ANALYSIS OF FINGERPRINT MATCHING PERFORMANCE WITH DEEP NEURAL NETWORKS

A Thesis Submitted to
the Graduate School of Engineering and Sciences of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

in Computer Engineering

by
Alper GÖÇEN

March 2022
İZMİR

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to *Nesli ERDOĞMUŞ* for her support and patience. I also thank all my friends and family members who have supported me during my master's education.

ABSTRACT

ANALYSIS OF FINGERPRINT MATCHING PERFORMANCE WITH DEEP NEURAL NETWORKS

Fingerprints are unique biometric properties for each person. In the literature and industry, they are widely used for identification purposes. Collecting biometric datasets is a tedious work since it is not possible without the owners' consent, and existing fingerprint datasets are either not sufficient to use in deep learning tasks by means of size or most of them are kept private to the collectors' use. This increases the need of synthetic fingerprint images and their use in a variety of tasks especially for training deep learning models.

In this study, the performance of a CNN architecture named Finger ConvNet[1] is compared to well-known networks and the question of whether a mixed dataset consisting of synthetically generated and real fingerprint images can reach a performance close or equal to ones having only real images is discussed.

As a result of experiments, it is shown that the number of real images in the dataset is an important factor and that the performance of the mixed dataset was less than the one having only real images proposed in the referred study[1].

ÖZET

DERİN SİNİR AĞLARI İLE PARMAK İZİ EŞLEŞTİRME PERFORMANSI ANALİZİ

Parmak izleri her kişi için benzersiz biyometrik özelliklerdir. Literatürde ve endüstride kimlik belirleme amacıyla yaygın olarak kullanılmışlardır. Biyometrik veri kümesi oluşturmak veri sahibinin izni olmadan yapılamadığından ve varolan veri kümelerinin derin öğrenme yöntemleri için yeterli olmaması, toplayan kişi ya da kurumun özel kullanımı için oluşturulması gibi sebeplerden ötürü zorlu bir iştir. Bu, sentetik parmak izi resimlerinin ve onların özellikle derin öğrenme gibi çeşitli problemlerde kullanımının önemini arttırmıştır.

Bu çalışmada, Finger Convnet[1] isimli bir sınıflandırıcı derin öğrenme modelinin performansı, literatürdeki iyi bilinen modellerle karşılaştırılmış, sentetik ve gerçek veri karışımından oluşan veri kümesinin performansının, yalnızca gerçek verilerden oluşanlara yakın ya da eşit olup olamayacağı sorusu tartışılmıştır.

Deneylerin sonucu olarak, karma veri kümesinin içindeki gerçek resim sayısının belirleyici bir faktör olduğu ve performansın referans çalışmadaki[1] sadece gerçek veri içeren veri kümesinden daha az olduğu görülmüştür.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Fingerprint recognition is a problem that has been in the literature for several decades and has become a topic of information technologies and computer science as the developments in automated systems, computer vision and artificial intelligence has advanced. A fingerprint is a graphical ridge pattern present in human fingers which with its uniqueness and permanence is one of the most reliable biometric identification characteristics. In history, a lot of methods have been applied to recognize humans by their fingerprints. The research started with manual methods such as marking the unique points in the fingerprint by human intervention and comparing the match percentage to decide how much two fingerprints are alike. Despite the intelligent methods applied in history devised to augment the performance and accuracy of fingerprint recognition, the increasing needs of the field became overwhelming.

After increasing demands, researchers started working on automating the fingerprint identification process and identifying the key visual features in fingerprints that make them unique, which is still the governing approach today. The features extracted from a fingerprint image are classified under 3 main groups: Level-1, Level-2, and Level-3 features from the least to the most detailed respectively. Level-1 features consist of the global ridge orientations and mostly used for fingerprint classification. Level-2 features are minutiae points which are ridge endings and bifurcations. There are other types of minutiae mentioned in the literature but these are the most reliably working features. Level-3 features consist of very detailed features such as pores, ridge contours, etc.

In this study, fingerprint features are acquired by training a CNN, and using it as a feature extractor. These features are then used to train a binary classifier to complete an overall fingerprint matching architecture. The CNN architecture is trained by a mixture of real and synthetic data to investigate if synthetically generated fingerprint data can be used for training when the real data at hand are not sufficient.

1.1 Motivation

Constructing a fingerprint dataset is a cumbersome work like any other biometric data. Collecting huge numbers of fingerprint images takes a lot of time and sometimes it is even impossible because of personal/cultural reasons. There are some organizations collecting this kind of datasets, however, most of them keep them private to their own use.

In this study, leveraging [2][3], we construct a dataset which is constituted by a small number of real fingerprint images which belong to human beings, and a large number of synthetically generated fingerprint images to analyze if a deep learning model trained by this dataset can reach a performance close or equal to the ones trained by completely real datasets. The generation of synthetic images is not part of this study and images generated using [25] are used. To compare the performances of the networks used in this work, the work in [1] is taken as a reference point and the networks named Finger ConvNet and Resnet18 are used to test the constructed mixed dataset.

1.1.1 Contributions

In this study, the architecture of the first CNN named Finger ConvNet is inspired by [1], in which they compared this network's performance with well-known CNN architectures such as VGG-16 and ResNet. However, they used around 100.000 images for training these architectures and, as mentioned before, collecting a dataset of this size may not always be possible. We leverage synthetic fingerprint generation methods and generate more than 90% of the dataset of size 10.980 synthetically and analyze whether or not a similar performance can be acquired with this mixed data.

While using a smaller dataset which consists only of the real data, the classification performance of the Finger ConvNet and Resnet18 deteriorated, and making the architectures simpler did not help improve the performance. We show in this work that the synthetic fingerprint generators can be used to construct a larger dataset to train deep learning models with sufficient performance when more data are needed.

1.2 Fingerprint Features

After 2012 when a neural network, specifically a CNN, won the ImageNet competition [4], deep neural networks' effects on visual recognition tasks immensely increased. In this work, we will specifically focus on feature extraction and making use of those features with a specific deep neural network architecture to recognize the visual pattern in the fingerprints.

Before starting to explain what will be done in this work, we find explaining what feature extraction is, and how we will extract the related features from a fingerprint image very valuable. After this, we will continue explaining the neural network architecture that will be used.

As mentioned previously there are Level-1, Level-2 and Level-3 features in a fingerprint image. Extracting and comparing these features, especially Level-2 features, help distinguish fingerprint images from each other. Level-2 features are named minutiae points, and they are a special type of ridge feature, where a ridge ends or constructs a structure called bifurcation. These features turned out to be unique features that distinguish a fingerprint from one another a few decades ago. Since then, minutiae are features which many researchers worked with in order to increase the accuracy and performance of the fingerprint recognition task. In Figure 1.1 an example of ridge ending and a bifurcation is shown:

(a) Ridge Ending (b) Bifurcation

Figure 1.1: Fingerprint level-2 features

Feature extraction operation is detecting and putting an image's discriminative features in a usable format. In this work, we use CNN architectures as feature extractors since they are very powerful at detecting and extracting visual features out of images. We use an architecture which consists of a CNN named Finger-ConvNet and a NN classifier proposed in [1].

To compare the performance of the mentioned network with another well known CNN architecture, we chose Resnet18 to train with the same setup. This network is also

put in the overall setup as a classifier with its pre-trained Imagenet[4] weights. On top of the model itself, a few fully-connected layers are added and these layers are fine-tuned for the model to learn fingerprint specific features. The performance analysis of these three networks will be discussed in the following chapters.

1.3 Thesis Outline

This thesis consists of five chapters.

In chapter 2, the literature survey on the use of CNNs for fingerprint classification and matching is given.

In chapter 3, the methodology for dataset cleaning, training and performance evaluation of CNN and binary classifier models are explained.

In chapter 4, Experimental results and their comparison to the original work is given.

In chapter 5, Conclusion and possible ways to enlarge this study are discussed.

CHAPTER 2

RELATED WORK

Traditional fingerprint recognition steps generally apply image acquisition, preprocessing, feature extraction, and matching steps. Some algorithms like [5][6] need the extraction of minutia to catch the most number of correspondences between them. Some other algorithms like [7] leverage the ridge information which consists of some chosen set of minutiae for pre-alignment to minimize the risk of misalignment of the two fingerprints. Since the amount of fingerprint data is increasing day by day, the speed of feature extraction of classic matching algorithms will not be sufficiently fast and the performance of the algorithms will deteriorate.

