensional approach , where entities maintain their dimensionality in order to maximize the performance of the ELM. Due to the ELM-AE's output weights, it is possible to transform features from the entity space back to the input data space. Across all ELM-AE representations, the output weights can be expressed in the following form:

$$\gamma = \begin{cases} (H^T H)^{-1} H^T X & M > N \\ H^T (H^T H)^{-1} X & M \leq N \end{cases} \tag{6}$$

where $H = [h_1, \ldots, h_M]$ represents the outputs of the hidden layer of the ELM-AE, denotes the operator associated with the generalized Moore–Penrose inverse of a matrix, and $X = [x1, \ldots, x_M]$ corresponds to both input and output data of the ELM-AE (unsupervised training. Additionally, N denotes the number of hidden nodes. The output weights of the ELM-AE are specifically designed to effectively extract features (or entities) from the input data by leveraging singular values. The parameter of regularization (C) is excluded in ELM-AEs for simplicity.

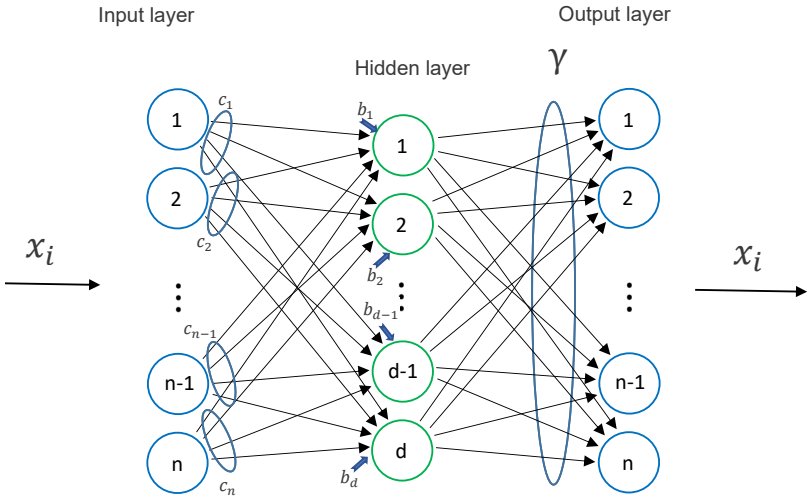

Figure 3. General architecture of an ELM-AE. The colors indicate the type of neuron, following the same scheme as the original ELM.

Algorithm 2 Unsupervised Feature Extraction Phase (Autoencoder)

Dataset $X = \{x_1, x_2, \ldots, x_M\}$, where $x_i \in \mathbb{R}^{n \times j}$, number of hidden neurons d, activation function $f(\cdot)$ Extracted features from the autoencoder
1: Initialize random weights $c = \{c_1, c_2, \ldots, c_M\}$ and biases $b = \{b_1, b_2, \ldots, b_M\}$ such that $c^T c = I$ and $b^T b = 1$
where I is the identity matrix
2: For each sample x_i, calculate the hidden layer output:

$$h_i = f(cx_i + b)$$

3: Construct the hidden layer output matrix $H = [h_1, h_2, \ldots, h_M]$
4: Calculate the output weights γ by solving the linear system:

$$\gamma = \begin{cases} (H^T H)^{-1} H^T X & \text{if } M > N \\ H^T (H^T H)^{-1} X & \text{if } M \leq N \end{cases}$$

5: Return the learned features H.

The ELM-AE is stacked layer by layer following a hierarchical structure. Prior to supervised least squares optimization, the weights of each M-ELM hidden layer are initialized using the ELM-AE, which performs unsupervised learning at the layer level, excluding random entity mapping. Mathematically, the output of each hidden layer is written as follows:

$$H^i = f[(\gamma^i)^T H^{i-1}], \tag{7}$$

where H^i represents the output matrix of the i-th hidden layer (for $i - 1 = 0$ is the input layer, and the inputs to the M-ELM are represented as H^{i-1}). Once the features of the previous hidden layer have been computed, the weights and hyperparameters (including

the activation function and the number of hidden neurons) for the current hidden layer are determined. Within this multilayer learning framework, the AE functions as a feature extractor, utilizing the encoded results to approximate the original inputs while minimizing reconstruction errors. The output of the connections between the final hidden layer and the output node ($t_i \in \mathbb{R}^{N\times m}, i = 1, \ldots, N$) can be determined analytically by solving a linear system, similar to the standard ELM approach.

Algorithm 3 Supervised Classifier Learning Phase (Standard ELM)

Extracted features H from AE, labels T, number of hidden neurons N, activation function $f(\cdot)$ Trained classifier output weights γ_{ELM}

1: Initialize random weights and biases for the ELM hidden layer.

2: Compute the hidden layer output matrix H_{ELM} using the extracted features H:

$$H_{ELM} = f(W \cdot H + b)$$

where W are the weights and b are the biases for the hidden layer.

3: Solve for the output weights γ_{ELM} by minimizing the least squares error:

$$\gamma_{ELM} = (H_{ELM}^T H_{ELM})^{-1} H_{ELM}^T T$$

4: Return the trained classifier output weights γ_{ELM}.

4. Methods and Materials

To evaluate performance, the 5-FCV scheme is used [49], ensuring unbiased and precise classification metrics [10,13]. This is because the validation and training sets are not static but come from five parts the DB. Thus, 20% is taken for validation, while the remaining four parts are used for training. For each database, feature descriptor, and ELM approach, the overall results are reported from an average of five runs. Evidently, the validation set is intended to find the ELM hyperparameters that maximize its classification ability, and only the number of hidden neurons (N) is considered for heuristic optimization for simplicity and performance purposes (see next section). Accuracy and absolute error of the PR. Accuracy is chosen, as it is a widely accepted metric for artificial intelligence models and aligns with previous studies used for comparison [10,13,29,50,51]. PR, on the other hand, is particularly relevant for addressing unbalanced datasets, making it a standard metric in fingerprint analysis [10,13]. The absolute value of PR allows for straightforward identification of the superior model, with values approaching zero indicating a better penetration rate. Given the naturally unbalanced distribution of the database, the target PR for this study is 0.2948 [13].

In this study, SFINGE software [52] was utilized to facilitate comparisons with significant and recent works reported in the state of the art [10,13]. In addition, this software generates a substantial database consisting of thousands of fingerprint samples, enabling conclusive observations regarding fingerprint-based individual classification [10,13,51]. SFINGE software produces synthetic fingerprints of different quality levels (low, medium, and high) with a realistic class distribution (unbalanced classes). These synthetic fingerprints generated by SFINGE simulate fingerprints obtained by optical scanners, which are much closer to real systems, such as NIST-4 [53]. Moreover, it is worth noting that the feasibility of adopting the SFINGE database has already been demonstrated in several editions of the FVC [54–58], where it achieved results comparable to those of real fingerprint databases, following the natural class distribution. To emulate various scenarios, three quality profiles were generated using SFINGE: HQNoPert, Default, and VQandPert (see Figure 4). The HQNoPert database consists of high-quality, undisturbed fingerprints. The default database includes fingerprints of medium quality and slight perturbations

in localization and rotation. The VQAndPert database comprises fingerprint of varying quality levels, incorporating perturbations related to the location, rotation, and geometric distortions. The primary distinction among the databases lies in the quality of the generated images. In total, 10,000 fingerprints were generated for each quality category, resulting in a comprehensive dataset of 30,000 fingerprints. The generation of 10,000 fingerprints per category ensures a fair and consistent basis for comparison with the studies reported in the state of the art [10,13].

Figure 4. Samples concerning fingerprint image quality: (**a**) default; (**b**) HQNoPert; (**c**) VQAndPert.

5. Results and Discussion

This section addresses three key aspects: hyperparameter optimization, evaluation and comparison with the state of the art, and complexity analysis. In hyperparameter optimization, hidden neurons and additional layers in the ELM models were adjusted to improve generalization and robustness. In the comparison with the state of the art, the performance of the ELM-M3 model was evaluated against the benchmarking models in fingerprint classification tasks. Finally, the complexity analysis shows that, while ELM-M3 is computationally more expensive than its original version, it is more efficient than CNN-based approaches, achieving a balance between simplicity and performance.

