



# Article Automatic Fingerprint Classification Using Deep Learning Technology (DeepFKTNet)

Fahman Saeed 🗅, Muhammad Hussain \*🗅 and Hatim A. Aboalsamh 🕩

Department of Computer Science, King Saud University, Riyadh 11451, Saudi Arabia; fahmanali@gmail.com (F.S.); hatim@ksu.edu.sa (H.A.A.)

\* Correspondence: mhussain@ksu.edu.sa

Abstract: Fingerprints are gaining in popularity, and fingerprint datasets are becoming increasingly large. They are often captured utilizing a variety of sensors embedded in smart devices such as mobile phones and personal computers. One of the primary issues with fingerprint recognition systems is their high processing complexity, which is exacerbated when they are gathered using several sensors. One way to address this issue is to categorize fingerprints in a database to condense the search space. Deep learning is effective in designing robust fingerprint classification methods. However, designing the architecture of a CNN model is a laborious and time-consuming task. We proposed a technique for automatically determining the architecture of a CNN model adaptive to fingerprint classification; it automatically determines the number of filters and the layers using Fukunaga–Koontz transform and the ratio of the between-class scatter to within-class scatter. It helps to design lightweight CNN models, which are efficient and speed up the fingerprint recognition process. The method was evaluated two public-domain benchmark datasets FingerPass and FVC2004 benchmark datasets, which contain noisy, low-quality fingerprints obtained using live scan devices and cross-sensor fingerprints. The designed models outperform the well-known pre-trained models and the state-of-the-art fingerprint classification techniques.

Keywords: multisensory fingerprint; interoperability; DeepFKTNet; deep learning; classification

MSC: 68T05

# 1. Introduction

A person can be recognized in security systems by a unique username and password, but they can be readily stolen [1]. The fingerprint is one of the first imaging modalities of biometric identification. It is more accurate and less expensive than other biometric modalities [2,3]. A fingerprint's surface has ridges and valleys, which do not change during a lifetime [4]. Fingerprint recognition can be used for authentication or identifying purposes. In verification, the fingerprint is compared to the templates of a particular subject in the database, but in identification, the unknown fingerprint is compared to the templates of all subjects in the database to ascertain the subject's identity [5]. Fingerprints are gaining in popularity and their datasets are becoming increasingly large. They are recorded utilizing a variety of low-cost embedded sensors in smart devices such as smartphones and computers. The high processing complexity of a fingerprint identification system is one of its primary drawbacks. One way to address this issue is to categorize fingerprints in a database to condense the search space. The existing classification methods are effective when fingerprints are recorded using the same sensor. However, when fingerprints are collected using various sensors (referred to as cross-sensor or sensor interoperability problem), classification performance is deteriorated; even verification of the same person's finger is degraded [6–8]. While considerable research has been conducted on cross-sensor fingerprint verification [8–12], there has been no study on cross-sensor fingerprint classification, which motivates us to work on this topic.



Citation: Saeed, F.; Hussain, M.; Aboalsamh, H.A. Automatic Fingerprint Classification Using Deep Learning Technology (DeepFKTNet). *Mathematics* 2022, *10*, 1285. https://doi.org/10.3390/ math10081285

Academic Editors: Florin Leon, Mircea Hulea and Marius Gavrilescu

Received: 27 February 2022 Accepted: 6 April 2022 Published: 12 April 2022

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



**Copyright:** © 2022 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/).

Numerous fingerprint categorization systems have been developed, some relying on non-conventional approaches and others on convolutional neural networks. The references provide an exhaustive overview of non-CNN methods [13,14]. The success rate of a fingerprint classification approach is highly dependent on the quality of the description of the discriminating information of a fingerprint. Directional ridge patterns and singularities are critical distinguishing characteristics of fingerprints, as demonstrated by the techniques proposed in [15–20], which utilize this information in a variety of ways to classify fingerprints. Gue et al. [15] employ the amount and kind of core points as fingerprint descriptors, as well as rule-based categorization, to classify fingerprints. Additionally, this approach classifies indistinguishable fingerprints using center-to-delta flow and balance arm flow. Its categorization accuracy is 92.7% on average. Jung and Lee [21] split a fingerprint into  $16 \times 16$  pixel blocks, compute their representative directions, use Markov models to identify the core block, and then divide the fingerprint into four areas, each of which is represented using distributions of ridge directional values. This method has a classification accuracy of 97.4%. Dorasamy et al. [17] employed a simplified rule-based technique and two features: directional patterns and singular points for fingerprint description. The classification accuracy of this scheme is 92.2%. Saeed et al. [18] proposed a modified histogram of oriented gradients (HOG) fingerprint classification algorithm. The HOG descriptor's orientation field computation is not ridge pattern specific. In order to improve the HOG descriptor's ability to represent a fingerprint, we compute an orientation field that is suited to the ridge pattern. This technique achieved an average accuracy of 98.70% on the noisy fingerprint database FVC2004. Saeed et al. [19] suggested a new approach for classifying noisy fingerprints from live scan devices using statistical features (mean, standard deviation, kurtosis and skewness) from dense scale invariant feature transform (d-SIFT). This method achieved 97.6% accuracy using FVC2004, a noisy, low-quality live scanned fingerprint database. Sudhir et al. [22] employed GLCM, LBP, and SURF for feature extraction, while SVM and BoF classifiers were used for classification. Based on FVC2004, they got average accuracy of 74.50 using SVM and 84.75 using BoF.

Deep CNN has shown remarkable results in many applications [23–26]; it has been used to classify fingerprints [27–32] and has achieved encouraging results. Zia et al. [33] introduced the Bayesian DCNNs (B-DCNNs) by incorporating Bayesian model uncertainty to increase fingerprint categorization accuracy. They achieved 95.3% accuracy on FVC004 (5 class), showing a 0.8–1.0% improvement in model accuracy compared to the baseline DCNN. In Nguyen et al. [34], the CNN approach is suggested for the noise reduction stage of noisy fingerprint. Two main steps are involved in this procedure. Non-local information is used to construct a pre-processing phase for noisy image. Fingerprints are then separated into patches and utilized for CNN training, resulting in a model for CNN de-noising of future noisy images, which can subsequently be smoothed using Gaussian filtering to remove pixel artifacts. Fingerprints that have been pre-processed are separated into overlapping patches during the CNN training step. To train the convolutional neural network, they feed these patches into it. They've built a three-tiered network with distinct filters and operators at each level. Third layer convolutional layer predicts enhancing patches and reconstructs the output image. Using the Gaussian algorithm and a canny algorithm they strength the information edge, this approach is able to filter out noise. When all images have been processed by the morphological procedure, the result will be improved. They extracted features from pre-processed fingerprints (arch, loop and whorl) and classified them using for classifiers: random forest, SVM, CNN, and K-NN and obtained accuracies of 97.78%, 95.83%, 96.11%, and 92.05%, respectively.