Artificial neural networks (ANNs) have always been used as fingerprint classifiers in the literature. [8] proposed various models of multilayer perceptrons to deal with the classification problem. Mascinska and Tyma [9] used Kahonen's Self Organizing Maps (SOMs), which have been used in other works as well. ANNs have also been used as components in classification architectures like in [10][11] (Probabilistic ANNs + Structural models).

Before deep neural networks started to output the state-of-the-art results, SVMs (Support Vector Machines), which is a well-performing classification technique for problems with two classes, had been the dominating approach in classification problems. A modification on the SVM structure were needed, since fingerprint classification is a multiclass classification problem. Decomposition techniques used in [12][13] was one of the approaches to deal with this problem. In [14] Yao et al. employed SVMs with error correction mechanisms. This approach continued in [15] incorporating RNNs to the multiclassifier. Some recent approaches with SVMs like [16][17] use Genetic Programming to cleanse the features that will be used in the classification.

Ghaddab et al.[5] proposed Expanded Delaunay Triangularization (EDT-C) which is a new fingerprint matching architecture based on minutiae triplets. The introduced matcher leverages an extended form of Delaunay triangularization and optimizes the process that allows it to consolidate local matchings in a fast way. Peralta et al. [6] proposed an approach to filter spurious minutiae to increase the matching algorithm's efficiency

Today, deep learning based solutions got significant attention and did a very successful job in computer vision problems, such as face recognition, object detection, and etc. Since CNNs specifically have a very good performance on this kind of problems,

they are also used in fingerprint recognition. In [18], a minutiae extraction system based on CNNs was proposed which does not require any preprocessing steps before extraction. In that work, a dataset with overlapping patches from 200 labeled fingerprint images were chosen. Patches are firstly processed by a network called JudgeNet to detect patches that have exactly one minutiae point. Then the exact position of the minutia was calculated by the further post-processing steps. A precision value of 94.59%, a recall value of 91.63%, and an F1-score of 93.09% were reported. In another work [19], a network called FingerNet was proposed, which included some domain-specific knowledge in the neural network architecture (orientation mapping, segmentation, etc.) and resulted in better results compared to the state-of-the-art on the NIST SD27 [20] and FVC 2002 [21] datasets. These datasets were used by another work named MinutiaeNet which consists of two subnetworks named CoarseNet and FineNet where CoarseNet takes whole fingerprint images as well as orientation map, enhanced image, and segmentation map and generates a minutia score map. FineNet processes each candidate patch, a square region, whose center is the candidate minutiae point, to refine the orientation score map.

Siamese networks are also used in fingerprint matching tasks. A siamese network is composed of two identical feature extractor architectures and perform the matching task by using semantic similarities in deeper layers. In [22], the authors used a siamese network named SiameseFinger to perform cross-sensor fingerprint matching. The network takes the features extracted by the Gabor-HoG descriptor and they got a comparable result with that of the state-of-the-art architectures using FingerPass and MOLF datasets. In [23], a multi-siamese network is used to match contactless and contact-based fingerprint images by using additional handcrafted fingerprint features like minutiae, core point and etc. Their proposed architecture got a 8.39% EER result which outperforms the state-of-the-art architectures, and compared to the other CNN architectures such as VGG, Triplet CNN and also minutiae-based methods [24] on a publicly available dataset named PolyU.

In [1], which this study is inspired by, a CNN classifier namely Finger Convnet is proposed. This architecture outperformed other well-known CNN architectures such as VGG-16 and ResNet on fingerprint matching problem. They achieved around a 98% validation accuracy on the classifier and NN classifier and 98% of test accuracy on fingerprint matching task on Ten-Finger Card and ID Card Data datasets which is better than the other networks' performances.

CHAPTER 3

METHODOLOGY

3.1 Dataset Preparation

The overall dataset consists of both real and synthetic fingerprint images. It has 480 aligned fingerprint images from the FVC2002 dataset. The FVC2002 part is divided into 60 classes each having 8 impressions. The rest of the dataset is constituted by synthetic fingerprint images. In this study the synthetic fingerprint images generated by [25] are used and the generation process was not done in the scope of this work. The alignment process is completed using the singular points and orientation map. All the images in the dataset are enhanced leveraging oriented Gabor filters, since the synthetic fingerprint images need a cleaning process. Enhancement operation enabled us to clean them, since it makes the ridge structure way more visible and clean. After the enhancement operation, the synthetic part of the dataset is cleaned by eliminating noisy images, since they increase the inner-class variance of the dataset and add unnecessary noise.

(a) Real image from FVC2002 (b) Synthetically generated image

Figure 3.1: Images from different subsets

3.1.1 Alignment of Synthetic Images

Since deep learning models learn fingerprint images by interpreting their ridge structure, we need to make sure that the classes in our dataset are aligned with each other. We use an alignment algorithm that makes use of the singular points and orientation maps of the fingerprint images. Since we already know the singular points and orientation maps, we do not take the step in which they are found into consideration. The algorithm steps are as follows:

1. Cut the image as a right and left half vertically from the core point.
2. Calculate the average orientation angle of the left and right halves.
3. Rotate the fingerprint image so that the difference between the two average angles become the minimum
4. Translate the image so that the core point becomes the image center

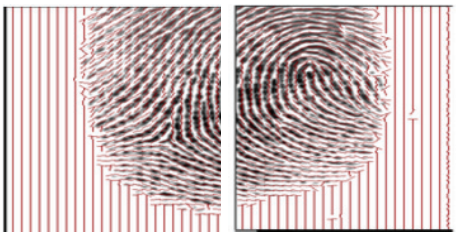
(a) Before alignment (b) After alignment

Figure 3.2: Alignment operation results

3.1.2 Image Enhancement

The synthetically generated fingerprint images had some samples that had unclear and noisy ridge structures because of a problem in the generation operation. For some of the images the ridge structures could not be constructed properly, and they needed to be eliminated, since they added unnecessary noise and increased the inner-class variation in the dataset. All the images having unclear or incomplete ridge structures were removed from the dataset.

(a) Before alignment



(a) Left orientation map (b) Right orientation map

(a) Aligned Image

Figure 3.5: Steps of the alignment process

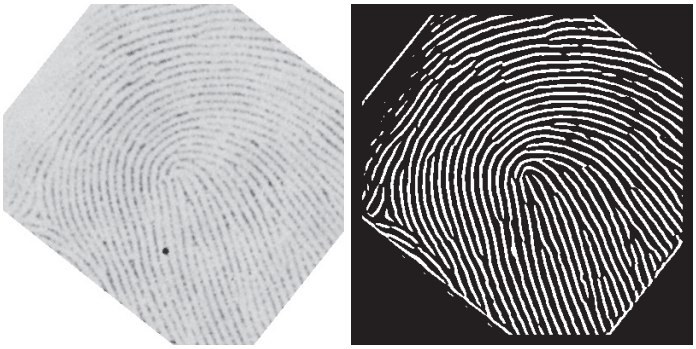
Since in the beginning, the synthetic images' ridge structures were not so clear, an enhancement operation using oriented Gabor filters was applied on the images. The orientation of the Gabor filters are based on the orientation of the ridges. For the synthetic images we have the orientation map of the images, therefore there is not a separate calculation step. After having the orientation image, the ridge frequency image and region masks are calculated as proposed in [26]. After these calculation steps a bank of Gabor filters is applied on the images as a last step.

$$h(x, y; \phi, f) = \exp\left\{-\frac{1}{2}\left[\frac{x_\phi^2}{2\delta_x^2} + \frac{y_\phi^2}{2\delta_y^2}\right]\right\} \cos\left(2\pi f x'_\phi\right) \tag{3.1}$$

Equation 3.1 shows the general form of the Gabor filters used in the enhancement operation. The parameters of the filter are f: the frequency of the sinusoidal plane wave, ϕ: the orientation of the filter and δ_x and δ_y: the Gaussian envelope standard deviations. The f parameter is completely defined by the local ridge frequency of the fingerprint image, and the values of δ_x and δ_y were taken as 4.0 as proposed in [26]. Since we already know ϕ we do not have to calculate it separately.

This operation made the ridge structure way more clear, and enabled us to eliminate the images having noisy ridge structure.

The image classes were analyzed one by one, and the unclear images were eliminated before the training operation while keeping the class balance as constant as possible. The standard and enhanced images are shown in 3.6, and also the problematic images are shown in 3.7. As a final result, 20% of the total synthetic fingerprint images were eliminated and 10500 images are left for training.