5.1. Hyperparameter Optimization

For this study, the sigmoid function ($g(x) = 1/[1 - exp(x)]$) is considered as the activation function, since it guarantees the universal approximation of the SLFNs, making it suitable for any ELM algorithm dealing with nonlinear problems. In addition, the regularization parameter (C) is omitted across all ELM versions for simplicity. It should be noted that the inclusion of the regularization parameter (C) in the model serves to prevent overfitting by balancing the training error and the norm of the output weights. Hyperparameter optimization for the different ELM models is first performed heuristically with the aim of maximizing their performance while minimizing complexity, thereby avoiding overtraining. This process considers various feature extractors (Capelli02, Hong08, and Liu10) and fingerprint qualities (Default, HQNoPert, and VQAndPert). Figures 5–7 illustrate the performance in terms of the number of hidden neurons in the original ELM (single hidden layer), showing results from both the training and validation stages. The training results demonstrate the successful convergence of single-hidden-layer networks trained with the ELM algorithm while highlighting the issue of overfitting that occurs as network complexity (i.e., the number of hidden neurons) increases. The validation results, however, are more indicative of the generalization capacity of the model, since they reflect performance on

unseen data. For each descriptor, there is a number of hidden neurons that maximizes the generalization ability of the ELM. This number is only slightly dependent on the fingerprint quality. Nevertheless, for robustness and generalization purposes (so that the results are independent of the fingerprint quality), the following numbers of hidden neurons are adopted: 1000, 2000, and 750 for the Capelli02, Hong08, and Lui10 descriptors, respectively. The accuracies achieved by these numbers of neurons are highlighted with markers. Note that the number of hidden neurons is constrained to 8000 samples, since, theoretically, it has been shown that to achieve the best performance of an ELM, the number of hidden neurons must be less than the number of samples. The following subsection discusses the results regarding fingerprint quality and descriptors in a particular and general manner.

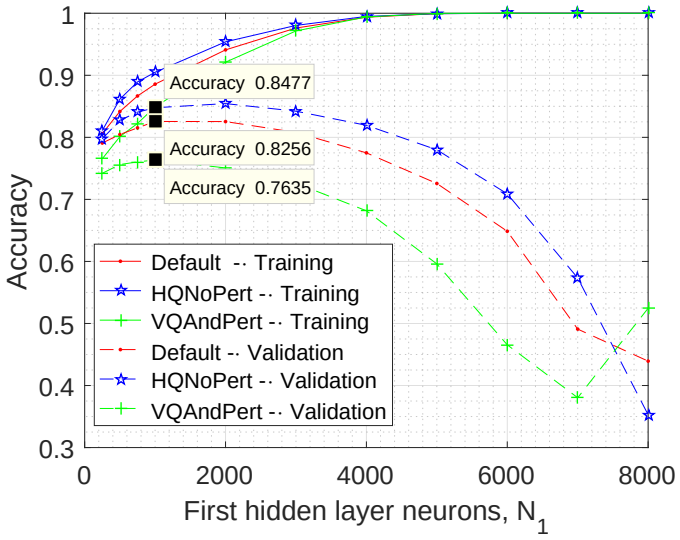

Figure 5. The accuracy in the training and validation phases in terms of the number of hidden neurons of the original ELM, considering Capelli02 as a descriptor.

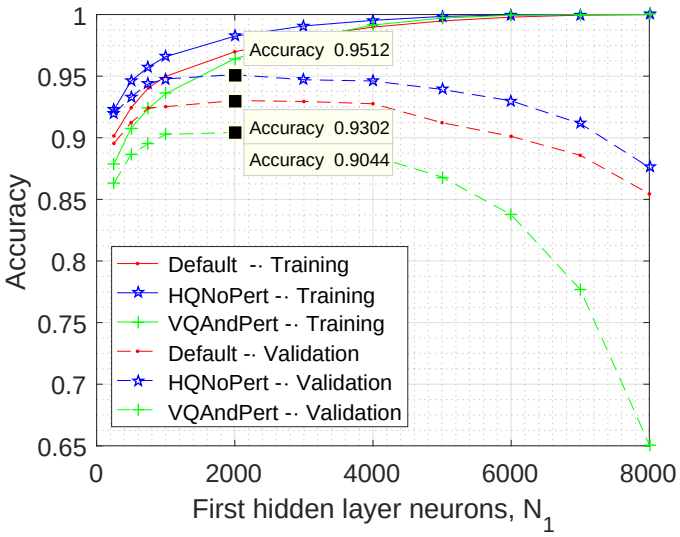

Figure 6. The accuracy vs. the number of hidden neurons of the original ELM in the training and validation phases when the descriptor corresponds to Hong08.

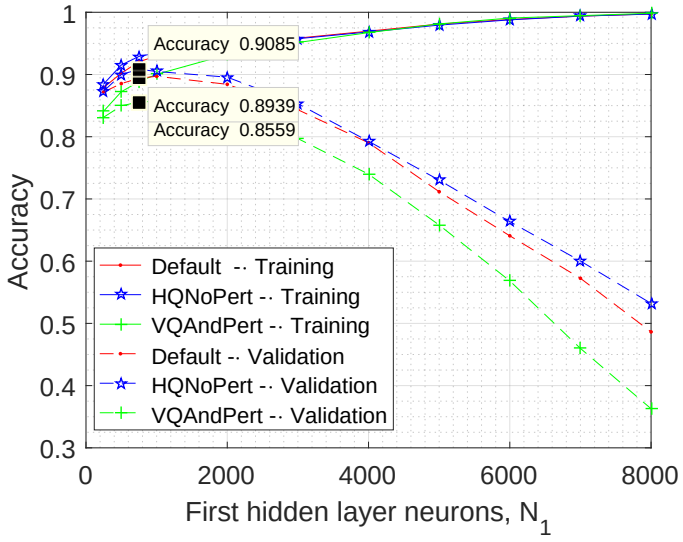

Figure 7. The accuracy vs. the number of hidden neurons of the original ELM in the training and validation phases when the descriptor corresponds to Liu10.

The results regarding the optimization of two- and three-hidden-layer ELM models (M-ELM) are presented below in Figures 8–13.

In Figures 8–10, performance is displayed as contour plots, showing the relationship between the number of neurons in the first and second layers. An irregular zone can be observed where the accuracy reaches its highest value, which is slightly dependent on the quality and, more noticeably, on the descriptor. To avoid overfitting, the optimization yields the following configuration: both the first and second hidden layers contain 1000 neurons ($N1 = N2 = 1000$). Increasing the complexity of the ELM (e.g., adding more neurons in each hidden layer) does not necessarily maximize performance, as evidenced by the significant decrease in accuracy when $N1 = N2 = 7000$. Furthermore, the optimal performance is achieved when the first hidden layer contains more neurons than the second. This is because one of the functions of unsupervised learning in ELM autoencoders is to derive meaningful feature representations at the input level. Thus, simplifying the network architecture in deeper layers is logical to maintain optimal performance.

The optimization results for the three-hidden-layer M-ELM model are shown in Figures 11–13, where different descriptors and fingerprint qualities are considered to ensure robustness in the findings. The results exhibit a volumetric irregularity in the optimized hidden-node values, with accuracy approaching 1.

To achieve efficient generalization, a simple architecture is selected; each figure includes an arrow to indicate the relevant feature. In addition, for visualization purposes, the desired accuracies are highlighted with larger characters (cloud of squares). Once more, it is observed that as additional hidden layers are incorporated, the number of required neurons decreases. This reduction is attributed to the fact that the initial layers are responsible for extracting the primary features of the fingerprints.