Nahar et al. [35] designed CNN models based on the LeNet-5 design for fingerprint classification. They evaluated their method using the augmented subset (DB1) from the FVC2004 dataset. They got an accuracy of 99.1%. In deep models, layers and filters are defined by experiments, and no special rule is used to choose them; tuning the hyperparameters is tiring and time-consuming. Motivated by the difficulty in the design of CNN architectures, we propose a technique that determines automatically and adaptively

the architecture of a CNN model using the fingerprints dataset. To begin, we use the LGDBP description Saeed, et al. [36] and K-medoid clustering algorithm [37] to choose representative fingerprints, and then we derive the layers filters using Fukunaga–Koontz Transform (FKT) [38]. To control the depth of a CNN model, we compute the ratio between traces of between-class scatter matrix  $S_b$  and within-class scatter matrix  $S_w$ .

The proposed fingerprint CNN classification system was evaluated against the stateof-the-art fingerprint classification schemes utilizing the benchmark multi-sensor datasets FingerPass and FVC2004. Specifically, the contributions of this work are as follows:

- We developed an efficient automatic method for classifying cross-sensor fingerprints based on a CNN model.
- We proposed a technique for the custom-designed building of a CNN model, which automatically determines the architecture of the model using the class discriminative information from fingerprints. The layers and their respective filters of an adaptive CNN model are customized using FKT, and the ratio of the traces of the between-class scatter matrix, and the within-class scatter matrix.
- We thoroughly evaluated the proposed method on two datasets. The proposed fingerprint classification scheme is quick, accurate, and performs well with noisy fingerprints obtained using live scan devices as well as cross-sensor fingerprints.

The rest of the paper is organized as follows. Section 2 presents the details of the proposed technique. The experimental results have been given in Section 3. Section 4 discusses the performance of the proposed method in detail. Section 5 concludes the article.

#### 2. Proposed Method

The convolutional neural network (CNN) is one of the most widely used and popular deep learning networks [39]. Its general structure comprises different types of layers, including the CONV layer with different filters, pooling layer, activation function layer, fully connected layer, and loss function [40]. It has been used for a wide range of tasks, including image and video recognition [41], classification of images [42], medical image analysis [43], computer vision [44], and natural language processing [45].

Many advancements in CNN learning methods and architecture have a place, allowing the network to handle larger, diverse, more complicated, and multiclass issues [46]. Following AlexNet's outstanding performance on the ImageNet dataset in 2012, many applications used CNNs [47]. A layer-wise representation of CNN reversed the trend toward extraction of features at low spatial resolution in deep architecture, as achieved in VGG [48]. Most modern architectures follow VGG's simple and homogeneous topology idea. The Google deep learning group introduced the divide, transform, and merge concept with the inception block. The inception block introduced the concept of branching within a layer, allowing for feature abstraction at various spatial scales [49]. Skip connections, developed by ResNet [50] for deep CNN training, gained popularity in 2015. Others, like Wide ResNet, are exploring the influence of multilevel transformations on CNN's learning capacity by increasing cardinality or widening the network [51]. So, the research turned from parameter optimization to network architecture design. Thus, new architectural concepts like channel boosting, spatial and feature-map exploitation, and attention-based information processing emerged [52]. The main issue in the design of CNN models is to tune the architecture of CNN for a specific application.

## 2.1. Problem Formulation

The fingerprints are categorized into four types: arch, left loop, right loop, and whorl. Identifying the type of a fingerprint is a multiclass classification problem. Let there be *N* subjects, and *K* fingerprints are captured from each subject with *M* different sensors; these fingerprints are categorized into *C* classes. Let  $\mathcal{F} = \left\{ F_{ij}^s \middle| 1 \le i \le K, 1 \le j \le N, 1 \le s \le M \right\}$ , where  $F_{ij}^s$  represents the *i*th fingerprint of the *j*th subject captured with *s*th sensor, be the set of fingerprints, and  $\mathbb{C} = \{1, 2, ..., C\}$ , where *C* is the number of classes, be the set of

fingerprint labels (classes). The problem of predicting the type of a fingerprint  $F_{ij}^s$  is to build a function  $\psi : \mathcal{F} \to \mathbb{C}$  that takes a fingerprint  $F_{ij}^s \in \mathcal{F}$  and assigns it a label  $c \in \mathbb{C}$ , i.e.,  $\psi(F_{ij}^s; \theta) = c$ , where  $\theta$  are the parameters. We design the function  $\psi$  using a CNN model, in this case  $\theta$  represents the weights and biases of the model. The model is built adaptively. Its design process is shown in Figure 1, and the detail is given in the rest of the section.



**Figure 1.** Design procedure of DeepFKTNet; (**a**) design of main DeepFKTNet architecture and (**b**) addition of global pooling and softmax layers and fine-tuning the model.

## 2.2. Adaptive CNN Model

The main constituent of a CNN model is a convolutional (CONV) layer. It extracts discriminative features from the input signal, applying convolution operation with filters of fixed size. CONV layers are stacked in a CNN model to extract a hierarchy of features. The number of filters in each CONV layer and the number of CONV layers in a CNN model are hyper-parameters, and finding the best configuration of a model for a specific application is a hard optimization problem; it entails the search of huge parameter space. In addition, the initialization of learnable parameters of a CNN model has a significant effect on the performance of the model when it is trained with an iterative optimization algorithm like Adam optimizer. Leveraging the discriminative content of fingerprints, we propose a simple method to find the best configuration of the model adaptively. Initially, we select the representative fingerprints from each type to guide the design process of a CNN model. The discriminative information in these fingerprints is used to determine the width (the number of filters) of each CONV layer and the depth (the number of CONV layers) of the model; it is also used for data-dependent initialization of the filters of CONV

layers. An overview of the design process is shown in Figure 1. We employ clustering to select the representative fingerprints, the Fukunaga–Koontz Transform (FKT) [38], which exploits class-discriminative information, to determine the number of filters in a CONV layer, and the ratio of the between-class scatter matrix  $S_b$  to the within-class scatter matrix  $S_w$  to adjust the depth (i.e., the number of CONV layers) of the CNN model. Finally, to minimize the number of learnable parameters and avoid overfitting, global pooling layers are introduced. By decreasing the resolution of the feature maps, the pooling layer seeks to achieve shift-invariance, and the pooling layer's feature map is linked directly to SoftMax [53]. The design process is worked out in detail and discussed in the following subsections, and its overview is shown in Figure 1.

#### 2.2.1. Selection of Representative Fingerprints

We extract discriminative information from fingerprints to specify the CONV layers and the depth of a CNN model adaptively. To do this, we cluster the training set to identify the most representative fingerprints of each class. For determining the representative fingerprints, discriminative features from fingerprints are extracted using the LGDBP descriptor [36] K-medoids [37] is used for clustering since it selects the instances as cluster centers and is suitable for finding the representative subset of the training set. The fingerprints corresponding to the cluster centers are chosen as the representative subset. The number of clusters for each class in the K-medoids algorithm is specified using the silhouette analysis [54]. Using this procedure, we select the set  $X = \{X_1, X_2, \ldots, X_C\}$ , where  $X_i = \{RF_{j}, j = 1, 2, 3, \ldots, n_i\}$  is the set of representative fingerprints of *i*th class.