(a) Before enhancement (b) After enhancement

Figure 3.6: Effect of enhancement operation on ridge structure clarity

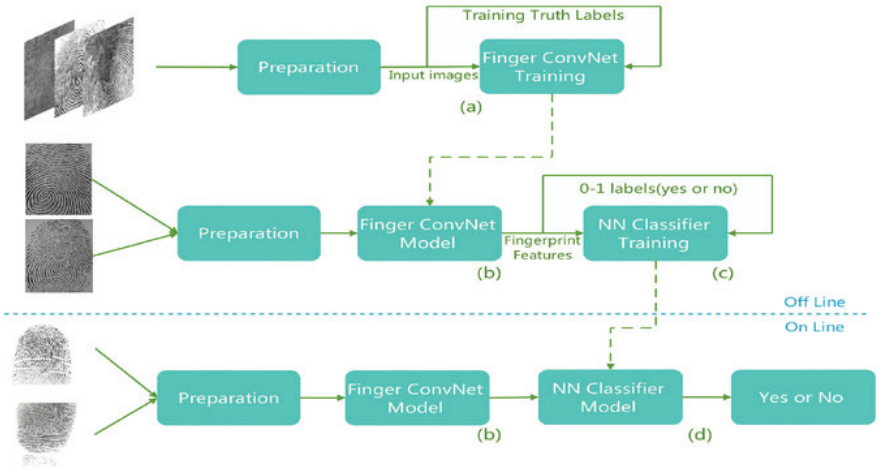
After the elimination operation our dataset has 480 real and 10500 synthetic images divided into 1776 classes. These images will be directly fed to the CNN classifiers.

3.2 Fingerprint Matching Setup

In this section, we analyse the fingerprint matching architecture proposed in [1]. The architecture is depicted in 3.8

(a) Good quality image (b) Distorted image

Figure 3.7: Clean and distorted images' ridge structures



(a) The aligned fingerprint images are given to a classifier network namely Finger ConvNet which does identification. This network is trained with the fingerprint images and the truth labels which indicate whom the finger impression belongs to. (b) After the training the model is able to extract the discriminative features of fingerprint images. (c) Two fingerprint features are concatenated and given to the neural network classifier for verification with YES or NO truth labels indicating if the fingers are matching or not.

Figure 3.8: Overall architecture

3.2.1 CNN Architectures

The CNN architecture in the proposed [1] matching system is used as a feature extractor and these features are used to train a binary neural network classifier. In this study, Finger ConvNet [1], and to compare its performance, Resnet18 with its pre-trained ImageNet weights are used. The details of these architectures will be given in the following

sections.

3.2.1.1 Finger ConvNet

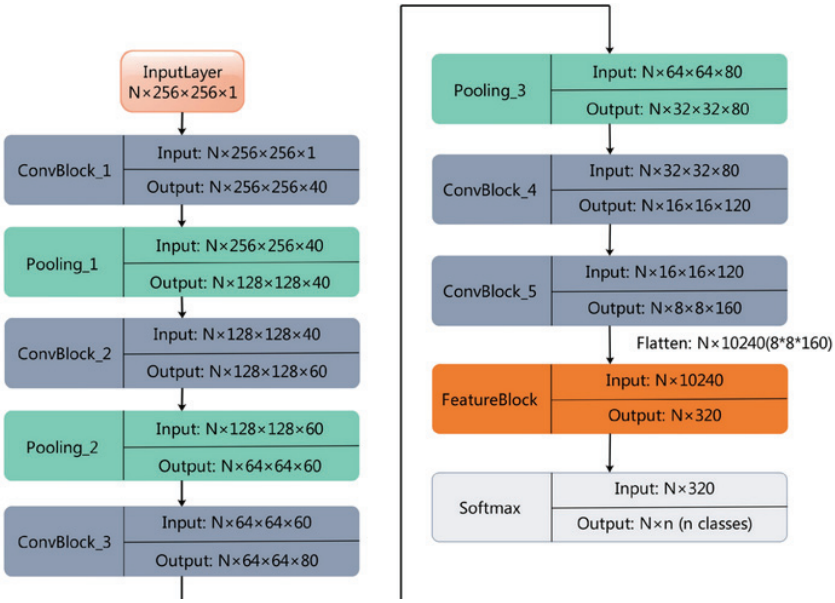


Figure 3.9: Finger ConvNet Architecture

Fingerprint images have their own characteristics and performance demands compared to other image types. Since VGG-16 [7] has strong expressive ability, Finger ConvNet was inspired by the VGG-16 architecture. This network is trained using the fingerprint images and their truth labels indicating the owner of the image, in order to extract the discriminative features out of fingerprint images for matching. Moreover, the matching performance of pre-trained Resnet18 network was compared to discuss the contribution of Finger ConvNet better. Finger ConvNet consists of five ConvBlocks each of which has two convolutions, one batch normalization, and one exponential linear unit layer. The last block is the feature block, and it is followed by a softmax layer.

The hidden layer at the end is kept as a feature layer, where the aforementioned features, which will distinguish one fingerprint from another, are formed. Since in the datasets, the fingerprint images can be of different sizes, all the images are cropped to the

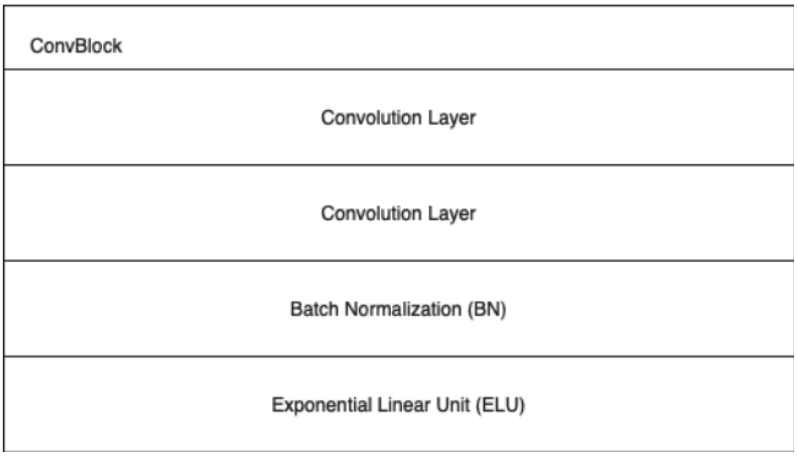


Figure 3.10: First 4 conv block structures

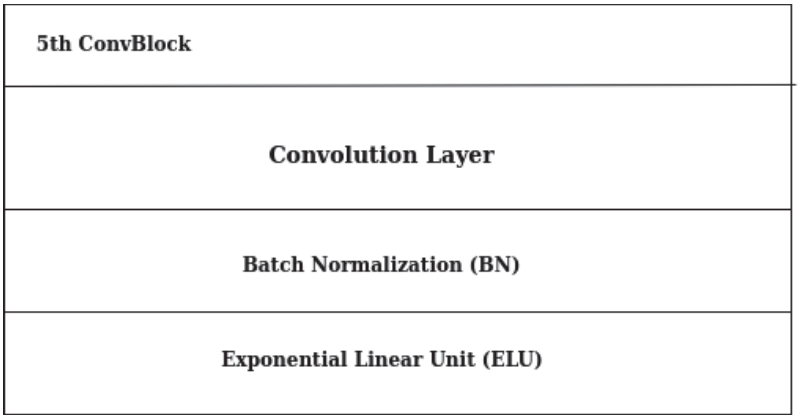


Figure 3.11: The 5th conv block structure

same size 256 x 256 x k, where k is 3 for colored images and 1 for grayscale images. In this setup, the feature later outputs 320-dimensional feature vectors. In the matching step, 320-dimensional feature vector couples are concatenated and, a binary classifier network is trained with these combined features. The Finger Convnet architecture and a single conv block structure is depicted in figures 3.9 and 3.10

Since the dataset size used in this study is smaller than the original work, the last convolution layer was removed while training Finger ConvNet. In the final setup, the 5th conv block has only one convolution later. The rest of the architecture is kept as is. The fifth conv block is shown in 3.11

3.2.1.2 Resnet18

The ResNet architecture is another CNN network that performs well in vision problems. We used this network to compare its performance with Finger ConvNet as a classifier in our matching system. Originally it has been proposed to overcome some problems faced while using deep NN architectures.