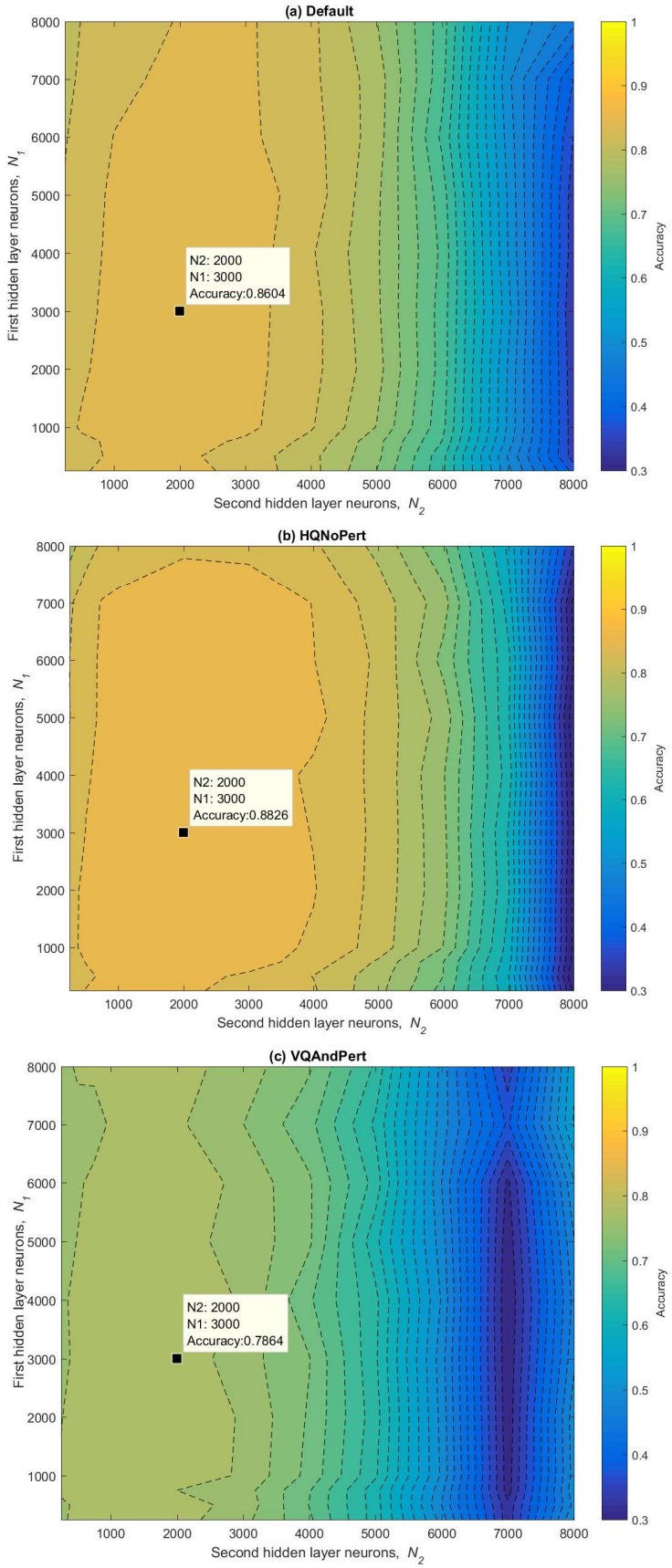

Figure 8. Accuracy in terms of the number of neurons of the two-layer hidden ELM considering the Capelli02 descriptor and the (**a**) default, (**b**) HQNoPert, and (**c**) VQAndPert databases.

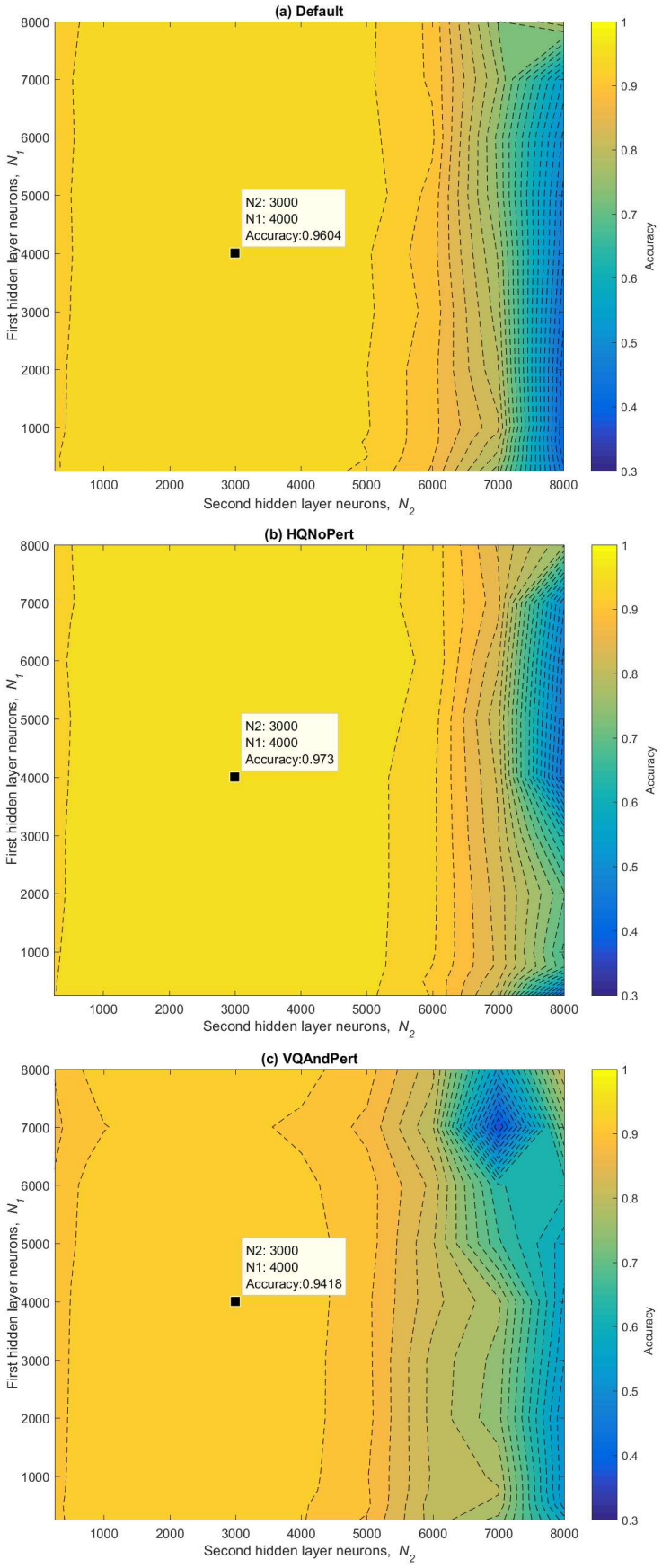

Figure 9. Accuracy in terms of the number of neurons of the ELM of two hidden layers considering the Hong08 descriptor and the (**a**) default, (**b**) HQNoPert, and (**c**) VQAndPert databases.

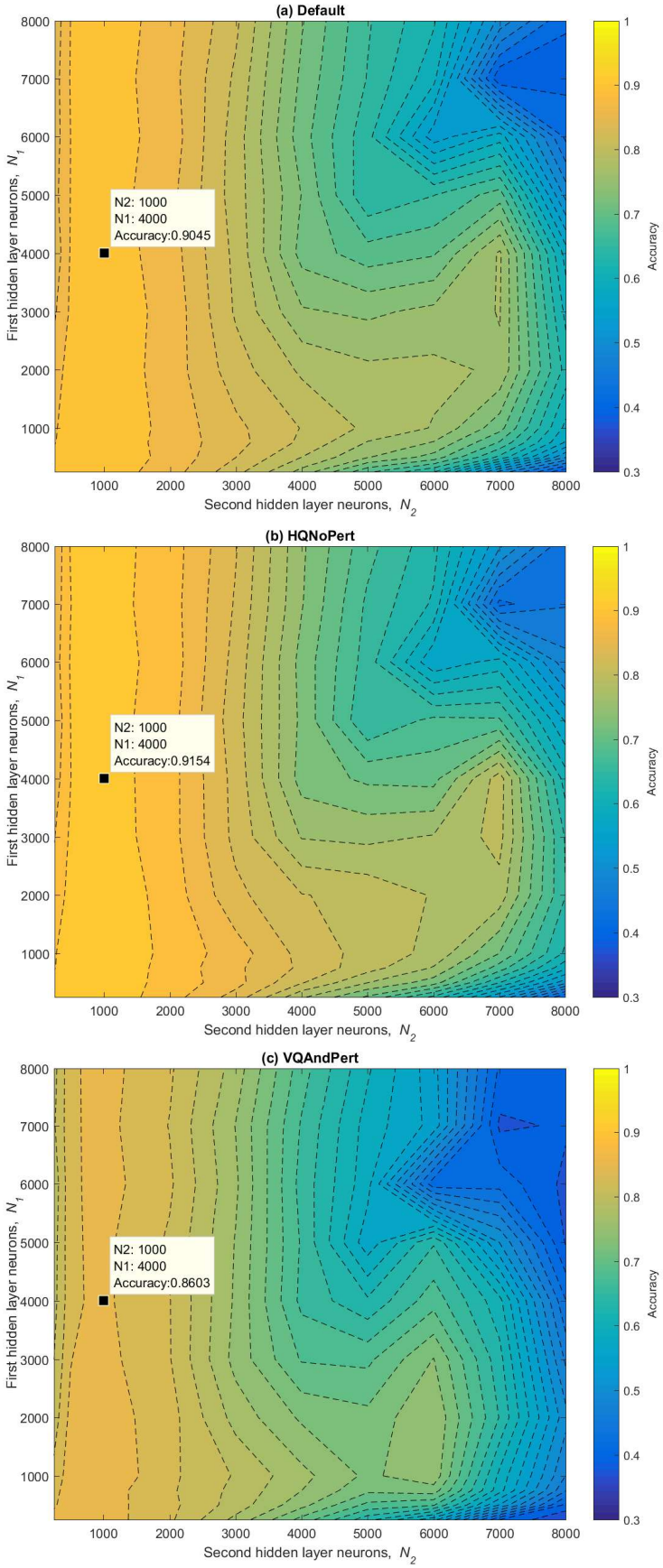

Figure 10. Accuracy as a function of the number of neurons of the two-hidden-layer ELM considering the Liu10 descriptor and the (**a**) default, (**b**) HQNoPert, and (**c**) VQAndPert databases.