#### 2.2.2. Design of the Main DeepFKTNet Architecture

The architectures of the state-of-the-art CNN models are usually not drawn from the data and are fixed and highly complex. On the contrary, we define a data-dependent architecture of DeepFKTNet. Its primary architecture is based on the answers to two questions: (i) how many CONV layers should be in the model and (ii) how many filters must be in each layer. These questions are addressed by an iterative algorithm that computes the number of filters in a CONV layer, adds it iteratively to the model, and terminates when a criterion is satisfied. We use the discriminative structural information embedded in fingerprints to determine the number of filters in a CONV layer and their initialization. The detail is given in Algorithm 1. We discuss the algorithm with motivation in the following paragraphs.

Initially, the set  $X = \{X_1, X_2, ..., X_C\}$  is used to determine the number of filters of the first CONV layer and initialize them. Inspired by the filter size of the first CONV layer in the state-of-the-art CNN models like ResNet [50], DenseNet [55], and Inception [49], we fixed the size of filter size of the first layer to  $7 \times 7$ . We extract patches of size  $w \times h$  from the representative fingerprints (steps 2–3 of Algorithm 1) and formulate the problem of determining the filters ( $f_i$ , i = 1, 2, ..., N) as finding the optimal projection direction vectors  $u_i$ , i = 1, 2, ..., d, which are determined by solving the following optimization problem:

$$U^* = \arg \max_{U} \frac{tr(U^T S_b U)}{tr(U^T S_w U)}$$
(1)

where  $S_b$  and  $S_w$  are the between-class and within-class scatter matrices (as computed in step 4 of the Algorithm 1). According to Fukunaga Koontz Discriminant Analysis (FKT) [38], the optimal projection direction vectors  $u_i$  are the eigenvectors of  $\hat{S}_b$  i.e.,

$$\hat{S}_b u = \lambda u \tag{2}$$

where  $\hat{S}_b = P^T S_b P$ ,  $P = Q D^{-1/2}$  and Q & D are obtained by the diagonalization of the sum  $S_b + S_w$  i.e.,  $S_b + S_w = Q D Q^T$  (steps 5–6 of Algorithm 1). The Equation (2) gives the optimal vectors, which simultaneously maximize  $tr(U^T S_b U)$  and minimize  $tr(U^T S_w U)$ . Unlike Linear Discriminant Analysis (LDA) [56], the inversion of  $S_w$  is not needed in this approach,

so it can tackle very high-dimensional data. Additionally, this approach seeks to find optimal vectors that are orthogonal. As the dimension of the patch vectors  $b_i$  related to the intermediate CONV layers is usually very high, and we need filters that are independent, so this approach is suitable for our design process. The problem of selecting the number of filters in the convolutional layer is to select the eigenvectors  $u_k$ , k = 1, 2, ..., L so that the ratio  $\gamma_k = \frac{Trace(SF_b)}{Trace(SF_w)}$  attains maximum value. Here the between-class scatter matrix  $SF_b$  and within-class matrix  $SF_w$  are computed for each  $u_k$  by projecting all activations  $a_j^i$  in the space spanned by  $u_k$  (steps 7–8 of the Algorithm 1). It ensures to select the filters which extract discriminative features. After selecting  $u_k$ , k = 1, 2, ..., L, the CONV block with L filters  $f_k$ , k = 1, 2, ..., L initialized with  $u_k$  is introduced in DeepFKTNet. Then, a pooling layer is added if needed (step 8–10 of the Algorithm 1).

Using the current architecture of DeepFKTNet, the set of activations  $Z = \{Z_1, Z_2, ..., Z_C\}$  of  $X = \{X_1, X_2, ..., X_C\}$  is computed. These activations are used to determine whether to add more layers to the net. It is decided by calculating the trace ratio  $TR = \frac{Trace(S'_b)}{Trace(S'_w)}$ , where  $S'_b$  and  $S'_w$  are the between-class and within-class scatter matrices of the activations Z. If TR is greater than the previous TR (PTR), it means that the addition of the current block of layers introduced the discriminative potential to the network. This criterion ensures that the features generated by DeepFKTNet have large inter-class variation and small intra-class scatter. To add another CONV block, the steps 3–10 are repeated with Z. To reduce the size of feature maps for computational effectiveness, pooling layers are added after the first and second CONV blocks. As the kernels and their number are determined from the fingerprint images, each layer can have a different number of filters.

It is to be noted that the eigenvector  $u_k$ , which are used to specify the kernels of a CONV layer, have the maximum  $\gamma_k$  and capture most of the variability in input fingerprint images without redundancy in the form of independent features. The depth of a CNN model (number of layers) and the number of kernels for each layer are important factors that determine the model complexity. Step 7 of Algorithm 1 determines the best kernels that ensure the preservation of maximum energy of the input image, and step 8 initializes these kernels to be suitable for the fingerprint domain. The selected kernels extract the features from fingerprint images so that the variability of the structures in fingerprint images is maximality preserved. It is also important that the features must be discriminative (i.e., have large inter-class variance and small intra-class scatter as we go deeper in the network). It is ensured using the trace ration  $TR = \frac{Trace(Sb)}{Trace(Sw)}$ , the larger the value of the trace ratio, the larger the inter-class variance and the smaller the intra-class scatter [57]. Step 11 in Algorithm 1 allows adding CONV layers as long as TR is increasing and determines the data-dependent depth of DeepFKTNet, as shown in Figure 2.