Residual networks were proposed by Microsoft Research Team in 2015 [27], and a ResNet won ILSVRC 2015 and proved itself to perform very well in image classification and recognition problems. The main reason residual networks were proposed was that the deeper the networks became, the more their accuracies got saturated since they were converging. This led to a degradation problem in the networks' performances. Resnet18 is used as the second architecture in this study because of its proven performance in classification tasks. Resnet18 was chosen specifically since Resnet35 and Resnet50 models had a slightly worse performance than this network since the dataset size is relatively small to get the optimal performances out of these deeper networks. The performances of Resnet34 and Resnet50 will also be discussed in Chapter 4.

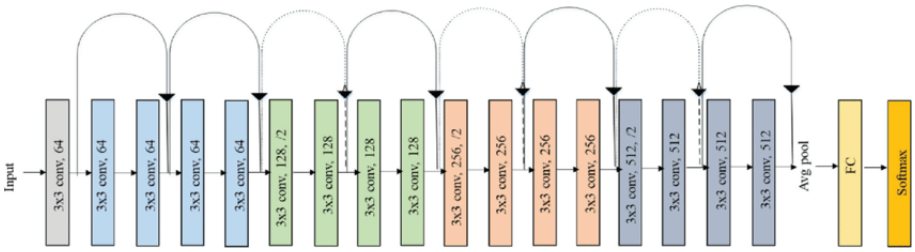


Figure 3.12: Resnet18 architecture

The Resnet architecture was used as a pre-trained model in this setup. All the layers are frozen and they are not trained. 2 dense layers having 128 and 320 layers respectively are put on top of the frozen layers for fine tuning. In order to prevent overfitting, after every dense layer a dropout layer is used. The complete setup of Resnet18 architecture used in this study is given in 3.13. At every layer, 30% of the neurons are disabled to prevent overfitting. The last dense layer having 320 neurons will output the 320-dimensional feature vectors that will be used to train the binary classifier network.

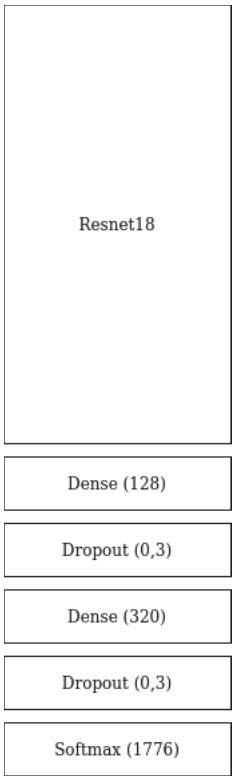


Figure 3.13: Resnet18 setup

3.2.2 Binary Classifier Network

The binary classifier network is the network that is used to measure the matching performance of the discriminative vectors that the classifier networks extracted. The extracted 320-dimensional vectors are concatenated as fingerprint couples and fed into the binary classifier, so that it learns whether two fingerprints are matching or not. The neural network classifier has a very basic structure. It has 640-dimensional input layer, 640-dimensional hidden layer and, 2-dimensional output layer which has sigmoid activation function.

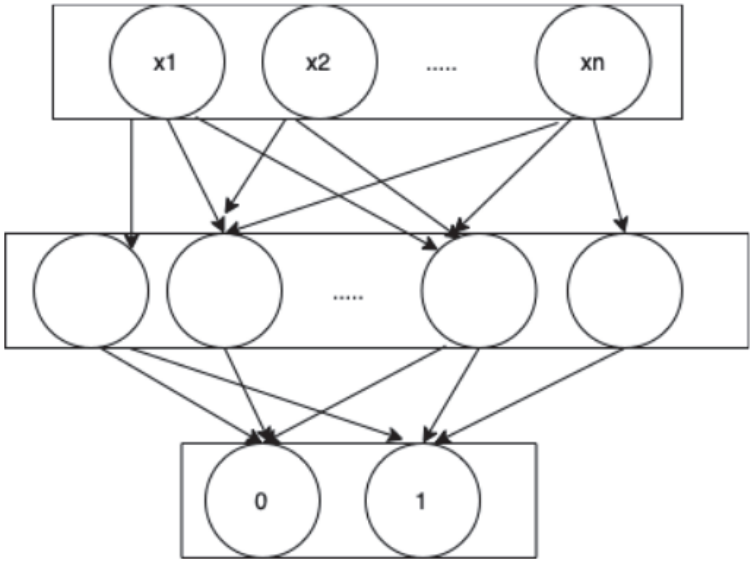


Figure 3.14: NN binary classifier

3.3 Training Setup

3.3.1 CNN Classifier Training Setup

After the alignment of the fingerprint images our setup is ready for training. In a CNN, to learn the mapping function Conv(.), we need to find the best set of parameters θ, which constitutes the weights and biases in each layer. We can achieve this by classifying the feature vectors by an n-dimensional softmax classifier.

Both classifier networks have a softmax layer as a last layer, and it calculates a probability distribution over 1776 classes. The best probability is chosen as the predicted class value while minimizing the categorical cross-entropy loss. We minimize the categorical cross-entropy loss to learn the network parameters. It can be formularized as follows:

$$L_{softmax}(f, t, \theta_{ident}) = -\sum_{n} yi \log(p_i) = -\log(p_t) \tag{3.2}$$

where **f** is the feature vector extracted from a fingerprint image, **t** represents the target class, θ_{ident} denotes the softmax layer, which gives a probability distribution over

fingerprint classes. yi is a 1xn vector that represents the predicted probability distribution, in which **yt = 1** indicates the target class and **yt = 0** for the rest of the classes. Finally, **pi** is the probability distribution that the network predicts.

3.3.1.1 Finger ConvNet Training Setup

All the convolution layers in the Finger ConvNet architecture were initialized with normal distribution having zero mean and 0,001 standard deviation. Adam optimizer is used as the optimizer function.

5-fold cross validation is applied to validate the performance of the classifier, and all folds were run for 700 epochs. On an Nvidia 3070RTX graphics card, every training session which consists of 5-folds lasted around 36 hours.

3.3.1.2 Resnet Training Setup

Resnet18, Resnet34 and Resnet50 training experiments were conducted with the same experimental setups. The Resnet's weights were frozen and 2 dense layers were added on top of the frozen layers to fine-tune the model. As a last layer, a softmax layer was added for the model to output a probability distribution over the fingerprint classes in the dataset. In all of the Resnet setups the convolutional layer weights were initialized with normal distribution having standard deviation and mean equal to 0,01 and 1 respectively. In order to validate the model performance, 5-fold cross validation was applied similar to Finger ConvNet experiments.

The experiments were conducted for 5-folds each having 250 epochs for Resnet18 and 450 epochs for Resnet34 and Resnet50. A single experiment on these networks lasted around 30 hours on a 3070RTX graphics card.

The experimental results of the classifier networks will be discussed in detail in chapters 4.

3.3.2 Binary Classifier Training Setup

In contrast to CNN classifiers, the binary classifier uses binary cross-entropy loss which is only a special case of categorical cross-entropy loss. This function outputs a probability distribution over 2 classes. The class value having a probability over 50% is chosen as the predicted class.

While building the dataset for an NN classifier, to keep the dataset balanced, the number of matching fingerprints are kept equal to the number of non-matching fingerprint couples. This prevented any issues that would be caused by an unbalanced dataset.

The feature vectors have been normalized so that they have zero mean and one standard deviation. The input and hidden layers were initialized with normal distribution having zero mean and 0,01 standard deviation, and all the biases were initialized as zero. In these layers, instead of ELU activation function, RELU was used. Batch normalization layers were used after the input and hidden layers to normalize the inputs passing through these layers.

In this study the real images are taken from FVC2002 dataset, and 120 images from this dataset are taken to construct the test set for the binary classifier network.

CHAPTER 4

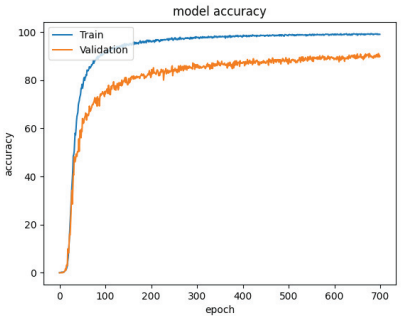
EXPERIMENTAL RESULTS

The classifier networks were trained with 10860 fingerprint images. 360 of them were real images used from FVC2002 dataset (120 of the 480 real images are put aside for the final tests), and 10500 of them were synthetically generated. As the experiments show, both classifier networks got more than 90% of validation accuracy. The matching performances will also be given in details in the following chapters.