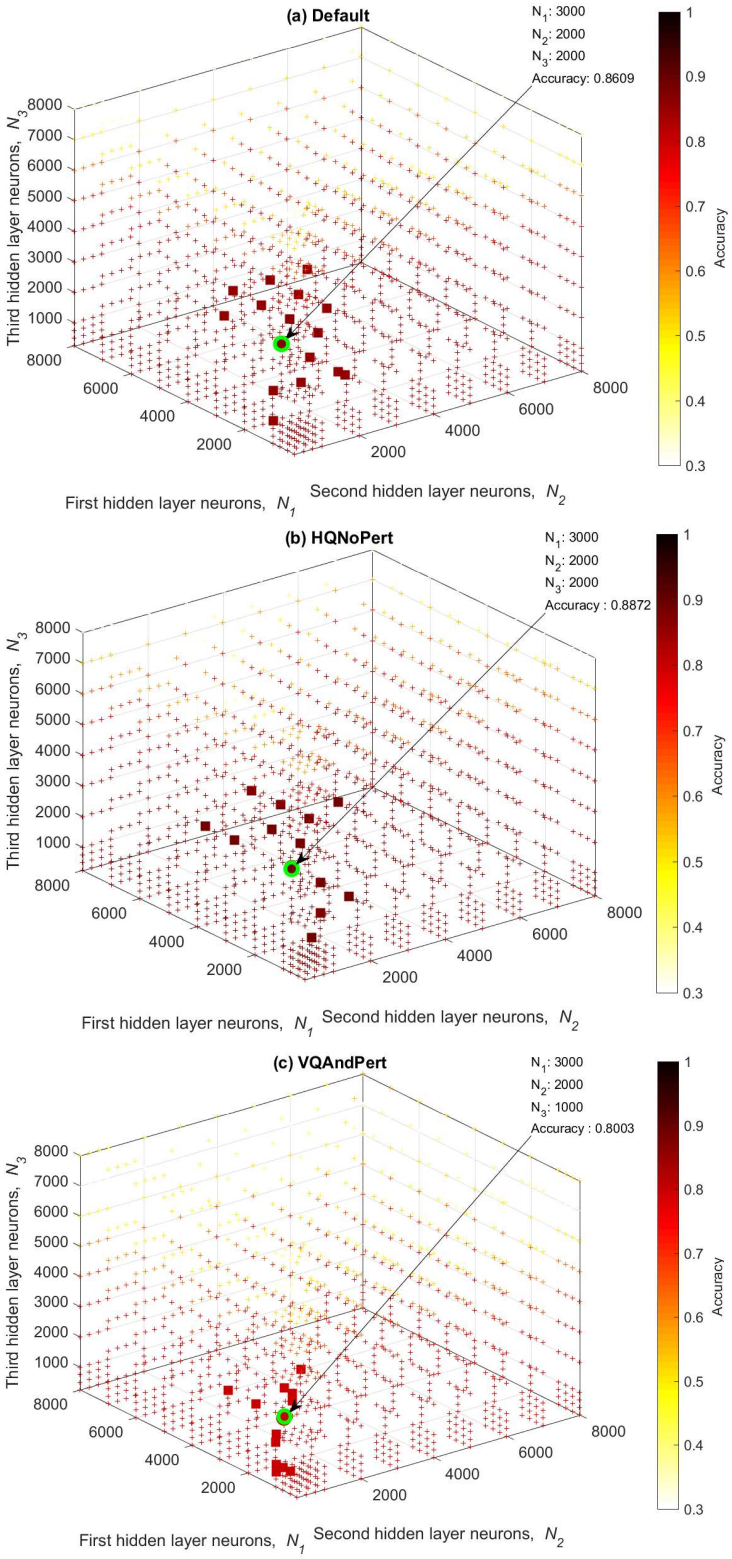

Figure 11. Accuracy as a function of the number of neurons of the three-hidden-layer ELM, taking into account the Capelli02 descriptor and the (**a**) default, (**b**) HQNoPert, and (**c**) VQAndPert databases.

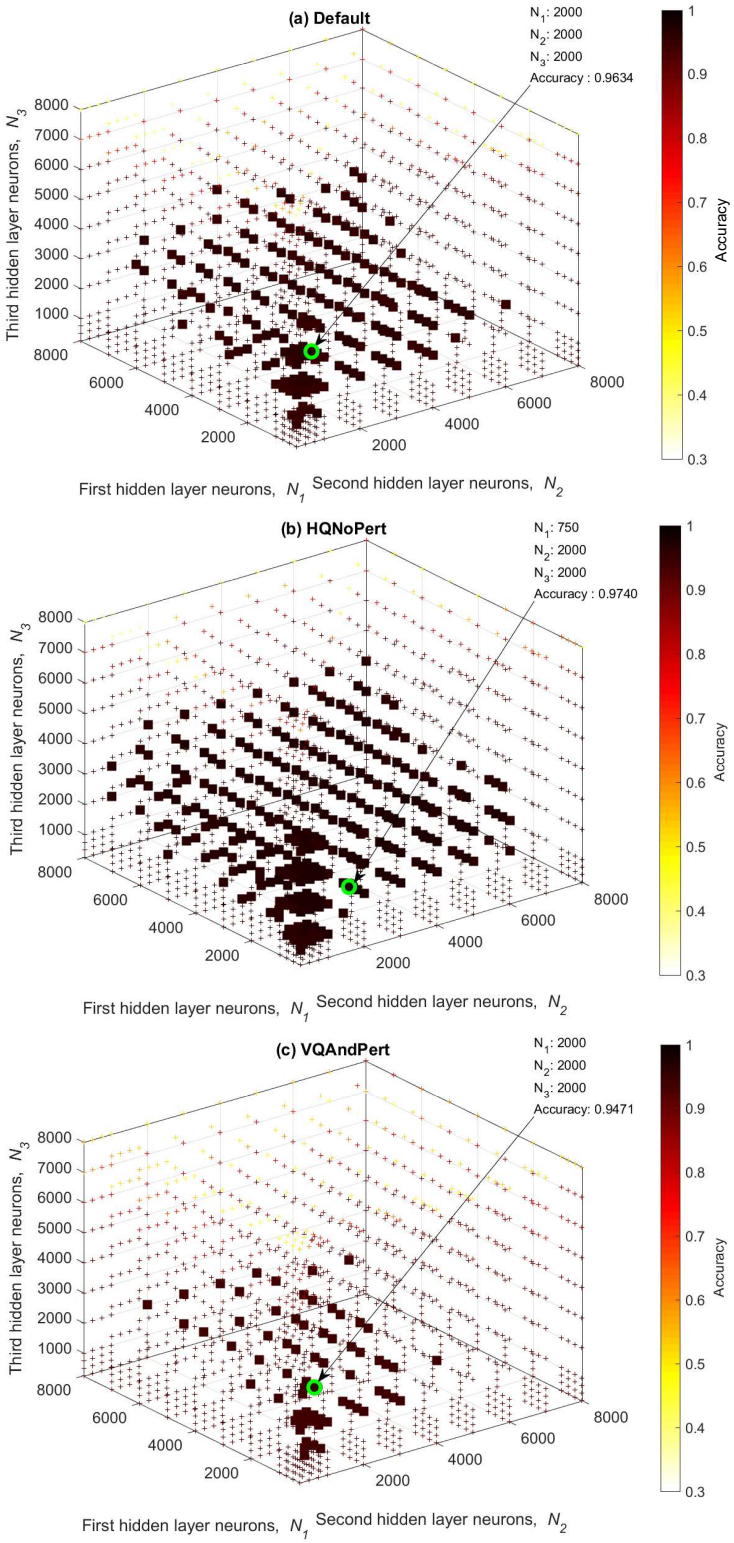

Figure 12. Accuracy as a function of the number of neurons of the three-hidden-layer ELM, taking into account the Hong08 descriptor and the (**a**) default, (**b**) HQNoPert, and (**c**) VQAndPert databases.