Algorithm 1: Design of the main DeepFKTNet Architecture **Input:** The set  $X = \{X_1, X_2, \dots, X_C\}$ , where  $X_i = \{RF_i, j = 1, 2, 3, \dots, n_i\}$  is the set of representative fingerprints of *i*th class. Output: The main DeepFKTNet Architecture. Initialize DeepFKTNet with input layer and set w = 7, h = 7, d = 1, and m (the number of filters) = 0 for the first layer; Step 1: PTR (previous TR) = 0. For i = 1, 2, 3, ..., CStep 2: Compute  $Z_i = \{a_i^i = RF_i, \text{ for each } RF_i \in X_i\}$ Step 3: For i = 1, 2, 3, ..., C $A_i = \emptyset$ For each  $a_i^i \in Z_i$ Extract patches  $p_1^l, p_2^l, \ldots, p_m^l$  of size  $w \times h$  with stride 1 from  $a_i^i$ , vectorize them into vectors of dimension  $D = w \times h \times d$  and append to  $A_i$ . Using  $A = [A_1, A_2, \ldots, A_C]$ , compute Step 4: -between-class scatter matrix  $S_b = \sum_{i=1}^{C} \left(\frac{1}{n_i}A_iJ_i - \frac{1}{n}AJ\right) \left(\frac{1}{n_i}A_iJ_i - \frac{1}{n}AJ\right)^T$ , where  $J_i$  is an  $n_i \times n_i$  matrix with all ones. -within-class scatter matrices  $S_w = \sum_{i=1}^{C} \left(A_i - \frac{1}{n_i}A_iJ_i\right) \left(A_i - \frac{1}{n_i}A_iJ_i\right)^T$ Diagonalize the sum  $\Sigma = S_b + S_w$  i.e.,  $\Sigma = QDQ^T$  and transform the scatter matrices Step 5: using the transform matrix  $P = QD^{-\frac{1}{2}}$ . i.e.,  $\hat{S}_{h} = P^{T}S_{h}P$ ,  $\hat{S}_{w} = P^{T}S_{w}P$ . Step 6: Compute eigenvectors  $u_k$ , k = 1, 2, ..., D of  $\hat{S}_b$  such that  $\hat{S}_b u = \lambda u$ For each eigenvector  $u_k$ , k = 1, 2, ..., DStep 7: -Reshape  $u_k$  to a filter  $f_k$  of size  $w \times h \times d$ -Compute  $Y = \{Y_1, Y_2, ..., Y_C\}$ , where  $Y_i = \{f_k * a_i^i, j = 1, 2, ..., n_i\}$ -Compute the between scatter matrix  $SF_b$  and within scatter matrix  $SF_w$  from Y. -Compute the trace ratio  $\gamma_k = \frac{Trace(SF_b)}{Trace(SF_w)}$ Select *L* filters  $f_k$ , k = 1, 2, ..., L corresponding to  $\gamma_k > 0$  (as shown in Figure 2 for layer 1). Step 8: Step 9: Add the CONV block to DeepFKTNet with filters  $f_k$ , k = 1, 2, ..., L. Update m = m + 1. If m = 1 or 2, add a max pool layer with pooling operation of size 2  $\times$  2 and stride 2 to Step 10: Deep FKTNet. Step 11: Compute  $Z = \{Z_1, Z_2, ..., Z_C\}$ , where  $Z_i = \{a_i^i = \text{DeepFKTNet}(RF_i), \text{ for each } RF_i \in X_i\}$ Using  $Z = \{Z_1, Z_2, ..., Z_C\}$ , compute the ratio  $TR = \frac{Trace(S'_b)}{Trace(S'_w)}$ Step 12: If  $PTR \leq TR$ , set PTR = TR, w = 3, h = 3, d = L and go to Step 3, otherwise stop.



Figure 2. Selection of best filters for layer1 of DeepFKTNet model for FingerPass dataset.

## 2.3. Addition of Global Pool and Softmax Layers

Activation of the last CONV block is with dimension  $h \times w \times L$ , and after flattening, it is fed to FC layers; the number of parameters is huge and leads to overfitting. To reduce the number of parameters and spatial dimensions of the last CONV block activation, we

feed it to global average pooling (GAP) and global max-pooling (GMP) layers [58]. The GAP average all the *hw* values, whereas the GMP takes into account the contributions of the neurons of maximum response; the number of neurons in the FC layer is  $h \times w \times L$ , and it is reduced to  $1 \times 1 \times L$  when only GMP or GAP is introduced. We concatenate the output of GMP and GAP layers to overcome the shortcoming of each and then feed it to the FC layer, followed by the SoftMax layer.

### 2.4. Fine-Tuning the Model

The DeepFKTNet model is evaluated using the challenge multisensory FingerPass dataset [59], and it is compared to the well-known deep models: ResNet [50] and DenseNet [55] pre-trained on the ImageNet dataset and fine-tuned using the same dataset as DeepFKTNet. For further validation, we evaluated our method using the challenge FVC2004 dataset [60] and compared it to the state-of-the-art methods. For each dataset, we select the most representative fingerprint images from the training set using K-medoids and LGDBP descriptor and then built its adaptive DeepFKTNet architecture using Algorithm 1.

#### 2.4.1. Datasets and the Adaptive Architectures

To verify the performance of the DeepFKTNet model on benchmark datasets, we used FingerPass and FVC2004 datasets. The FingerPass is a multi-sensor dataset; it was collected using nine different optical and capacitive sensors and two interaction types, i.e., press and sweep. The FingerPass contains a total of fingers separated into nine subsets based on sensors; each subset contains 12 impressions of 8 fingers from 90 persons.

FVC2004 dataset contains noisy images acquired by live scan devices. It has 4 sets: DB1 collected using optical V300 sensor, DB2 collected using optical U 4000, DB3 collected using thermal sweeping sensor, and DB4 is a synthetic fingerprint dataset. Each one contains 880 fingerprint images [60]. We categorized FVC2004 fingerprints into four categories: arch, left loop, right loop, and whorl. We merge the 4 sets of FVC2004 into one set of four classes; it is now a multi-sensor fingerprint dataset.

To setup best parameters for each DeepFKTNet model, the hyperparameter optimization software framework Optuna [61] is used to select the best hyperparameters for fine-tuning the DeepFKTNet model. Using Algorithms 1, the DeepFKTNet architecture obtained for the FVC2004 dataset consists of 5 CONV blocks, as shown in Figure 3a, whereas the architecture constructed for the FingerPass dataset has11 blocks, as depicted in Figure 3b. The number of filters for each CONV block and the depth of each model for each fingerprint dataset are determined using Algorithm 1. Using the Optuna optimization algorithm, we fine-tuned the hyperparameters and tested three optimizers (Adam, SGD, and RMSprop), learning rate between  $1 \times 10^{-1}$ , and  $1 \times 10^{-5}$ , patch size (5, 10, 15, 20, 30, 50), activation functions (Relu, LRelu, and Sigmoid), and dropout between 0.25 and 0.50. After training for 10 epochs, the best hyper-parameters for each dataset are shown in Table 1.

Dataset	Activate Function	Learning Rate	Pach's Size	Optimizer	Dropout
FingerPass	Relu	0.0005	16	RMSprop	0.45
FVC2004	Relu	0.0008	10	RMSprop	0.38

**Table 1.** The optimized hyperparameters using Optuna algorithm.



Figure 3. (a) FVC2004 FKTNET architecture. (b) Fingerprint FKTNET architecture.

## 2.4.2. Evaluation Procedure

For evaluation, we manually separated the FingerPass dataset into four classes (arch, left loop, right loop, and whorl). We divided the FingerPass dataset into three sets (80% training, 10% validation, and 10% testing) using two different scenarios. In scenario-1, the fingers from each sensor were divided into training, validation, and test sets. In scenario-2, fingers in the training, validation, and test sets are from different sensors.