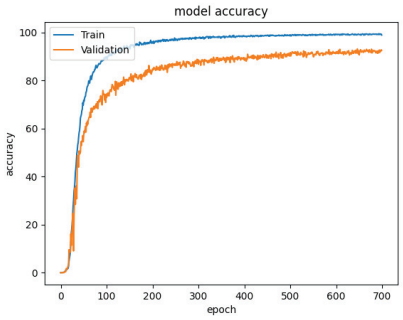
4.1 Finger ConvNet Results

Finger ConvNet architecture got higher results when trained with the mixed dataset, than only with the real data. The real part of the dataset gave worse results since the size of the data was not sufficient. The results show that a dataset that does not have sufficient amount of data can be enlarged using synthetically generated fingerprint data to make a classifier's performance better.

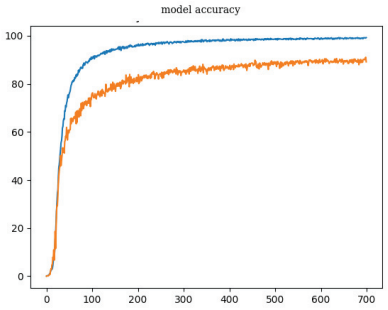
In order to validate the performance of the model, 5-fold cross validation was applied and the model had comparable training performances with the original study [1] on each fold. Figure 4.1 shows the accuracy charts of the 5 folds.



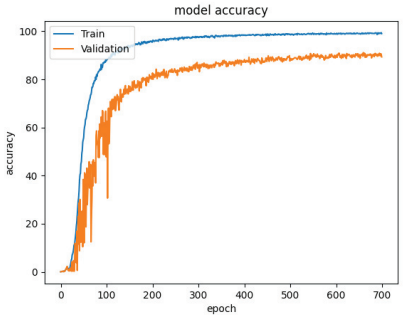
(a) Validation accuracy: 90.83%



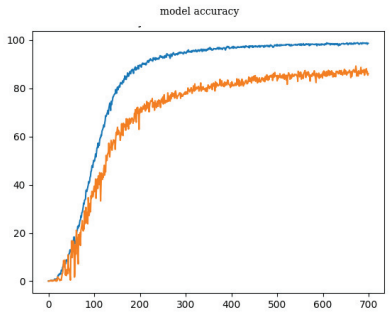
(b) Validation accuracy: 89.37%



(c) Validation accuracy: 92.19%

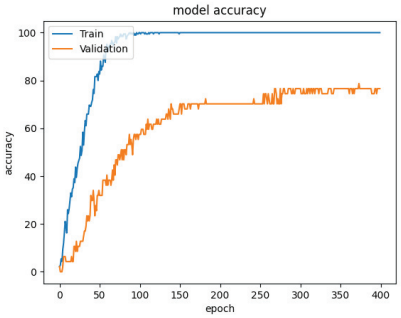


(d) Validation accuracy: 88.71%

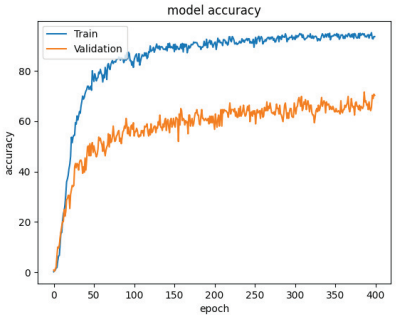


(e) Validation accuracy: 90.28%

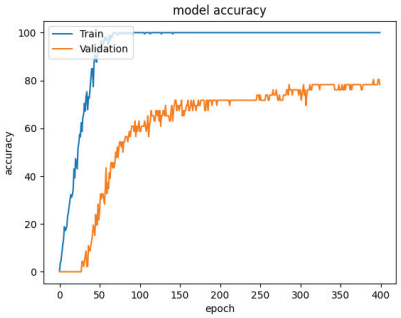
Figure 4.1: 5-fold cross validation results for Finger ConvNet



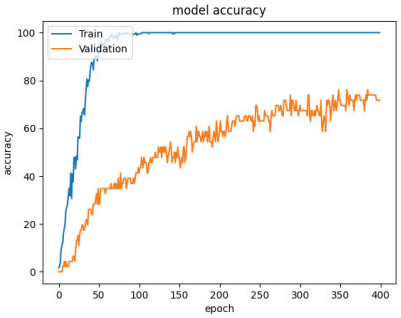
(a) Validation accuracy: 79.13%



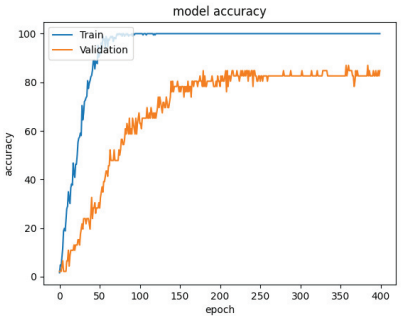
(b) Validation accuracy: 76.16%



(c) Validation accuracy: 81.07%



(d) Validation accuracy: 78.34%



(e) Validation accuracy: 82.12%

Figure 4.2: 5-fold cross validation results for Finger ConvNet with only real data

The model got a satisfying result with the mixed dataset, and when compared to the original work [1], considering the difference between the dataset sizes, the Finger ConvNet implemented in this study did a successful job.

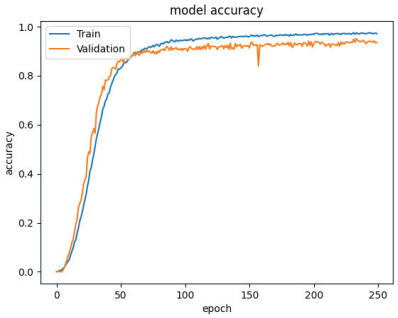
In order to understand the contribution that the synthetic dataset added on top of the real data, a training with only real data was performed. As seen in 4.2, with the dataset having only 480 images the training could not be performed properly and a bad validation accuracy was output. This shows that, enlarging a dataset with synthetic fingerprint data works as expected.

Table 4.1: Finger ConvNet 5-fold cross validation results with their mean(μ) and standard deviation(σ)

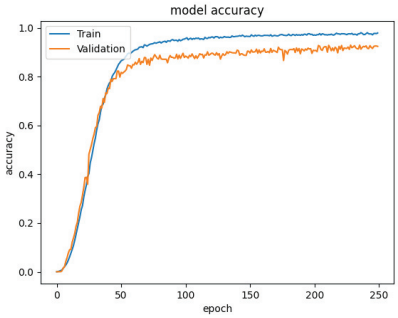
Fold	**Validation Accuracy (Mixed)**	**Validation Accuracy (Only Real)**
1	90.83%	79.13%
2	89.37%	76.16%
3	92.19%	81.07%
4	88.71%	78.34%
5	90.28%	82.12%
μ	90.27%	79.36%
σ	1.20%	1.90%

4.2 Resnet Results

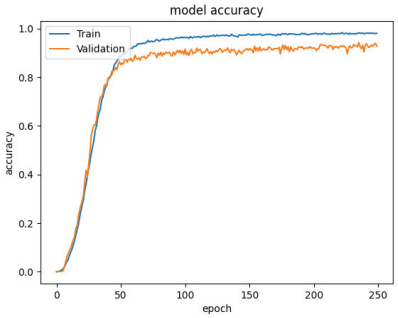
In this study, Resnet18 is chosen as the network to compare the Finger ConvNet's performance. The exact same experimental setup was designed for Resnet18 model and the training was performed. The 5-fold cross validation results for Resnet18 are shown in 4.3. As seen in these figures, this model got better results with the mixed dataset as proposed in the original work [1].