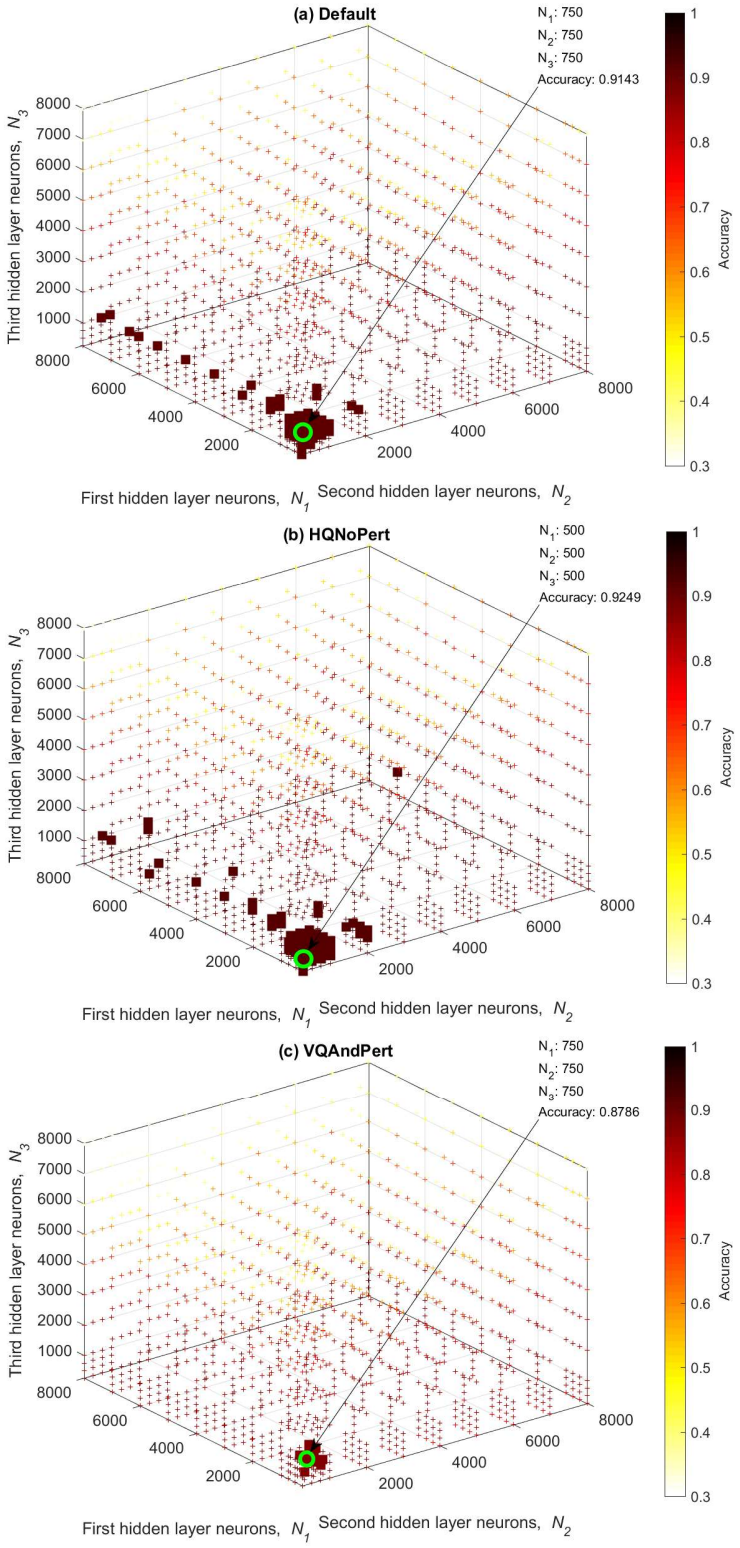

Figure 13. Accuracy as a function of the number of neurons of the three-hidden-layer ELM, taking into account the Liu10 descriptor and the (**a**) default, (**b**) HQNoPert, and (**c**) VQAndPert databases.

The accuracy of the optimized hyperparameters for the various ELM models is summarized in Table 1. Notably, the Hong08 descriptor, coupled with high-quality fingerprints (HQNoPert), yields the best performance. Moreover, the addition of a second hidden layer generally improves the classification accuracy by approximately 0.02 compared to the original ELM. The introduction of a third hidden layer further increases accuracy by an additional 0.01. The absolute values of the penetration rates for different ELM models are

provided in Table 2, considering the various descriptors (Capelli02, Hong08, and Liu10) and fingerprint qualities (default, HQNoPert, and VQAndPert). These results demonstrate the robustness of the proposed models in dealing with imbalanced learning problems, where success in the minority class is as important as in the majority class. The Hong08 descriptor demonstrates notable performance across various fingerprint qualities, similar to its behavior with accuracy metrics. Specifically, the improvement in the absolute value of the penetration rate achieved by the two-hidden-layer ELM compared to the original ELM (independent of fingerprint quality) is 0.0045, while the difference between the two- and three-hidden-layer ELMs is 0.0055. Consequently, only the results associated with the Hong08 descriptor are considered in subsequent analyses. In contrast, the worst performance is observed using the Capelli02 descriptor, not only across all fingerprint qualities but also across all evaluated metrics. This limitation arises from the method applied in Capelli02, which combines two approaches based on feature extraction via orientation maps (OMs). Furthermore, the Cappelli99a [42] method, which segments OMs into areas with similar orientations and compares them to predefined templates for each fingerprint class (arch, left loop, right loop, tented arch, and whorl) using a defined cost function, exhibits additional limitations. The class with the lowest segmentation cost is selected, and the resulting orientation map is used to populate the second part of the feature vector, centered on the Poincaré-determined core point. This approach is more prone to class confusion compared to Hong08 and Liu10. Experimental evidence from previous studies has consistently identified the Hong08 descriptor as the most reliable and effective option [10,13,34].

Table 1. Numbers of hidden neurons that maximize the prediction of the (a) original ELM, (b) ELM-M2, and (c) ELM-M3.

(a) Original ELM	Capelli02		Hong08		Liu10	
	N_1	Exac	N_1	Exac	N_1	Exac
Default	1000	0.8256	2000	0.9302	750	0.8939
HQNoPert	1000	0.8477	2000	0.9512	750	0.9085
VQAndPert	1000	0.7635	2000	0.9044	750	0.8559
(b) ELM-M2	Capelli02		Hong08		Liu10	
	N_1/N_2	Exac	N_1/N_2	Exac	N_1/N_2	Exac
Default	1000/1000	0.8604	1000/1000	0.9604	1000/1000	0.9045
HQNoPert	1000/1000	0.8826	1000/1000	0.9730	1000/1000	0.9154
VQAndPert	1000/1000	0.7864	1000/1000	0.9418	1000/1000	0.8603
(c) ELM-M3	Capelli02		Hong08		Liu10	
	$N_1/N_2/N_3$	Exac	$N_1/N_2/N_3$	Exac	$N_1/N_2/N_3$	Exac
Default	1000/250/1000	0.8609	250/250/1000	0.9634	250/250/250	0.9143
HQNoPert	1000/1000/1000	0.8872	250/250/750	0.9740	250/250/250	0.9249
VQAndPert	500/250/1000	0.8003	750/750/750	0.9471	500/500/500	0.8786

Table 2. Absolute values of the penetration rate obtained by the different ELMs according to the studied databases and descriptors.

	Original ELM			ELM-M2			ELM-M3		
	Capelli02	**Hong08**	**Liu10**	**Capelli02**	**Hong08**	**Liu10**	**Capelli02**	**Hong08**	**Liu10**
Default	0.1430	0.0523	0.0976	0.1407	0.0466	0.0837	0.1322	0.0426	0.0789
HQNoPert	0.1228	0.0381	0.0917	0.1183	0.0378	0.0749	0.1169	0.0335	0.0739
VQAndPert	0.1865	0.0736	0.1138	0.1839	0.0591	0.1335	0.1777	0.0575	0.1154

5.2. Evaluation and Performance Comparison with the State of the Art

Recent studies have employed ELMs for biometric classification tasks, such as fingerprint and [10,13,33,34] palm vein recognition [2], with an emphasis on age and gender detection using single-layer ELMs that include a regularization parameter. In order to compare our proposal with the most recent and important studies reported in the literature, performance metrics such as accuracy (overall performance of all classes) and penetration rate (performance of each class) are considered, making this comparison more conclusive. Table 3 compares the results obtained in this work using the Hong08 descriptor with the performances reported by Peralta et al. [10] and Zabala et al. [13]. The results of [33,34] are omitted. The work by Peralta et al. [10] achieved an accuracy of 98.7% but only considered a DB of scare fingerprint samples (FVC-2004 with 880 samples). Moreover, details about the ELM algorithm (hyperparameters) were not exposed, since the study focused on the fingerprint feature descriptor. The second study, apart from obtaining lower accuracy results (83.45%), performed heuristic optimization considering a lower number of hidden nodes than the number of training samples for the SFINGE database, meaning that optimization presented local and not global maxima. On the one hand, it is appreciated that the work based on a CNN [10] achieved the best performance for both accuracy (99.6%) and the absolute value of the penetration rate (0.31). This is practically perfect if the fingerprint is acquired in an ideal manner (HQNoPert). The superiority of the approach proposed in [10] is distinguishable for all fingerprint qualities, being only 0.02 better in terms of accuracy and 0.03 in terms of PT than the three-hidden-layer ELM proposed in this paper. The CNN developed by Peralta et al. [10], which incorporates a modified version of AlexNet with softmax functions, demonstrates superior performance compared to the architecture proposed in this study. The AlexNet architecture is characterized by five convolutional hidden layers and three fully connected hidden layers, designed based on the ELM algorithm. In [10], modifications were made to the number of units within the network to simplify the search space during training and enhance both the speed and convergence of the process. The network was trained using the stochastic gradient descent (SGD) algorithm, which reduced computational costs by utilizing only a subset of training instances in each iteration. This approach introduces a bias in error calculation relative to the optimum but significantly accelerates the process. In addition, this network is used together with images to classify fingerprints without requiring an explicit feature extraction process. This approach addresses challenges associated with noise or distorted fingerprint images more effectively than our proposed method, which relies on descriptor layers based on an ELM. While our approach demonstrates a 0.04 improvement in accuracy compared to the unbalanced ELM method characterized by the golden ratio, as proposed by Zabala-Blanco20 [13], it exhibits a marginal disadvantage of 0.0003 in terms of PR. This slight drawback can be attributed to variance in the sample mean of the measurements. When evaluating performance relative to fingerprint quality (independent of the classification model), it is evident that fingerprints of the highest quality (HQNoPert) achieve approximately 0.02 higher accuracy and 0.05 higher Pr rates compared to those of average quality (default). Similarly, HQNoPert

fingerprints outperform those of the lowest quality (VQAndPert) by 0.04 in accuracy and 0.03 in PR, highlighting the influence of image quality on classification performance.