For the FVC2004 dataset, we divided the dataset into training (80%), validation (10%), and testing (10%), keeping the balance. For performance evaluation, we used four commonly used metrics: accuracy (ACC), true positive rate (TPR), true negative rate (TNR), and Kappa [62–65]. The overall average of metrics has been computed. The used metrics [66,67] to evaluate the proposed system are:

$$ACC = \frac{TP + TN}{TP + FP + TN + FN}$$
(3)

$$TPR = \frac{TP}{TP + FN} \tag{4}$$

$$TNR = \frac{TN}{TN + FP} \tag{5}$$

$$Kappa = \frac{P_0 - P_e}{1 - P_e} \tag{6}$$

where *TP*, *TN*, *FP*, and *FN* are the numbers are true positives, true negatives, false positives, and false negatives;  $P_0$  and  $P_e$  are calculated from the confusion matrix; the detail is given in [68]. To compute *TP*, *TN*, *FP*, and *FN*, one class, in turn, is taken as positive, the other classes are assumed to be negative, and the *TPR* and *TNR* are calculated. Finally, mean *TPR* and *TNR* are calculated by averaging *TPR* and *TNR* over all classes. In the results, the mean *TPR* and *TNR* are reported.

#### 3. Experimental Results

This section presents the experimental results of the DeepFKTNet models designed for the two datasets.

We designed the DeepFKTNet model for each dataset and fine-tuned it using the training sets. We validated its performance on FingerPass and FVC2004 datasets and compared it with the widely used CNN models ResNet [50] and DenseNet [55], which were pre-trained on the ImageNet dataset and fine-tuned on the same training set that was used for the DeepFKTNet model. In the rest of the paper, we name the DeepFKTNet models as

DeepFKTNet-11 and DeepFKTNet-5, designed for the FingerPass and the FV2004 datasets, respectively.

The results of the three models DeepFKTNet-11, ResNet152, and DenseNet121 for scenario-1 are shown in Figure 4a and Table 2a. The DeepFKTNet-11 model generated adaptively on the FingerPass dataset outperforms the state-of-the-art ResNet152 and DenseNet121 models in terms of all metrics. Though DenseNet121 is not better than DeepFKTNet-11, it outperforms ResNet152 in terms of all metrics. Figure 4b and Table 2b show the results for scenario-2 on the FingerPass dataset. In this scenario, the results obtained with the DeepFKTNet-11 are almost similar to those obtained in scenario-1. The DeepFKTNet-11 outperforms ResNet152 and DenseNet121. Figure 5 illustrates the confusion matrices for both scenarios. These give insights into the system performance for different classes.



**Figure 4.** Comparison between FKTNET-11 and pre-trained ResNet-152 and DensNet-121 on Fingerprint dataset (4 classes) using scenario 1 (**a**) and scenario 2 (**b**).

Table 2. Comparison between FKTNF	T-11 and pre-trained	ResNet-152 and Dens	Net-121 on Finger-
print dataset scenario 1 (a) and scenar	io 2 ( <b>b</b> ).		

(a)					
	ACC%	SE%	SP%	AUC%	Kappa%
FKTNet-11	97.84	93.25	98.28	95.21	93.05
ResNet152	91.22	78.22	92.05	86.11	80.32
DensNet121	93.55	80.22	94.44	87.55	82.11
(b)					
	ACC%	SE%	SP%	AUC%	Kappa%
FKTNet-11	98.9	93.6	98.5	96.12	93.93
ResNet152	92.22	80.22	93.05	86.5	81.62
DensNet121	94.85	84.22	96.12	90.21	84.55



Figure 5. Confusion matrix based on FKTNET-11 model for scenario 1 and scenario 2.

The DeepFKTNet-5 model was adaptively designed for the challenge FVC2004 dataset; it was evaluated using the above evaluation procedure. We fine-tuned the developed DeepFKTNet-5 model and the pre-trained models ResNet152 and DenseNet121 using the same dataset. The results are shown in Figure 6; the DeepFKTNet-5 model outperforms the state-of-the-art ResNet152 and DenseNet121 models in terms of all metrics. Figure 7 illustrates the confusion matrices for the FVC2004 dataset. These give insights into the system performance for different classes.



**Figure 6.** Comparison between FKTNET-5 and pre-trained ResNet-152 and DensNet-121 on FVC2004 dataset (four classes).



Figure 7. Confusion matrix based on FKTNET-5 model for FVC2004 dataset.

### 4. Discussions

We addressed the multi-sensor fingerprint classification problem and proposed a novel method for automatically generating a custom-designed DeepFKTNet model from the target fingerprint dataset. The number of layers and filters for each layer are not specified randomly; they are determined from the best representative fingerprints selected using the K-medoids clustering algorithm and LDGBP descriptor from the fingerprint datasets.

The generated DeepFKTNet models are shallower than the state-of-the-art models, robust, involve a small number of learnable parameters, and suitable for fingerprint classification.

The results of the DeepFKTNet models on the FingerPass and FVC2004 datasets (Figures 4 and 6) indicate that they outperform the famous deep models ResNet152 and DenseNet121, which were pre-trained on the ImageNet dataset and fine-tuned using the same fingerprint datasets. The architecture of a DeepFKTNet model is drawn directly from the dataset; the internal structures of the data determine its design. For this reason, the DeepFKTNet model has a compact size and yields better classification results. Further, it does not suffer from the overfitting problem (see Table 3) since it involves a small number of learnable parameters (see Table 4), which is comparable with the number of training examples. If the number of learnable parameters is huge as compared to the training examples, the overfitting problem cannot be avoided. The training and testing accuracies shown in Table 3 indicate that the models do not suffer from overfitting. In addition, DeepFKTNet models are trained using the available training data, and the pre-training is not needed, unlike ResNet152 and DenseNet121.

Table 3. The train and test accuracy of DeepFKTNet-11 models for two scenarios.

Model	Train ACC	Test ACC
Scenarios 1	98.65	97.84
Scenarios 2	99.11	98.9

**Table 4.** The comparison between generated DeepFKTNet models from the two datasets and pretrained ResNet152 and DenseNet121. K is for kilobyte and G is for Gigabyte.

Model	DeepFKTNet-5	DeepFKTNet-11	ResNet152	DenseNet121
number of params	58.456 k	119.599 k	60.19 M	7.98 M
FLOPs 0.5 G 0.9	0.9 G	5.6 G	1.44 G	

The space complexity of a CNN model is measured in terms of the number of learnable parameters, whereas the number of FLOPS determines its time complexity. Table 4 gives the statistics of the space and time complexities of the models. Overall, the DeepFKTNet model got competitive performance with fewer layers and parameters. The DeepFKTNet models designed for the two datasets have a small number of parameters, in thousands against millions in ResNet152 and DensNet121 models. DeepFKTNet-5 and DeepFKTNet-11 have fewer FLOPs than ResNet152 and DensNet121 and better performance. The DeepFKTNet-11 is relatively more complex than DeepFKTNet-5; the reason is that the FingerPass dataset involves a large number of sensors as compared to the FVC2004 dataset, and there is more variety of patterns in the FingerPass dataset, and to encode the discriminative pattern, more rich structure is needed.