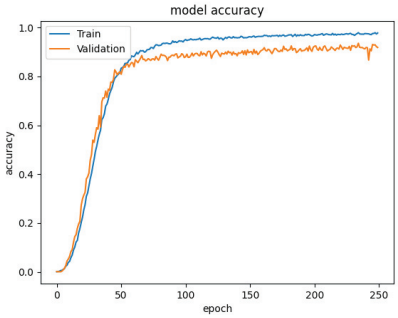
(a) Validation accuracy: 96.23%



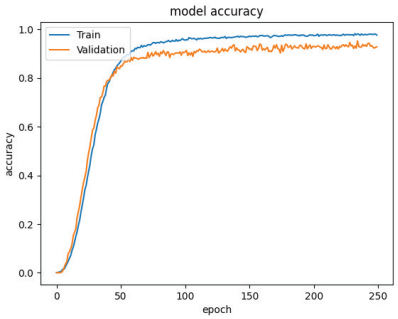
(b) Validation accuracy: 95.93%



(c) Validation accuracy: 94.72%



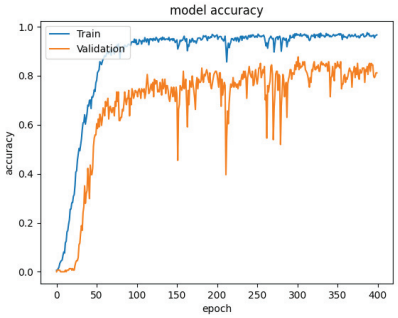
(d) Validation accuracy: 95.19%



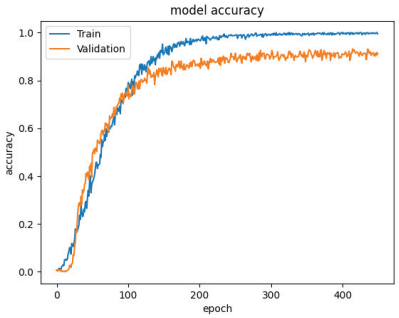
(e) Validation accuracy: 96.78%

Figure 4.3: 5-fold cross validation results for Resnet18

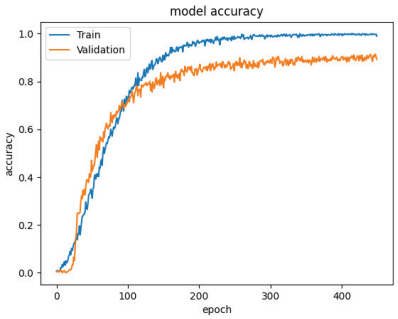
As performed with Finger ConvNet model, a training was performed with only real data with Resnet18, as well. As expected, the results were worse than the mixed dataset. As seen in 4.5, completing the dataset with synthetic fingerprint images increased the performance in Resnet18 experiments.



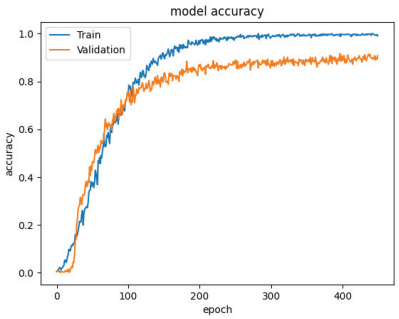
(a) Validation accuracy: 86.54%



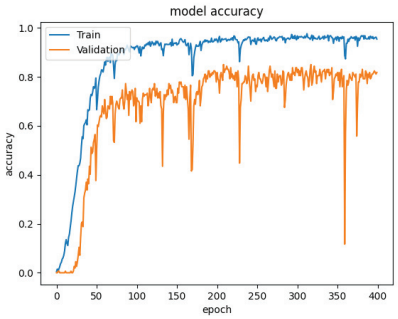
(b) Validation accuracy: 87.23%



(c) Validation accuracy: 88.37%

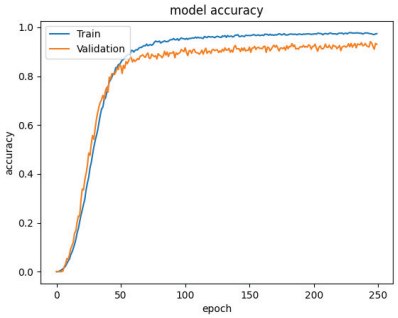


(d) Validation accuracy: 87.64%

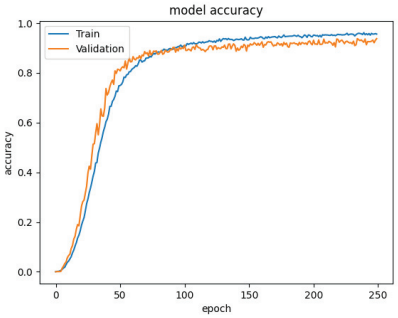


(e) Validation accuracy: 85.98%

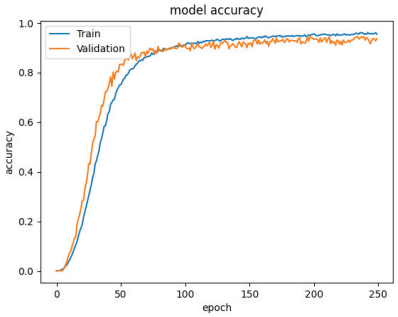
Figure 4.4: 5-fold cross validation results for Resnet18 with only real data



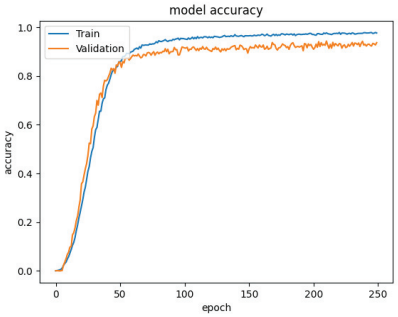
(a) Validation accuracy: 93.97%



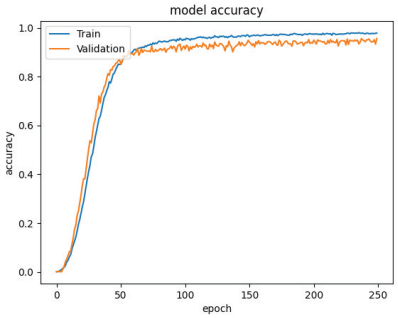
(b) Validation accuracy: 93.88%



(c) Validation accuracy: 93.24%

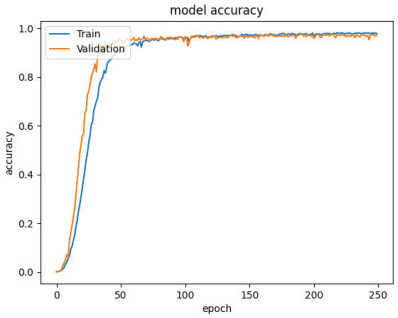


(d) Validation accuracy: 95.01%

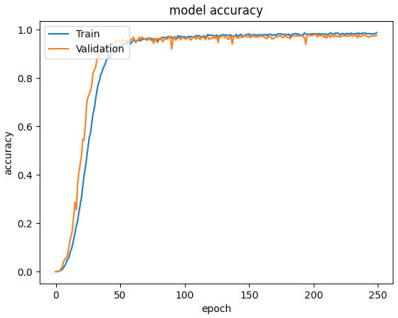


(e) Validation accuracy: 92.76%

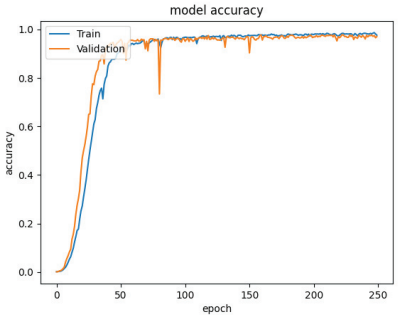
Figure 4.5: 5-fold cross validation results for Resnet34 with mixed dataset



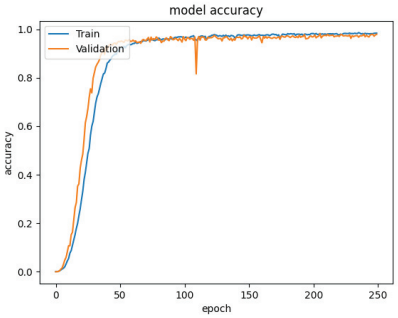
(a) Validation accuracy: 94.21%



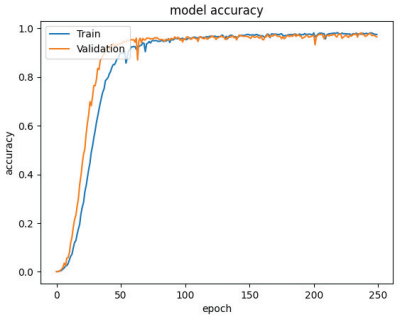
(b) Validation accuracy: 94.81%



(c) Validation accuracy: 96.23%



(d) Validation accuracy: 94.93%



(e) Validation accuracy: 96.12%

Figure 4.6: 5-fold cross validation results for Resnet50 with mixed dataset

Table 4.2: Resnet18 5-fold cross validation results with their mean(μ) and standard deviation(σ)

Fold	Validation Accuracy (Mixed)	Validation Accuracy (Only real)
1	96.23%	86.54%
2	95.93%	87.23%
3	94.72%	88.37%
4	95.19%	87.64%
5	96.78%	85.98%
μ	95.77%	87.15%
σ	0.67%	0.83%

Table 4.3: Resnet18, Resnet34 and Resnet50 5-fold accuracy comparison

Fold	Resnet18	Resnet34	Resnet50
1	96.23%	93.97%	96.13%
2	95.93%	93.88%	95.27%
3	94.72%	93.24%	94.23%
4	95.19%	95.01%	96.54%
5	96.78%	92.76%	94.13%
μ	95.77%	93.77%	95.26%
σ	0.67%	0.76%	0.97%

4.3 Binary Classifier Results

In the overall fingerprint matching architecture proposed in this study, the binary classification network is used to evaluate the performance of the previously mentioned classfication networks. The features extracted from the trained classifiers are used to train the binary classifier network. In order to set up a system to perform a fingerprint matching, the feature vector couples are concatenated and fed into the network as shown in 4.7. After the network is trained the performance on the test set is evaluated.