Table 3. Performance comparison between the best identified methodologies and the other considered approaches.

	Hong08 and Original ELM Original		Hong08 and ELM-M2		Hong08 and ELM-M3		Zabala-Blanco20 [13]		Peralta18 [10]	
	Exac	**| *PR* |**	**Exac**	**| *PR* |**	**Exac**	**| *PR* |**	**Exac**	**| *PR* |**	**Exac**	**| *PR* |**
Default	0.9302	0.0523	0.9604	0.0466	0.9634	0.0426	0.9300	0.0388	0.9807	0.0153
HQNoPert	0.9512	0.0381	0.9730	0.0378	0.9740	0.0335	0.9400	0.0299	0.9960	0.0031
VQAndPert	0.9044	0.0736	0.9418	0.0591	0.9471	0.0575	0.8800	0.0533	0.9640	0.0279

5.3. Complexity Analysis

Another important parameter to consider when comparing different learning techniques is the computational cost, particularly the training time. The computational complexity of the proposed models was evaluated on a computer equipped with an Intel® Core™ i7-6700 processor, 8 GB of RAM, and a 64-bit Windows 10 operating system (Santa Clara, CA, USA). Table 4 reports the training times (in seconds), along with the sample mean and standard deviation, for different fingerprint qualities. A total of 1000 simulations were conducted to ensure representative results for each case to validate the observations. It is important to mention that the results presented for [10] were directly replicated using high-performance hardware—specificallym an Nvidia GeForce GTX TITAN GPU (2688 cores, 6144 MB GDDR5 RAM) and a 2.6 GHz Intel Core i5 processor with 4 GB of RAM (Santa Clara, CA, USA). Regarding the software environment, Peralta et al. [10] employed the C++ Caffe library, a low-level language offering faster execution than the programming language utilized in this study (MATLAB 2024a). Given the focus of this work on ELMs and acknowledging the hardware and software limitations compared to the CNN-based model, a complexity analysis was performed. Our method is notoriously disadvantaged in terms of hardware/software. As demonstrated in Table 4, the ELM architecture with three hidden layers emerges as the optimal alternative in this study, a conclusion supported by the performance metrics discussed in the preceding subsection. Although the three-layer ELM (ELM-M3) is more computationally expensive than the original ELM, it is significantly more efficient in terms of computational cost compared to ELM-M2, the unbalanced ELM, and the CNN-based approach (Figure 14). Notably, the computational cost of ELM-M3 is influenced by fingerprint quality, as its performance-optimizing hyperparameters are dependent on the SFINGE database, which accounts for variations in quality. These observations are supported by the number of neurons used in each artificial intelligence approach, as detailed in Table 5. As anticipated, learning time is directly proportional to the model's parameters. Considering the dual objectives of simplicity and performance, the proposed ELM-M3 emerges as the most viable alternative for fingerprint classification.

While deeper architectures like ELM-M3 may offer improvements in classification accuracy, it is important to consider the trade-offs associated with their increased complexity. For example, longer training times and a higher number of hidden neurons may result in higher inference times, especially in real-time systems. The ELM-M3 model employs 1250 hidden neurons distributed across three layers, which contributes to its improved performance. However, the increased complexity also comes with higher computational costs. This trade-off is particularly relevant when considering real-world applications, where inference time and system efficiency are critical. On the other hand, when comparing the results obtained by the ELM with three hidden layers with the work of Zabala et al. [13], an improvement of 0.034 in accuracy is observed. However, the penetration rate of the pro-

posed model is 0.0036 lower than that reported in [13]. This discrepancy can be attributed to differences in the learning approaches: the current work does not account for unbalanced learning where classes have varying probabilities of occurrence, whereas the approach proposed in [13] does. Furthermore, the optimization criterion in this study is based on raw accuracy, while that proposed in [13] optimizes performance using the geometric mean, considering class-specific averages. When comparing training times, the proposed ELM-M3 demonstrates substantial improvements, underscoring its computational efficiency. This is further validated by the number of hidden nodes in the architectures: the ELM-M3 employs 1250 hidden nodes ($N1 = 250$, $N2 = 250$, $N3 = 750$), whereas the model proposed in [13] utilizes 5000 hidden nodes. Thus, the reduced complexity of ELM-M3 offers a notable advantage in terms of computational cost.

Table 6 presents a comparison of testing times for the best proposed methodology and state-of-the-art approaches. Unfortunately, Peralta's manuscript does not report these times, and replicating his work is challenging due to the use of a high-performance computational architecture.

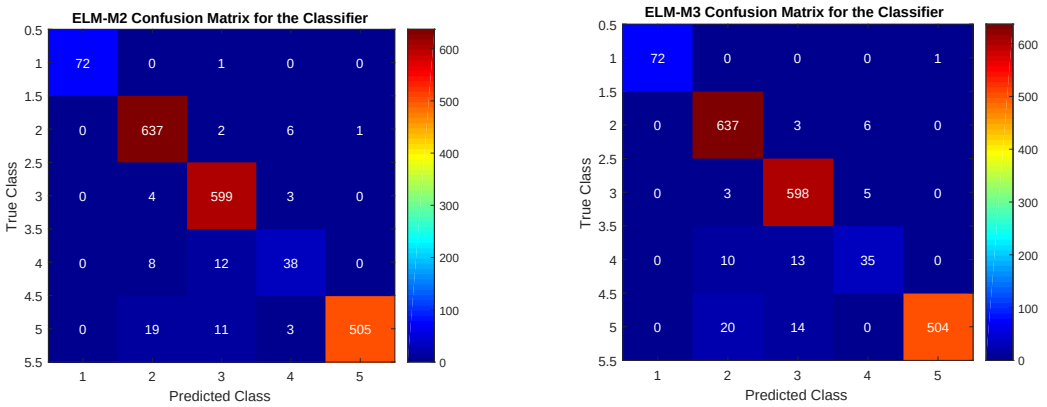

Figure 14. Confusion matrices for the ELM-M2 and ELM-M3 models.

Table 4. Evaluation of the computational times of the training of the best methodologies found in this work and other noteworthy approaches in the literature.

	Hong08 and Original ELM	**Hong08 and ELM-M2**	**Hong08 and ELM-M3**	**Zabala-Blanco20 [13]**	**Peralta18 [10]**
Default	1.0259 (±0.0633)	3.5320 (±0.2037)	2.5688 (±0.1205)	18.7819 (±0.3542)	957
HQNoPert	1.0545 (±0.0757)	3.6884 (±0.1506)	1.5860 (±0.0757)	18.8498 (±0.6013)	960
VQAndPert	1.0963 (±0.0531)	3.3228 (±0.1027)	3.0118 (±0.2172)	15.5062 (±0.3279)	960

Table 5. Number of total hidden neurons.

	Hong08 and Original ELM	**Hong08 and ELM-M2**	**Hong08 and ELM-M3**	**Zabala-Blanco20 [13]**	**Peralta18 [10]**
Default	2000 (N1 = 2000)	2000 (N1 = 1000 + N2 = 1000)	1500 (N1 = 250 + N2 = 250 + N3 = 1000)	5000 (N1 = 5000)	14,875
HQNoPert	2000 (N1 = 2000)	2000 (N1 = 1000 + N2 = 1000)	1250 (N1 = 250 + N2 = 250 + N3 = 750)	5000 (N1 = 5000)	14,875
VQAndPert	2000 (N1 = 2000)	2000 (N1 = 1000 + N2 = 1000)	2250 (N1 = 750 + N2 = 750 + N3 = 750)	5000 (N1 = 5000)	14,875

Table 6. Evaluation of the computational testing times of the best methodologies found in this work and other noteworthy approaches in the literature.