Further, for investigating which features the DeepFKTNet models focus on for decision making, we employed GradCam [69]. Figure 8 shows some heat maps generated with GradCam for DeepFKTNet-11. The fingerprint images from class arches and their GradCam visualizations are shown in Figure 8a,b, the fingerprint images from the class left loop and their GradCam visualizations are shown in Figure 8c,d. Figure 8e,f depicts fingerprint images from the class right loop and their GradCam visualization, whereas Figure 8g,h show fingerprint images from the class whorls and their GradCam visualizations. The visual analysis of the decision-making process of DeepFKTNet shows that it concentrates on the discriminative regions of fingerprints and extracts class discriminative features.



**Figure 8.** Visualizations of activation maps using the GradCam method for four samples from different classes of FingerPass dataset: (**a**) arches finger; (**b**) arches's gradcam; (**c**) left loop finger; (**d**) left loop's gradcam; (**e**) right loop finger; (**f**) right loop's gradcam; (**g**) whorls finger; and (**h**) whorls gradcam.

For a fair comparison, the DeepFKTNet-5 has been compared with the state-of-the-art fingerprint classification methods, which were validated on the benchmark public FVC2004 dataset; the comparison results are given in Table 5.

The DeepFKTNet-5 model outperforms the state-of-the-art methods (handcraft and CNN methods) on the same dataset in terms of accuracy. The method of Jeon et al. [70], despite being a complex ensemble of CNN models, got an accuracy of 97.2%, which is less than that of DeepFKTNet-5. Zia et al. [33] employed B-DCNNs with five convolution layers and two FC layers (with 1024 and 512 neurons) for fingerprint classification and validated on the FVC2004 dataset; it does not yield better accuracy than that of DeepFKTNet-5 (95.3% vs. 98.89%). Its complexity is high; it has more FLOPs (0.65 G vs. 0.5 G) and more learnable parameters (38.66 M vs. 58.456 k). Nguyen et al. [34] employed a two-stage CNN model for enhancing and then training and prediction. They used LBCNN [71] method in the first stage, which has 0.352 M learnable parameters, and then employed a three-ternary model for training and prediction. They got an accuracy of 96.1% based on FVC2004 (three classes), which is less than DeepFKTNet-5. Nahar et al. [35] used a modified LNet-5 model for fingerprint classification; they got 99.1% accuracy but with only a subset (DB1) from FVC2004, whereas the DeepFKTNet-5 model evaluated on the combined multi-sensor dataset of the four datasets (DB1, DB2, DB3, and DB4) from FVC2004. Also, the LNet-5 has a higher number of parameters, 19.25 M and 1.42 G FLOPs vs. 58.456 k and 0.5 G FLOPs of DeepFKTNet-5. The reason for the better performance and less complexity of DeepFKTNet-5 is that it is custom-designed, keeping in view the internal discriminative structures of fingerprints.

Paper	Mathad		Performa	Performance (%)			
Taper	Metnod	ACC SE SP K		Kappa			
Gupta et al. [72] 2015	Singular point	97.80	-	-	-		
Darlow et al. [73] 2017	Minutiae and DL	94.55	-	-	-		
Andono et al. [74] 2018	Bag-of-Visual-Words	90	-	-	-		
Saeed et al. [19] 2018	statistics of D-SIFT descriptor	97.40	-	-	-		
Saeed et al. [18] 2018	Modified HOG descriptor	98.70	-	-	-		
Jeon et al. [70] 2017	Ensemble CNN model	97.2	-	-	-		
Zia et al. [33] 2019	B-DCNNs	95.3					
Nguyen et al. [34] 2019	CNN (tested on 3 classes of FVC2004)	96.1					
Nahar et al. [35] 2022	Modified LeNet (tested on FVC2004-DB1)	99.1					
DeepFKTNet-5	DeepFKTNet model	98.89	95.46	99.18	96.82		

Table 5. Comparison between DeepFKTNet-5 and the state-of-the-art methods.

# 5. Conclusions

We introduced a technique for automatically creating a custom-designed CNN model for multi-sensor fingerprint categorization. Since CNN models contain a large number of parameters and are designed randomly, we used the FKT approach to build a low-cost, high-speed CNN model tailored for the target fingerprint dataset. The developed DeepFK-TNet model is data-dependent, with a distinctive architecture for each fingerprint dataset. DeepFKTNet-11 for the FigerPass dataset and DeepFKTNet-5 for FVC2004 outperform pre-trained deep ResNet152 and DenseNet121 models on identical datasets and assessment processes. The performance, complexity, and number of parameters of the DeepFKTNet models created are substantially fewer than those of ResNet152 and DenseNet. Compared to the state-of-the-art techniques on the FVC2004 dataset, the DeepFKTNet-5 model is simpler in terms of complexity and parameter count and achieves comparable performance. In future work, we will enhance DeepFKTNet to address the problem of cross-sensor fingerprint verification.

**Author Contributions:** Conceptualization, F.S. and M.H.; methodology, M.H. and F.S.; software, F.S.; validation, F.S., M.H. and H.A.A.; formal analysis, H.A.A. and M.H.; investigation F.S., M.H.; resources, F.S. and H.A.A.; data curation, F.S., M.H.; writing—original draft preparation, F.S.; writing—review and editing, M.H.; visualization, H.A.A.; supervision, M.H.; project administration, M.H.; funding acquisition, M.H and H.A.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** This Project was funded by the National Plan for Science, Technology and Innovation (MAARIFAH), King Abdulaziz City for Science and Technology, Kingdom of Saudi Arabia, under Project no. 13-INF946-02.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

**Data Availability Statement:** Used public domain datasets, FVC2004 dataset: available online: http://bias.csr.unibo.it/fvc2004/download.asp (accessed on 26 February 2022).

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

## References

- Grabatin, M.; Steinke, M.; Pöhn, D.; Hommel, W. A Matrix for Systematic Selection of Authentication Mechanisms in Challenging Healthcare related Environments. In Proceedings of the 2021 ACM Workshop on Secure and Trustworthy Cyber-Physical Systems, Virtually, TN, USA, 28 April 2021; pp. 88–97.
- Maltoni, D.; Maio, D.; Jain, A.K.; Prabhakar, S. Handbook of Fingerprint Recognition; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2009.
- Pandey, F.; Dash, P.; Samanta, D.; Sarma, M. ASRA: Automatic singular value decomposition-based robust fingerprint image alignment. *Multimed. Tools Appl.* 2021, 80, 15647–15675. [CrossRef]