Before feeding the features to the binary classifier the concatenated feature vectors are normalized so that they have a zero mean. The 4.8 shows the feature vectors around the origin.

The binary classifier was trained very smoothly and the results from Finger Con-

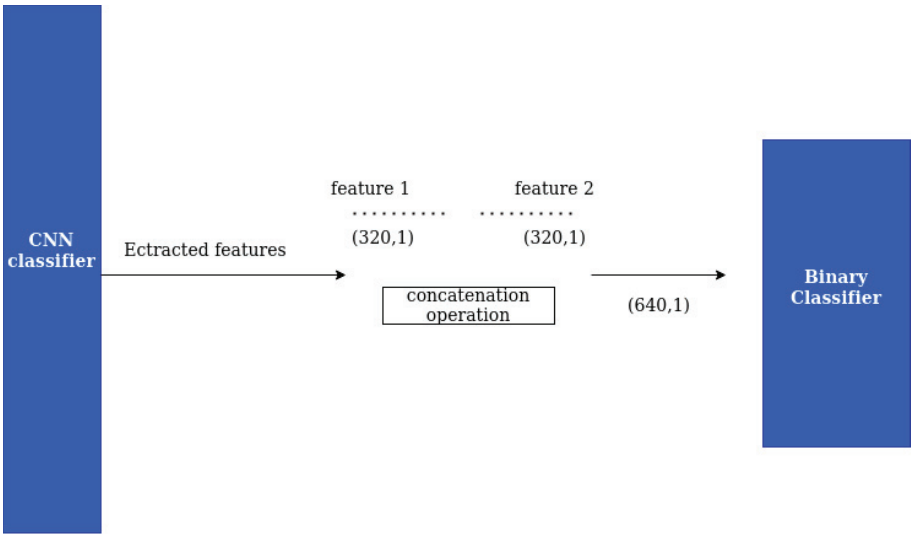


Figure 4.7: Binary classifier data preparation

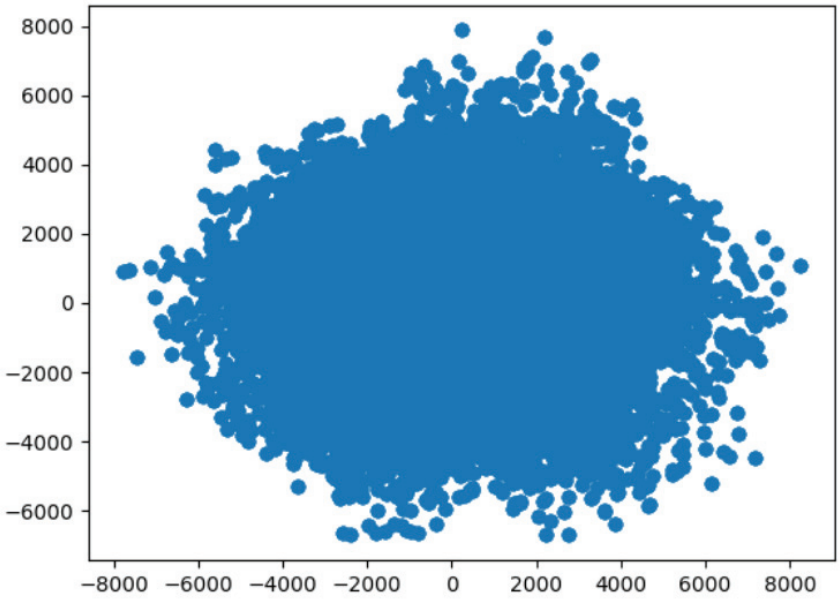
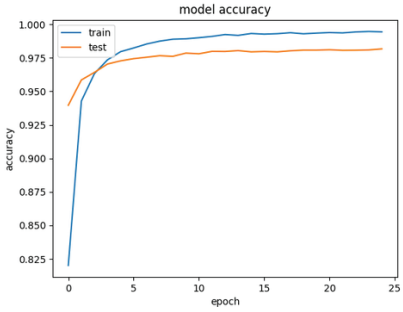


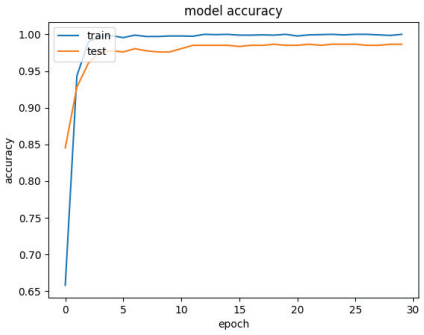
Figure 4.8: Normalized concatenated feature vectors

vNet and ResNet18 features were given in 4.8. The Finger ConvNet features got 80.37% test accuracy with 89.25% precision and 88.48% recall values using mixed-dataset features. It got 77.87% test accuracy with 85.43 precision and 84.56% recall values using the features from only-real dataset with the same setup. The Resnet18 model got 84.12% test accuracy with 87.54% precision and 88.09% recall using mixed-dataset features and 82.76% test accuracy with 84.29% precision and 84.12% recall using only-real dataset features. The whole testing setup was conducted using the test set from FVC2002, which

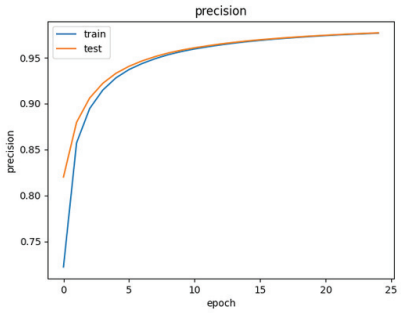
was mentioned before, consisting of only real fingerprint images. The precision and recall being higher than the test accuracy could be because of the number of non-matching fingerprint images in the binary classifier dataset than number of matching images.



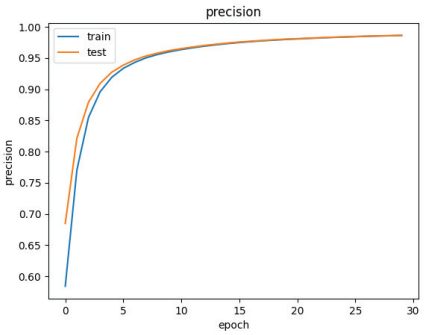
(a) Finger Convnet Accuracy: 96.09%



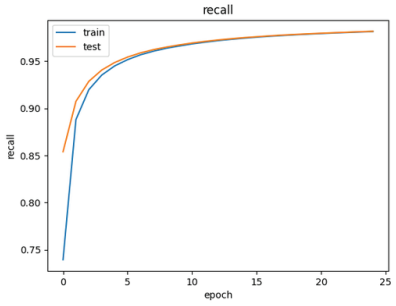
(b) Resnet Accuracy: 97.46%



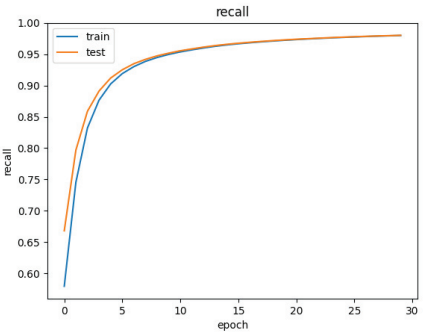
(c) Finger ConvNet Precision: 96.38%



(d) Resnet Precision: 96.91%



(e) Finger ConvNet Recall: 97.19%



(f) Resnet Recall: 97.38%

Figure 4.9: Binary classifier results

4.4 Summary

The experimental results show that Finger ConvNet and Resnet architectures can be trained with the mixed dataset, and the performances are close to the results proposed in [1]. The reason there is a difference between the performances is that there is still a considerable amount of difference between the dataset sizes. The Finger ConvNet got around a 92% mean validation accuracy with the mixed dataset, and compared to the original work, this is a reliable result, although it might change depending on the fault tolerance of the problem, for fingerprint classification and matching.