	Hong08 and ELM-M3	**Zabala-Blanco20 [13]**	**Peralta18 [10]**
HQNoPert	1.5827	2.6321	not reported

6. Conclusions and Future Works

The continuous growth of the human population has led to a corresponding increase in the size of fingerprint databases, resulting in billions of records and significantly slowing down fingerprint recognition processes. In this context, fingerprint classification serves as an effective preprocessing step to reduce the response times of biometric fingerprint systems. It is worth noting that fingerprint classification is a natural process, given that fingerprints can be categorized into five distinct classes. CNNs are currently the most accurate tools for fingerprint classification, achieving nearly 100% accuracy. However, their high computational cost and reliance on high-performance hardware make them impractical for widespread use. This manuscript introduces an alternative approach for fingerprint classification aimed at person recognition. The proposed method achieves competitive accuracy while minimizing computational costs, employing two- and three-hidden-layer ELMs, both based on the M-ELM architecture. These models significantly reduce training time—on the order of seconds—compared to state-of-the-art approaches reported in the literature. Additionally, computational resource requirements are substantially lowered, enabling the use of standard commercial machines without the need for high-performance GPUs. The results demonstrate that the three-layer ELM achieves the best accuracy while maintaining low computational costs and minimal training times. However, a slight decrease in TP rates is observed compared to the literature methods. The approach proposed in this study outperforms the unbalanced ELM described by Zabala et al. [13], both in terms of performance and complexity. The comparison with the method proposed by Zabala-Blanco et al. [13] is particularly relevant, as their method has shown excellent results on small- to medium-sized databases (1000 to 3000 samples).

In the literature, two commonly recognized databases for evaluating fingerprint verification algorithms are NIST [6,23,27,29,34,37–39] and FVC [1,7,23,27,33,54–58]. However, the NIST database presents significant limitations due to its balanced classes, making it unsuitable for real-world data scenarios; therefore, its use can be disregarded. On the other hand, the FVC database could be considered in future work to validate our results, although it also has challenges. Its main drawback is the limited number of samples (less than 800 per version), which reduces the robustness of experiments, and it does not address a five-class classification problem, an important aspect to consider when developing more complex verification models (to reduce the penetration rate).

Future work will focus on developing faster M-ELM approaches using numerical methods to optimize the computation of weights in the hidden and output layers of the ELM-AE. Heuristic optimization techniques will also be explored to fine tune all hyperparameters of the M-ELM, further improving performance and reducing computational costs. Additionally, testing will be conducted on larger databases emulating real-world scenarios such as the populations of countries like Chile (19 million), Spain (47 million), and Germany (83 million). Another promising avenue involves integrating the unbalanced learning approach of Zabala et al. [13] into the final layer of the M-ELM to maximize performance on unbalanced datasets. A comparative analysis of computational costs will also be carried out using consistent software and hardware configurations across different ELM models in order to provide irrefutable evidence of the advantages of M-ELM in terms of efficiency and scalability.

Author Contributions: Conceptualization, A.Q. and D.Z.-B.; methodology, A.Q. and D.Z.-B.; software, A.Q. and D.Z.-B.; formal analysis, A.Q. and D.Z.-B.; investigation, A.Q. and D.Z.-B.; writing—original draft preparation, A.Q. and D.Z.-B.; writing—review and editing, A.Q. and D.Z.-B.; project administration, D.Z.-B.; funding acquisition, D.Z.-B. 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 original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: This work was supported by the Laboratory of Technological Research in Pattern Recognition (LITRP) and FONDECYT REGULAR 2020 Nº 1200810 Very Large Fingerprint Classification based on a Fast and Distributed Extreme Learning Machine, Agencia Nacional de Investigación y Desarrollo, Ministerio de Ciencia, Tecnología, Conocimiento e Innovación, Gobierno de Chile.

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

Abbreviations

The following abbreviations are used in this manuscript:

MLP	Multilayer perceptron
SLFN	Single-hidden-layer feedforward neural network
ELM	Extreme learning machine
ELM-AE	Extreme learning machine autoencoder
M-ELM	Multilayer extreme learning machine
W-ELM	Unbalanced extreme learning machines
ELM-M3	Three-layer ELM
SVM	Support vector machine
CNN	Convolutional neural network
NIST	National Institute of Standards and Technologies
FVC	Fingerprint verification competition
OM	Orientation map
SFINGE	Synthetic fingerprint generator
RELU	Rectified linear unit
G-mean	Geometric mean
Exac	Root mean square error
PR	Absolute error of the penetration rate
OF	Orientation field
RF	Random forest
DB	Database
SGD	Stochastic gradient descent
CDF	Center-to-delta flow
MKL	Multi-space KL
CT	Curvelet transform
GLCM	Grey-level co-occurrence matrix
HQNoPert	High-quality no perturbations
VQandPert	Varying quality and perturbations
HOG	Histogram of oriented gradients
GUI	Graphic user interface
PolyU	Polytechnic University
LR	Logistic regression
5-FCV	Five-fold cross-validation
ROI	Singularity region of interest
DT	Decision tree
K-NN	K-nearest neighbors
MSE	Mean squared error
NIST-4	Standards and Technology Special Database 4

OM	Orientation map
MM	Minutiae map
OC	Orientation collinearity

References

1. Guo, J.M.; Liu, Y.F.; Chang, J.Y.; Lee, J.D. Fingerprint classification based on decision tree from singular points and orientation field. *Expert Syst. Appl.* **2014**, *41*, 752–764. [CrossRef]
2. Zabala-Blanco, D.; Hernández-García, R.; Barrientos, R.J.; Mora, M. Evaluation of the standard and regularized ELMs for gender and age classification based on palm vein images. In Proceedings of the 2021 40th International Conference of the Chilean Computer Science Society (SCCC), La Serena, Chile, 15–19 November 2021; IEEE: New York, NY, USA, 2021; pp. 1–8.
3. Galar, M.; Derrac, J.; Peralta, D.; Triguero, I.; Paternain, D.; Lopez-Molina, C.; García, S.; Benítez, J.M.; Pagola, M.; Barrenechea, E.; et al. A survey of fingerprint classification Part I: Taxonomies on feature extraction methods and learning models. *Knowl.-Based Syst.* **2015**, *81*, 76–97. [CrossRef]
4. Yaro, A.S.; Maly, F.; Prazak, P.; Maly, K. Enhancing Fingerprint Localization Accuracy With Inverse Weight-Normalized Context Similarity Coefficient-Based Fingerprint Similarity Metric. *IEEE Access* **2024**, *12*, 73642–73651. [CrossRef]
5. Shrein, J.M. Fingerprint classification using convolutional neural networks and ridge orientation images. In Proceedings of the 2017 IEEE Symposium Series on Computational Intelligence (SSCI), Honolulu, HI, USA, 27 Novembe–1 December 2017; IEEE: New York, NY, USA, 2017; pp. 1–8.
6. 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 (ICPR), Cancun, Mexico, 4–8 December 2016; IEEE: New York, NY, USA, 2016; pp. 931–936.
7. Nguyen, H.T.; Nguyen, L.T. Fingerprints classification through image analysis and machine learning method. *Algorithms* **2019**, *12*, 241. [CrossRef]
8. Abdul-Al, M.; Kyeremeh, G.K.; Qahwaji, R.; Ali, N.T.; Abd-Alhameed, R.A. A Novel Approach to Enhancing Multi-Modal Facial Recognition: Integrating Convolutional Neural Networks, Principal Component Analysis, and Sequential Neural Networks. *IEEE Access* **2024**, *12*, 140823–140846. [CrossRef]
9. Nsekuye, J.B.; Hawbett, M.M.; Elouadi, A. Resnet-F: Fingerprint Analysis Using Cnns for Type Identification and Ridge Quantification. *SSRN* **2024**, 4893120. [CrossRef]
10. 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]
11. El Hamdi, D.; Elouedi, I.; Fathallah, A.; Nguyen, M.K.; Hamouda, A. Fingerprint Classification Using Conic Radon Transform and Convolutional Neural Networks. In Proceedings of the International Conference on Advanced Concepts for Intelligent Vision Systems, 19th International Conference, ACIVS 2018, Poitiers, France, 24–27 September 2018; Springer: Cham, Switzerland, 2018; pp. 402–413.
12. Sureshkumar, V.; Prasad, R.S.N.; Balasubramaniam, S.; Jagannathan, D.; Daniel, J.; Dhanasekaran, S. Breast cancer detection and analytics using hybrid CNN and extreme learning machine. *J. Pers. Med.* **2024**, *14*, 792. [CrossRef]
13. 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]
14. Hong, J.H.; Min, J.K.; Cho, U.K.; Cho, S.B. Fingerprint classification using one-vs-all support vector machines dynamically ordered with naı¨ ve Bayes classifiers. *Pattern Recognit.* **2008**, *41*, 662–671. [CrossRef]
15. Xiao, D.; Li, B.; Mao, Y. A multiple hidden layers extreme learning machine method and its application. *Math. Probl. Eng.* **2017**, *2017*, 4670187. [CrossRef]
16. Huang, G.B.; Zhu, Q.Y.; Siew, C.K. Extreme learning machine: A new learning scheme of feedforward neural networks. In Proceedings of the 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541), Budapest, Hungary, 25–29 July 2004; IEEE: New York, NY, USA, 2004; Volume 2, pp. 985–990.
17. Huang, G.B.; Zhu, Q.Y.; Siew, C.K. Extreme learning machine: Theory and applications. *Neurocomputing* **2006**, *70*, 489–501. [CrossRef]
18. Tang, J.; Deng, C.; Huang, G.B. Extreme learning machine for multilayer perceptron. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *27*, 809–821. [CrossRef] [PubMed]
19. Agarwal, S.; Roth, D. Learning a sparse representation for object detection. In Proceedings of the European Conference on Computer Vision, Copenhagen, Denmark, 28–31 May 2002; Springer: Cham, Switzerland, 2002; pp. 113–127.
20. Agarwal, S.; Awan, A.; Roth, D. Learning to detect objects in images via a sparse, part-based representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 1475–1490. [CrossRef] [PubMed]

