B. TECH. PROJECT REPORT

On

Presentation Attack Detection in Fingerprint Biometrics

By
PRAKHAR RAI 180001035

DISCIPLINE OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY INDORE
May 2022

Presentation Attack Detection in Fingerprint Biometrics

A PROJECT REPORT

Submitted in partial fulfillment of the requirements for the award of the degrees

of
BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING

Submitted by:
Prakhar Rai

Guided by:
Dr. Somnath Dey
(Associate Professor, CSE)

DISCIPLINE OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY INDORE
May 2022

CANDIDATE'S DECLARATION

I hereby declare that the project entitled **"Presentation Attack Detection in Fingerprint Biometrics"** submitted in partial fulfillment for the award of the degree of Bachelor of Technology in 'Computer Science and Engineering' completed under the supervision of **Dr. Somnath Dey, Associate Professor, Computer Science and Engineering, IIT Indore** is an authentic work.

Further, I declare that I have not submitted this work for the award of any other degree elsewhere.

Prakhar Rai 2022/05/27

Signature and name of the student(s) with date

CERTIFICATE by BTP Guide(s)

It is certified that the above statement made by the student is correct to the best of my knowledge.

Somnath Dey

27.05.2022

Signature of BTP Guide(s) with dates and their designation

Preface

The report on Presentation Attack Detection in Figerprint Biometrics is prepared under the supervision of **Dr. Somnath Dey**, Associate Professor, Computer Science and Engineering, IIT Indore.

We attempted to provide a full description of the Presentation Attack Detection problem and various software-based techniques in this study so that the solution is both economically sound and practicable. In order to improve the outcomes, we focused on combining Machine Learning and Deep Learning-based methodologies. We have attempted to describe the topic as clearly as possible to the best of our abilities and knowledge.

Prakhar Rai
B.Tech. IV Year
Discipline of Computer Science and Engineering
IIT, Indore

Contributions

The major contributions made by me are as follows:

1. Proposal and Python implementation of Dynamic Hybridisation of Mobilenetv1 and SVM.

2. Training of the above proposed model over LivDet 2013, 2015 and 2017 dataset.

3. Performance evaluation and comparative analysis of the proposed models for LivDet 2013, 2015 and 2017.

I would also like to appreciate the contributions made my project partner. The major contribution made by him include:

1. Proposal and Python implementation of Static Hybridisation of MobileNetv1 and SVM.

2. Training of the above proposed model over LivDet 2013, 2015 and 2017 dataset.

3. Performance evaluation and comparative analysis of the proposed models for LivDet 2013, 2015 and 2017.

Acknowledgements

I wish to thank **Dr. Somnath Dey** for his kind support, expertise and valuable guidance. He provided a perfect environment for critical thinking and research acumen and was always available for discussions, doubt clearance and guidance at every part of the project. He has constantly motivated us to take the project to its very culmination.

I would also like to acknowledge **Mr. Anuj Rai** for his support and sincere cooperation. It is his guidance and help, due to which we able to complete the second phase of our project involving machine learning. He was always available for discussion and doubt clearance.
I also thank my project partner **Pradeep Patidar** who contributed sincerely to the making of this project.

Prakhar Rai
B.Tech. IV Year
Discipline of Computer Science and Engineering
IIT, Indore

Presentation Attack Detection in Fingerprint Biometrics

Abstract

There has been a rapid growth in services like finance which utilize fingerprint biometrics for user authentication. This has led to an increase in the need for a secure and reliable fingerprint recognition system to provide privacy and prevent fraud. We are proposing a novel end-to-end fingerprint presentation attack detection method based on the combination of Machine learning and Deep learning. In our proposed model, a Deep CNN architecture i.e. MobileNet v1 is used for the extraction of important features from input images while the actual classification takes place using a support vector machine.

The proposed model is tested on LivDet 2013, 2015 and 2017 datasets and compared with various state-of-the-art methods. Our model achieves an average accuracy of **98.88%** in LivDet 2013, **96.74%** in LivDet 2015 and **94.90%** in LivDet 2017 datasets.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A fingerprint biometric recognition system is an easy, cost-effective, and user-friendly method of authentication. As compared with other biometric systems, it needs less time, resources, and human effort to validate a person's identity. Due to such a level of easiness and automation, it is being adapted by various commercial organizations and security agencies for person verification and authentication. However it's adeptness, it suffers from various challenges such as Theft of identity, account hacking, unauthorized access,and many more. Presentation attack or impersonation is a challenging issue for these systems. This attack is performed by presenting an artifact of a genuine user's fingerprint to the biometric sensor to gain access to it. Various spoofing materials such as Woodglue, Gelatine, Modasil are available at a cheap cost that makes it convenient to place this attack on fingerprint biometric systems.