The Resnet18, Resnet34 and Resnet50 networks got better results than the Finger ConvNet model. This is an expected case, because in this study, a transfer learning approach was applied on this models, and transfer learning is already the governing approach when insufficient amount of data are present at hand. Resnet18 got around 96% of mean validation accuracy with the mixed dataset and it is very close to the proposed results of pre-trained networks in [1].

The small difference between the model performances compared to the dataset size differences can be explained with the fact that the number of classes in the dataset used in the original work increases as the dataset size becomes larger. In the real dataset experiments in this study, the performance deteriorated even more, since the dataset size was not sufficient even for the weights in the model to converge well.

The final test results from the experiments conducted on the real fingerprint images gave a worse performance than the original work. The reason for that could be that the number of real fingerprint images were the minority in our mixed dataset. The performance could be increased by having a larger portion of the dataset consisting of real fingerprint images. Synthetic fingerprint images increased the performance of the feature extractors, however, the same amount of increase was not reflected in the overall matching performance. Nevertheless, there is a slight increase between the matching performances of only-real and the mixed datasets

The comparative results of classifier networks and datasets in this study and the original work is given in tables 4.4, 4.5 and 4.6. The table 4.5 shows that Resnet18 got a better result than Finger ConvNet, since the dataset is still smaller that the one used in the original work. Since transfer learning approach was applied in Resnet experiment setups, a better result is expected. Table 4.6 shows that, Finger ConvNet and Resnet18 got similar result in matching task as well, by keeping a similar difference they had in the feature extraction setup.

Table 4.4: Dataset comparisons

	Original Study	**Mixed Dataset**	**Only Real Dataset**
Number of classes	20-29k	1776	60
Total number of images	100-170k	10860	360

Table 4.5: Feature extractor validation accuracy comparisons

	Original Study	**Mixed Dataset**	**Only Real Dataset**
Finger ConvNet	96.89%	90.48%	79.44%
Resnet18	96.18%	96.02%	89.15%

Table 4.6: Fingerprint matching performance comparison

	Original Study	**Mixed Dataset**	**Only Real Dataset**
Finger ConvNet features	96.89%	80.37%	79.36%
Resnet18 features	96.34%	84.12%	82.76%

CHAPTER 5

CONCLUSION

The fact that training neural networks require large dataset sizes is a known truth in the deep learning literature. Providing the model with the sufficient amount of data is an important concern, since the generalization power of the model after the training process depends on this.

The problem of collecting a large dataset has always been a problem since the very first feed forward networks stepped into the machine learning literature. The motivation of this project is to compare the performances of well-performing neural network architectures with the classifier network architecture proposed in [1] with different dataset sizes. Using a dataset consisting of only real images may not always be possible, since collecting a large dataset is a tedious work. As a solution to this problem, the idea of employing a synthetically generated dataset for neural network training has arisen.

As a result of the experiments conducted in this study, it is shown that enlarging a dataset using synthetic fingerprint images help improve the feature extractor performances. However, the number of real fingerprint images in the training dataset is an important factor. Since the number of images in our dataset was not as many as in the original work, there is a difference between the matching performances on Finger ConvNet experiments. This shows that synthetic fingerprint images can be used in neural network training for classification and matching problems provided that the number of real fingerprint images is not too small.

REFERENCES

[1] C. H. Yonghong Liu, Baicun Zhou, "A novel method based on deep learning for aligned fingerprints matching," *Applied Intelligence*, 2019.

[2] P. İrtem, E. İrtem, and N. Erdoğmuş, "Impact of variations in synthetic training data on fingerprint classification," in *BIOSIG 2019 - Proceedings of the 18th International Conference of the Biometrics Special Interest Group* (A. Brömme, C. Busch, A. Dantcheva, C. Rathgeb, and A. Uhl, eds.), (Bonn), pp. 189–196, Gesellschaft für Informatik e.V., 2019.

[3] M. D. Capelli R, Erol A and M. D, "Synthetic fingerprint image generation," *International Conference on Pattern Recognition*, 2000.

[4] I. S. Alex Krizhevsky, Geoffrey E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, 2012.

[5] K. O. Ghaddab MH, Jouini K, "Fast and accurate fingerprint matching using expanded delaunay triangulation," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 1998.

[6] T. I. Peralta D, Galar M and M.-H. O, "Minutiae filtering to improve both efficacy andefficiency of fingerprint matching algorithms," *Engineering Applications of Artificial Intelligence*, 2014.

[7] W. Y. Luo X, Tian J, "A minutiae matching algorithm in fingerprint verification," in *International Conference on Pattern Recognition*, IEEE, 2000.

[8] P. J. C. Jin, "Fingerprint classication in dct domain using rbf neural networks," *J. Inform. Sci. Eng. 25*, 2009.

[9] G. T. K. Moscinska, "Neural network based ngerprint classication," *Proceedings of the Third International Conference on Articial Neural Networks*, 1993.

[10] A. Senior, "A combination ngerprint classier," *IEEE Trans. Pattern Anal. Mach. Intell. 23*, 2001.

[11] A. Senior, "Fingerprint classication by decision fusion," *Automatic Fingerprint Recognition Systems*, 2004.

[12] J. G. A. Lorena, A. Carvalho, “A review on the combination of binary classiers in multiclass problems,” in *Artif. Intell. Rev.*, pp. 1–4, IEEE, 2008.

[13] E. B. M. Galar, A. Fernández, H. Bustince, and F. Herrera, “An overview of ensemble methods for binary classiers in multi-class problems: experimental study on one-vs-one and one-vs-all schemes,” *Pattern Recogn.*, 2012.

[14] M. P. Y. Yao, P. Frasconi, “Fingerprint classication with combination of support vector machines,” *Proceedings of the 3rd International Conference on Audio- and Video-Based Biometric Person Authentication*, 2003.

[15] M. P. Y. Yao, G.L. Marcialis, P. Frasconi, and F. Roli, “Combining at and structured prepresentations for ngerprint classication with recursive neural networks and support vector machines,” *Pattern Recogn. 36*, 2003.

[16] H. W. J. Li, W.-Y. Yau, “Combining singular points and orientation image information for ngerprint classication,” *Pattern Recogn. 41*, 2008.

[17] M. X. J. Hu, “Fingerprint classication based on genetic programming,” *International Conference on Computer Engineering and Technology (ICCET)*, 2010.

[18] C. B. L. Jiang, T. Zhao and A. Yong, “A direct fingerprint minutiae extraction approach based on convolutional neural networks,” *IJCNN*, 2016.

[19] J. F. Y. Tang, F. Gao and Y. Liu, “Fingernet: An unified deep network for fingerprint minutiae extraction,” *International Joint Conference on Biometrics*, 2017.

[20] C. I. Watson and C. L. Wilson, “Nist special database 4,” *Fingerprint Database, National Institute of Standards and Technology*, vol. 17, no. 77, p. 5, 1992.

[21] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain, “Fvc2002: Second fingerprint verification competition,” in *Object recognition supported by user interaction for service robots*, vol. 3, pp. 811–814, IEEE, 2002.

[22] M. H. Adhwa Alrashidi, Ashwaq Alotaibi and H. AlShehri, “Cross-sensor fingerprint matching using siamese network and adversarial learning,” *International Conference on Pattern Recognition*, 2021.

[23] A. K. Chenhao Lin, “Multi-siamese networks to accurately match contactless to contact-based fingerprint images,” *International Joint Conference on Biometric*, 2017.

[24] A. K. Chenhao Lin, “Improving cross sensor interoperability for fingerprint identification,” *nternational Conference on Pattern Recognition*, 2016.

[25] E. İrtem, “Synthetic generation of fingerprints,” *Izmir Institute of Technology*, 2020.

[26] A. J. Lin Hong, Yifei Wan, “Fingerprint image enhancement: Algorithm and performance evaluation,” *International Conference on Pattern Recognition*, 2016.

[27] S. R. Kaiming He, Xiangyu Zhang and J. Sun, “Deep residual learning for image recognition,” *Computer Vision and Pattern Recognition*, 2015.