21. Kim, T.K.; Wong, S.F.; Cipolla, R. Tensor canonical correlation analysis for action classification. In Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 17–22 June 2007; IEEE: New York, NY, USA, 2007; pp. 1–8.
22. Kasun, L.L.C.; Zhou, H.; Huang, G.B.; Vong, C.M. Representational learning with extreme learning machine for big data. *IEEE Intell. Syst.* **2013**, *28*, 31–34.
23. Zia, T.; Ghafoor, M.; Tariq, S.A.; Taj, I.A. Robust fingerprint classification with Bayesian convolutional networks. *IET Image Process.* **2019**, *13*, 1280–1288. [CrossRef]
24. Cappelli, R.; Maio, D.; Maltoni, D. A multi-classifier approach to fingerprint classification. *Pattern Anal. Appl.* **2002**, *5*, 136–144. [CrossRef]
25. Liu, M. Fingerprint classification based on Adaboost learning from singularity features. *Pattern Recognit.* **2010**, *43*, 1062–1070. [CrossRef]
26. Henry, E.R. *Classification and Uses of Finger Prints*; HM Stationery Office, Printed by Darling and Son, Limited: London, UK, 1913.
27. Gal, Y.; Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Proceedings of the International Conference on Machine Learning, PMLR, New York, NY, USA, 20–22 June 2016; pp. 1050–1059.
28. Jian, W.; Zhou, Y.; Liu, H. Lightweight convolutional neural network based on singularity ROI for fingerprint classification. *IEEE Access* **2020**, *8*, 54554–54563. [CrossRef]
29. Militello, C.; Rundo, L.; Vitabile, S.; Conti, V. Fingerprint Classification Based on Deep Learning Approaches: Experimental Findings and Comparisons. *Symmetry* **2021**, *13*, 750. [CrossRef]
30. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
31. 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 IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
32. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
33. 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 (NCC), Riyadh, Saudi Arabia, 25–26 April 2018; IEEE: New York, NY, USA, 2018; pp. 1–5.
34. Zabala-Blanco, D.; Mora, M.; Hernández-García, R.; Barrientos, R.J. The Extreme Learning Machine Algorithm for Classifying Fingerprints. In Proceedings of the 2020 39th International Conference of the Chilean Computer Science Society (SCCC), Coquimbo, Chile, 16–20 November 2020; IEEE: New York, NY, USA, 2020; pp. 1–8.
35. 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 (ICCC), Chengdu, China, 13–16 December 2017; IEEE: New York, NY, USA, 2017; pp. 1942–1946.
36. Michelsanti, D.; Ene, A.D.; Guichi, Y.; Stef, R.; Nasrollahi, K.; Moeslund, T.B. Fast fingerprint classification with deep neural networks. In Proceedings of the International Conference on Computer Vision Theory and Applications, Porto, Portugal, 7 February–1 March 2017; SCITEPRESS Digital Library: Setúbal, Portugal, 2017; pp. 202–209.
37. Rajanna, U.; Erol, A.; Bebis, G. A comparative study on feature extraction for fingerprint classification and performance improvements using rank-level fusion. *Pattern Anal. Appl.* **2010**, *13*, 263–272. [CrossRef]
38. Cao, K.; Pang, L.; Liang, J.; Tian, J. Fingerprint classification by a hierarchical classifier. *Pattern Recognit.* **2013**, *46*, 3186–3197. [CrossRef]
39. Luo, J.; Song, D.; Xiu, C.; Geng, S.; Dong, T. Fingerprint classification combining curvelet transform and gray-level cooccurrence matrix. *Math. Probl. Eng.* **2014**, *2014*, 592928. [CrossRef]
40. Spanier, A.B.; Steiner, D.; Sahalo, N.; Abecassis, Y.; Ziv, D.; Hefetz, I.; Kimchi, S. Enhancing Fingerprint Forensics: A Comprehensive Study of Gender Classification Based on Advanced Data-Centric AI Approaches and Multi-Database Analysis. *Appl. Sci.* **2024**, *14*, 417. [CrossRef]
41. Galar, M.; Derrac, J.; Peralta, D.; Triguero, I.; Paternain, D.; Lopez-Molina, C.; García, S.; Benítez, J.M.; Pagola, M.; Barrenechea, E.; et al. A survey of fingerprint classification part II: Experimental analysis and ensemble proposal. *Knowl.-Based Syst.* **2015**, *81*, 98–116. [CrossRef]
42. Cappelli, R.; Lumini, A.; Maio, D.; Maltoni, D. Fingerprint classification by directional image partitioning. *IEEE Trans. Pattern Anal. Mach. Intell.* **1999**, *21*, 402–421. [CrossRef]
43. Cappelli, R.; Maltoni, D. Multispace KL for pattern representation and classification. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 977–996. [CrossRef]
44. Karu, K.; Jain, A.K. Fingerprint classification. *Pattern Recognit.* **1996**, *29*, 389–404. [CrossRef]

45. Zhang, Q.; Yan, H. Fingerprint classification based on extraction and analysis of singularities and pseudo ridges. *Pattern Recognit.* **2004**, *37*, 2233–2243. [CrossRef]
46. Liu, M.; Jiang, X.; Kot, A.C. Fingerprint reference-point detection. *EURASIP J. Adv. Signal Process.* **2005**, *2005*, 634350. [CrossRef]
47. Bazen, A.M.; Gerez, S.H. Systematic methods for the computation of the directional fields and singular points of fingerprints. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 905–919. [CrossRef]
48. Liu, M.; Jiang, X.; Kot, A.C. Fingerprint retrieval by complex filter responses. In Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06), Hong Kong, China, 20–24 August 2006; IEEE: New York, NY, USA, 2006; Volume 1, pp. 1042–1042.
49. Moreno-Torres, J.G.; Sáez, J.A.; Herrera, F. Study on the impact of partition-induced dataset shift on *k*-fold cross-validation. *IEEE Trans. Neural Netw. Learn. Syst.* **2012**, *23*, 1304–1312. [CrossRef] [PubMed]
50. Zabala-Blanco, D.; Mora, M.; Azurdia-Meza, C.A.; Firoozabadi, A.D.; Játiva, P.J.P.; Montejo-Sánchez, S. Multilayer extreme learning machine as equalizer in OFDM-based radio-over-fiber systems. *IEEE Lat. Am. Trans.* **2021**, *19*, 1790–1797. [CrossRef]
51. Wu, F.; Zhu, J.; Guo, X. Fingerprint pattern identification and classification approach based on convolutional neural networks. *Neural Comput. Appl.* **2020**, *32*, 5725–5734. [CrossRef]
52. Cappelli, R.; Maio, D.; Maltoni, D. Synthetic fingerprint-database generation. In Proceedings of the Object Recognition Supported by User Interaction for Service Robots, Quebec City, QC, Canada, 11–15 August 2002; IEEE: New York, NY, USA, 2002; Volume 3, pp. 744–747.
53. Watson, C.I.; Wilson, C.L. NIST special database 4. *Fingerpr. Database Natl. Inst. Stand. Technol.* **1992**, *17*, 5.
54. Maio, D.; Maltoni, D.; Cappelli, R.; Wayman, J.L.; Jain, A.K. FVC2000: Fingerprint verification competition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 402–412. [CrossRef]
55. Maio, D.; Maltoni, D.; Cappelli, R.; Wayman, J.L.; Jain, A.K. FVC2002: Second fingerprint verification competition. In Proceedings of the Object Recognition Supported by User Interaction for Service Robots, Quebec City, QC, Canada, 11–15 August 2002; IEEE: New York, NY, USA, 2002; Voume 3, pp. 811–814.
56. Maio, D.; Maltoni, D.; Cappelli, R.; Wayman, J.L.; Jain, A.K. FVC2004: Third fingerprint verification competition. In Proceedings of the International conference on biometric authentication, Hong Kong, China, 15–17 July 2004; Springer: Cham, Switzerland, 2004; pp. 1–7.
57. Cappelli, R.; Ferrara, M.; Franco, A.; Maltoni, D. Fingerprint verification competition 2006. *Biom. Technol. Today* **2007**, *15*, 7–9. [CrossRef]
58. Dorizzi, B.; Cappelli, R.; Ferrara, M.; Maio, D.; Maltoni, D.; Houmani, N.; Garcia-Salicetti, S.; Mayoue, A. Fingerprint and on-line signature verification competitions at ICB 2009. In Proceedings of the International Conference on Biometrics, Alghero, Italy, 2–5 June 2009; Springer: Cham, Switzerland, 2009; pp. 725–732.