- 4. Khosroshahi, M.E.; Woll-Morison, V. Visualization and fluorescence spectroscopy of fingerprints on glass slide using combined 405 nm laser and phase contrast microscope. *J. Vis.* **2021**, *24*, 665–670. [CrossRef]
- Banik, A.; Ghosh, K.; Patil, U.K.; Gayen, S. Identification of molecular fingerprints of natural products for the inhibition of breast cancer resistance protein (BCRP). *Phytomedicine* 2021, *85*, 153523. [CrossRef] [PubMed]
- Lugini, L.; Marasco, E.; Cukic, B.; Gashi, I. Interoperability in fingerprint recognition: A large-scale empirical study. In Proceedings of the 2013 43rd Annual IEEE/IFIP Conference on Dependable Systems and Networks Workshop (DSN-W), Budapest, Hungary, 24–27 June 2013; pp. 1–6.
- Alrashidi, A.; Alotaibi, A.; Hussain, M.; AlShehri, H.; AboAlSamh, H.A.; Bebis, G. Cross-Sensor Fingerprint Matching Using Siamese Network and Adversarial Learning. *Sensors* 2021, 21, 3657. [CrossRef]
- Priesnitz, J.; Rathgeb, C.; Buchmann, N.; Busch, C.; Margraf, M. An overview of touchless 2D fingerprint recognition. EURASIP J. Image Video Process. 2021, 2021, 1–28. [CrossRef]
- AlShehri, H.; Hussain, M.; AboAlSamh, H.; AlZuair, M. A large-scale study of fingerprint matching systems for sensor interoperability problem. Sensors 2018, 18, 1008. [CrossRef]
- Alshehri, H.; Hussain, M.; Aboalsamh, H.A.; Emad-Ul-Haq, Q.; Alzuair, M.; Azmi, A.M. Alignment-free cross-sensor fingerprint matching based on the co-occurrence of ridge orientations and Gabor-HoG descriptor. *IEEE Access* 2019, 7, 86436–86452. [CrossRef]
- Marasco, E.; Feldman, A.; Romine, K.R. Enhancing Optical Cross-Sensor Fingerprint Matching Using Local Textural Features. In Proceedings of the 2018 IEEE Winter Applications of Computer Vision Workshops (WACVW), Lake Tahoe, NV, USA, 15 March 2018; pp. 37–43.
- 12. Lin, C.; Kumar, A. A CNN-based framework for comparison of contactless to contact-based fingerprints. *IEEE Trans. Inf. Forensics Secur.* 2018, 14, 662–676. [CrossRef]
- Galar, M.; Derrac, J.; Peralta, D.; Triguero, I.; Paternain, D.; Lopez-Molina, C.; García, S.; Benítez, J.M.; Pagola, M.; Barrenechea, E. A survey of fingerprint classification Part I: Taxonomies on feature extraction methods and learning models. *Knowl.-Based Syst.* 2015, *81*, 76–97. [CrossRef]
- Galar, M.; Derrac, J.; Peralta, D.; Triguero, I.; Paternain, D.; Lopez-Molina, C.; García, S.; Benítez, J.M.; Pagola, M.; Barrenechea, E. A survey of fingerprint classification Part II: Experimental analysis and ensemble proposal. *Knowl.-Based Syst.* 2015, *81*, 98–116. [CrossRef]
- 15. 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]
- 16. Bhalerao, B.V.; Manza, R.R. Development of Image Enhancement and the Feature Extraction Techniques on Rural Fingerprint Images to Improve the Recognition and the Authentication Rate. *IOSR J. Comput. Eng.* **2013**, *15*, 1–5.
- Dorasamy, K.; Webb, L.; Tapamo, J.; Khanyile, N.P. Fingerprint classification using a simplified rule-set based on directional patterns and singularity features. In Proceedings of the 2015 International Conference on Biometrics (ICB), Phuket, Thailand, 19–22 May 2015; pp. 400–407.
- 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; pp. 1–5.
- 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 (ICCAIS), Riyadh, Saudi Arabia, 4–6 April 2018; pp. 1–4.
- Dhaneshwar, R.; Kaur, M.; Kaur, M. An investigation of latent fingerprinting techniques. *Egypt. J. Forensic Sci.* 2021, 11, 1–15. [CrossRef]
- Jung, H.-W.; Lee, J.-H. Noisy and incomplete fingerprint classification using local ridge distribution models. *Pattern Recognit.* 2015, 48, 473–484. [CrossRef]
- Vegad, S.; Shah, Z. Fingerprint Image Classification. In Data Science and Intelligent Applications; Springer: Berlin/Heidelberg, Germany, 2021; pp. 545–552.
- 23. Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, G.; Cai, J. Recent advances in convolutional neural networks. *Pattern Recognit.* 2018, 77, 354–377. [CrossRef]
- 24. Grigorescu, S.; Trasnea, B.; Cocias, T.; Macesanu, G. A survey of deep learning techniques for autonomous driving. *J. Field Robot.* **2020**, *37*, 362–386. [CrossRef]
- Abou Arkoub, S.; El Hassani, A.H.; Lauri, F.; Hajjar, M.; Daya, B.; Hecquet, S.; Aubry, S. Survey on Deep Learning Techniques for Medical Imaging Application Area. In *Machine Learning Paradigms*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 149–189.
- 26. Dong, S.; Wang, P.; Abbas, K. A survey on deep learning and its applications. *Comput. Sci. Rev.* 2021, 40, 100379. [CrossRef]
- 27. Mishra, A.; Dehuri, S. An experimental study of filter bank approach and biogeography-based optimized ANN in fingerprint classification. In *Nanoelectronics, Circuits and Communication Systems;* Springer: Berlin/Heidelberg, Germany, 2019; pp. 229–237.
- Jian, W.; Zhou, Y.; Liu, H. Lightweight Convolutional Neural Network Based on Singularity ROI for Fingerprint Classification. IEEE Access 2020, 8, 54554–54563. [CrossRef]
- Nahar, P.; Tanwani, S.; Chaudhari, N.S. Fingerprint classification using deep neural network model resnet50. *Int. J. Res. Anal. Rev.* 2018, 5, 1521–1537.