Biometric qualities cannot be readily shared, forgotten, or copied, biometric recognition creates a tight link between a person and it's identity. As a result, biometric recognition is fundamentally superior to the two traditional techniques of recognition, namely passwords and tokens, and is more resistant to social engineering assaults (e.g., phishing).
Biometric recognition prevents users from making fraudulent repudiation claims since it requires the user to be present at the moment of authentication. Furthermore, only biometrics can offer negative identification capabilities, which are used to determine if a certain individual is enrolled in a system despite the fact that the user may deny it. Biometric recognition has been generally acclaimed as a natural, dependable, and indispensable component of any identity management system because of these features.

Fingerprint recognition systems are vulnerable to presentation attacks. In a presentation attack,

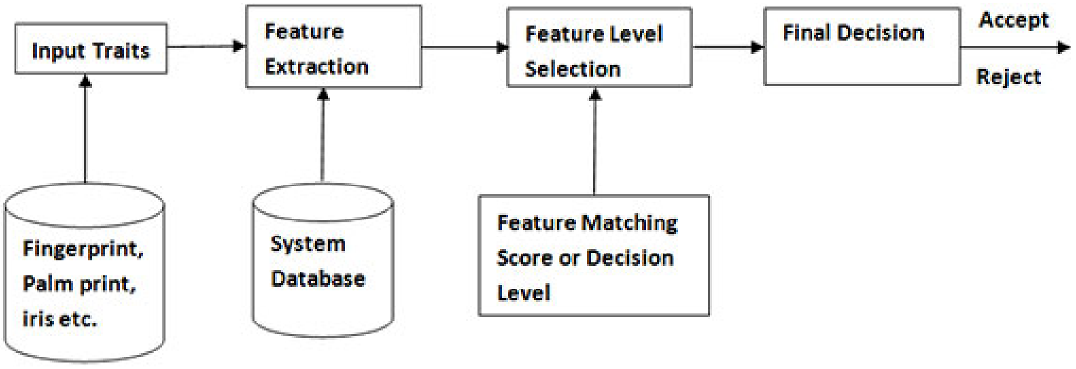

FIGURE 1.1: Live and fake fingerprint images with different spoof materials.

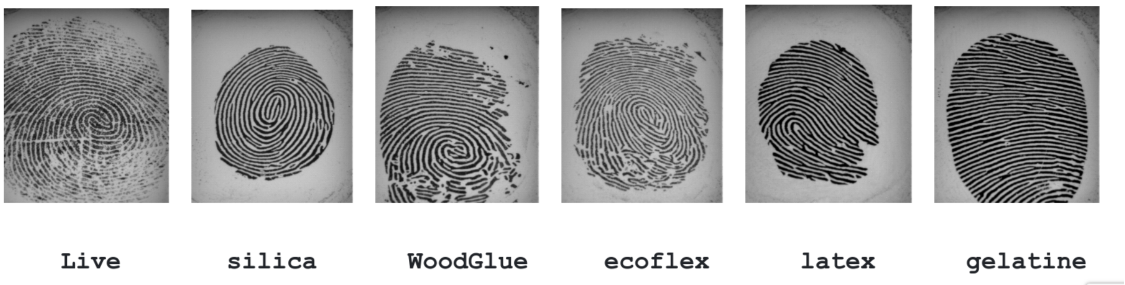

FIGURE 1.2: Live and fake fingerprint images created with different spoof materials.

a synthetic fingerprint is presented for authentication that is replicated from a genuine user. Various sensors are used to acquire live and fake fingerprint images. A liveness detection method has been developed to fight various forms of spoof attacks by identifying the features of live and fake fingerprint photographs. Many hardware and software-based techniques have been proposed by researchers in recent years. The concerns, however, remain difficult in terms of robustness, efficacy, and efficiency. In this work, we look at a variety of software-based solutions for distinguishing between actual and false fingerprints, as well as a complete review of previous efforts to solve the problem.

While most methods can identify a spoof fingerprint created with a given material, their performance in a cross-sensor or cross-material scenario varies. Because of their increased classification capabilities, many researchers have turned to deep-learning-based systems to solve this challenge. These algorithms capture minute information from fingerprint pictures using a series of convolution layers, followed by dense layers for classification. Our main objective is to merge machine learning with deep learning to produce a new classifier that utilises both techniques.

Moving further we have provided literature survey in Chapter 2, the proposed methodology in Chapter 3 and thereafter the results for the proposed method.