- 30. Rim, B.; Kim, J.; Hong, M. Fingerprint classification using deep learning approach. Multimed. Tools Appl. 2020, 1–17. [CrossRef]
- 31. Ali, S.F.; Khan, M.A.; Aslam, A.S. Fingerprint matching, spoof and liveness detection: Classification and literature review. *Front. Comput. Sci.* **2021**, *15*, 1–18. [CrossRef]
- 32. Bolhasani, H.; Mohseni, M.; Rahmani, A.M. Deep learning applications for IoT in health care: A systematic review. *Inform. Med. Unlocked* **2021**, *23*, 100550. [CrossRef]
- 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]
- Nguyen, H.T.; Nguyen, L.T. Fingerprints classification through image analysis and machine learning method. *Algorithms* 2019, 12, 241. [CrossRef]
- 35. Nahar, P.; Chaudhari, N.S.; Tanwani, S.K. Fingerprint classification system using CNN. *Multimed. Tools Appl.* **2022**, 1–13. [CrossRef]
- 36. Saeed, F.; Hussain, M.; Aboalsamh, H.A. Method for Fingerprint Classification. U.S. Patent 9,530,042, 13 June 2016.
- 37. Zhang, Q.; Couloigner, I. A new and efficient k-medoid algorithm for spatial clustering. In Proceedings of the International Conference on Computational Science and Its Applications, Singapore, 9–12 May 2005; pp. 181–189.
- 38. Huo, X. A statistical analysis of Fukunaga-Koontz transform. IEEE Signal Process. Lett. 2004, 11, 123–126. [CrossRef]
- 39. Dhillon, A.; Verma, G.K. Convolutional neural network: A review of models, methodologies and applications to object detection. *Prog. Artif. Intell.* **2020**, *9*, 85–112. [CrossRef]
- Alzubaidi, L.; Zhang, J.; Humaidi, A.J.; Al-Dujaili, A.; Duan, Y.; Al-Shamma, O.; Santamaría, J.; Fadhel, M.A.; Al-Amidie, M.; Farhan, L. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *J. Big Data* 2021, *8*, 1–74. [CrossRef]
- 41. Abdullah, S.M.S.A.; Ameen, S.Y.A.; Sadeeq, M.A.; Zeebaree, S. Multimodal emotion recognition using deep learning. J. Appl. Sci. Technol. Trends 2021, 2, 52–58. [CrossRef]
- Jena, B.; Saxena, S.; Nayak, G.K.; Saba, L.; Sharma, N.; Suri, J.S. Artificial intelligence-based hybrid deep learning models for image classification: The first narrative review. *Comput. Biol. Med.* 2021, 137, 104803. [CrossRef]
- 43. Lu, J.; Tan, L.; Jiang, H. Review on convolutional neural network (CNN) applied to plant leaf disease classification. *Agriculture* **2021**, *11*, 707. [CrossRef]
- 44. Fang, W.; Love, P.E.; Luo, H.; Ding, L. Computer vision for behaviour-based safety in construction: A review and future directions. *Adv. Eng. Inform.* **2020**, *43*, 100980. [CrossRef]
- Lavanya, P.; Sasikala, E. Deep learning techniques on text classification using Natural language processing (NLP) in social healthcare network: A comprehensive survey. In Proceedings of the 2021 3rd International Conference on Signal Processing and Communication (ICPSC), Coimbatore, India, 13–14 May 2021; pp. 603–609.
- 46. Khan, A.; Sohail, A.; Zahoora, U.; Qureshi, A.S. A survey of the recent architectures of deep convolutional neural networks. *Artif. Intell. Rev.* **2020**, *53*, 5455–5516. [CrossRef]
- Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* 2012, 25. [CrossRef]
- 48. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. arXiv 2014, arXiv:1409.1556.
- 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.
- 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.
- 51. Hamel, P.; Eck, D. Learning features from music audio with deep belief networks. In Proceedings of the ISMIR, Utrecht, The Netherlands, 9–13 August 2010; pp. 339–344.
- 52. Khan, A.; Sohail, A.; Ali, A. A new channel boosted convolutional neural network using transfer learning. *arXiv* 2018, arXiv:1804.08528.
- 53. Lin, M.; Chen, Q.; Yan, S. Network in network. *arXiv* 2013, arXiv:1312.4400.
- 54. Rousseeuw, P.J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **1987**, 20, 53–65. [CrossRef]
- 55. Huang, G.; Liu, Z.; Weinberger, K.; van der Maaten, L. Densely connected convolutional networks. CVPR 2017. *arXiv* 2016, arXiv:1608.06993.
- Izenman, A.J. Linear discriminant analysis. In Modern Multivariate Statistical Techniques; Springer: Berlin/Heidelberg, Germany, 2013; pp. 237–280.
- Mika, S.; Ratsch, G.; Weston, J.; Scholkopf, B.; Mullers, K.-R. Fisher discriminant analysis with kernels. In Proceedings of the Neural Networks for Signal Processing IX: Proceedings of the 1999 IEEE Signal Processing Society Workshop (Cat. No. 98th8468), Madison, WI, USA, 25 August 1999; pp. 41–48.
- Cook, A. Global Average Pooling Layers for Object Localization. 2017. Available online: https://alexisbcook.github.io/2017 /globalaverage-poolinglayers-for-object-localization/ (accessed on 19 August 2019).
- Jia, X.; Yang, X.; Zang, Y.; Zhang, N.; Tian, J. A cross-device matching fingerprint database from multi-type sensors. In Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), Tsukuba, Japan, 11–15 November 2012; pp. 3001–3004.

- 60. 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: Berlin/Heidelberg, Germany, 2004; pp. 1–7.
- Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna: A next-generation hyperparameter optimization framework. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 2623–2631.
- 62. Gao, Z.; Li, J.; Guo, J.; Chen, Y.; Yi, Z.; Zhong, J. Diagnosis of Diabetic Retinopathy Using Deep Neural Networks. *IEEE Access* 2019, 7, 3360–3370. [CrossRef]
- 63. Quellec, G.; Charrière, K.; Boudi, Y.; Cochener, B.; Lamard, M. Deep image mining for diabetic retinopathy screening. *Med. Image Anal.* 2017, 39, 178–193. [CrossRef] [PubMed]
- 64. Chowdhury, A.R.; Chatterjee, T.; Banerjee, S. A Random Forest classifier-based approach in the detection of abnormalities in the retina. *Med. Biol. Eng. Comput.* **2019**, *57*, 193–203. [CrossRef]
- Zhang, W.; Zhong, J.; Yang, S.; Gao, Z.; Hu, J.; Chen, Y.; Yi, Z. Automated identification and grading system of diabetic retinopathy using deep neural networks. *Knowl. -Based Syst.* 2019, 175, 12–25. [CrossRef]
- Haghighi, S.; Jasemi, M.; Hessabi, S.; Zolanvari, A. PyCM: Multiclass confusion matrix library in Python. J. Open Source Softw. 2018, 3, 729. [CrossRef]
- 67. Powers, D.M. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv* 2011, arXiv:2010.16061.
- 68. Fleiss, J.L.; Cohen, J.; Everitt, B.S. Large sample standard errors of kappa and weighted kappa. *Psychol. Bull.* **1969**, *72*, 323. [CrossRef]
- Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 618–626.
- Jeon, W.-S.; Rhee, S.-Y. Fingerprint pattern classification using convolution neural network. Int. J. Fuzzy Log. Intell. Syst. 2017, 17, 170–176. [CrossRef]
- Juefei-Xu, F.; Naresh Boddeti, V.; Savvides, M. Local binary convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 19–28.
- 72. Gupta, P.; Gupta, P. A robust singular point detection algorithm. Appl. Soft Comput. 2015, 29, 411–423. [CrossRef]
- Darlow, L.N.; Rosman, B. Fingerprint minutiae extraction using deep learning. In Proceedings of the 2017 IEEE International Joint Conference on Biometrics (IJCB), Denver, CO, USA, 1–4 October 2017; pp. 22–30.
- 74. Andono, P.; Supriyanto, C. Bag-of-visual-words model for fingerprint classification. Int. Arab J. Inf. Technol. 2018, 15, 37-43.