Chapter 2

Related Work

The literature on previously known techniques of Presentation Attack Detection is discussed in the next chapter. It mostly focuses on Deep Learning and Transfer Learning methodologies.Researchers have suggested different state-of-the-art fingerprint presentation assault detection algorithms in this area. A reliable method for identifying PAD that uses histogram equalisation to improve contrast is Arora et al. [1]. Fingerprint pictures are submitted into the VGG architecture as a classifier after contrast enhancement. The authors used a variety of fingerprint datasets to evaluate their conclusions, including FVC 2006, ATVSFFp, Finger vein dataset, LivDet 2013 and 2015 databases. A fingerprint presentation assault detection approach based on multi-modal CNN is offered in Anusha et al. [2]. They recommend the spatial attention module and the channel attention module. The first uses the colour channel relationship to extract fundamental information from fingerprint photos and finds the most discriminating patch. A deep learning-based approach for patching a fingerprint picture and feeding it to a CNN Model is Chugh et al. [3].They use this fine detail information to divide a fingerprint picture into patches. The model predicts the spoofness score for all of the patches that are fused together to give the global spoofness score. This method was evaluated on the Livdet datasets from 2013, 2015, and 2017, as well as the MSU-FPAd dataset. In [4], Nogueira et al., in their research, they employed pre-trained CNN architectures with style transferring. Among other designs, they used VGG, Alexnet, and CNN with support vector machines. The Livdet datasets from 2009, 2011, and 2013 are used to validate the method. In [5], Rohith et al. emphasized on texture-based features for fingerprint presentation attack detection . In their work, they, utilized Speeded Up Robust Feature (SURF) and Pyramid extension of Histogram of Gradients (PHOG) to extract the shape as a liveness property. A model based on additive learning was proposed in [6], Ajita et al.. The multi-class classifier used in this study produces three different outcomes: live, false, and unknown. The fingerprints that have been classified as unknown are utilised to train a model against cross-material situations. Ridge and valley clarity, Ridge and valley smoothness, Number of aberrant ridges and valleys, Frequency domain analysis, Orientation confidence level, and more custom features were recommended for fingerprint presentation assault detection in [7], Ram Prakash et al.. By combining these properties, the suggested technique has done a range of tests. Random Forest was used as a classifier, and the approach was tested using the Livdet datasets from 2009, 2013, and 2015. The utility of statistics features is shown in [8], Choi et al..The features i.e. Histogram, Directional contrast, Ridge thickness, and ridge signal are used to train a Support vector machine classifier on a custom-made dataset. A gradient-based technique was proposed by Xia et al.[9].The second and third-order co-occurrence matrices of the gradients were extracted and utilised as a feature in the SVM classifier's training.

Chapter 3

Proposed Methodology

We determined that deep learning and machine learning tools are equally successful for classification after analysing the literature, however that machine learning-based tools rely on other methodologies to extract differential characteristics. The work suggested in [7], [6], [10] demonstrates that machine learning may be used more effectively if feature extraction algorithms are capable of extracting characteristics that aid in the categorization of fingerprint samples made from unknown materials.
Considering the capabilities of SVM, we chose to design a hybrid technique in which a deep learning model contributes to feature extraction and SVM contributes to feature classification. This hybridization can take two forms. The first is static, in which both models train independently, but in our proposed model, the loss supplied by SVM is used to train deep CNN, allowing the architecture to operate end-to-end. We used both techniques in our study and discovered that the dynamic combination outperforms the static one. Both sets of findings are compared to several state-of-the-art approaches.

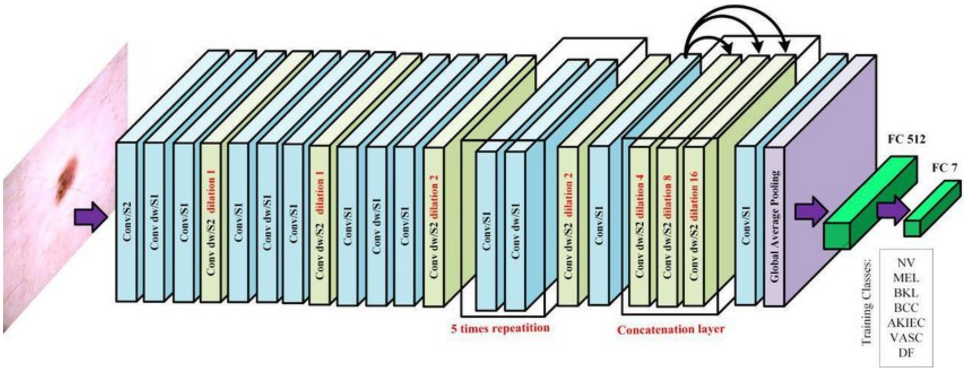

FIGURE 3.1: Block diagram of MobileNet V1 Architecture.

3.1 MobileNet v1

MobileNet 3.1 is a CNN architecture that is efficient and portable and is used in real-world applications. To build lighter models, MobileNets primarily employ depthwise separable convolutions rather than the standard convolutions used in previous architectures. MobileNets adds two new global hyperparameters (width multiplier and resolution multiplier) that allow model developers to trade off latency or accuracy for speed and low size based on their needs.

3.1.1 Architecture

MobileNet is composed of depth-separable convolution layers. Each depthwise separable convolution layer is made up of a depthwise and a pointwise convolution. A MobileNet has 28

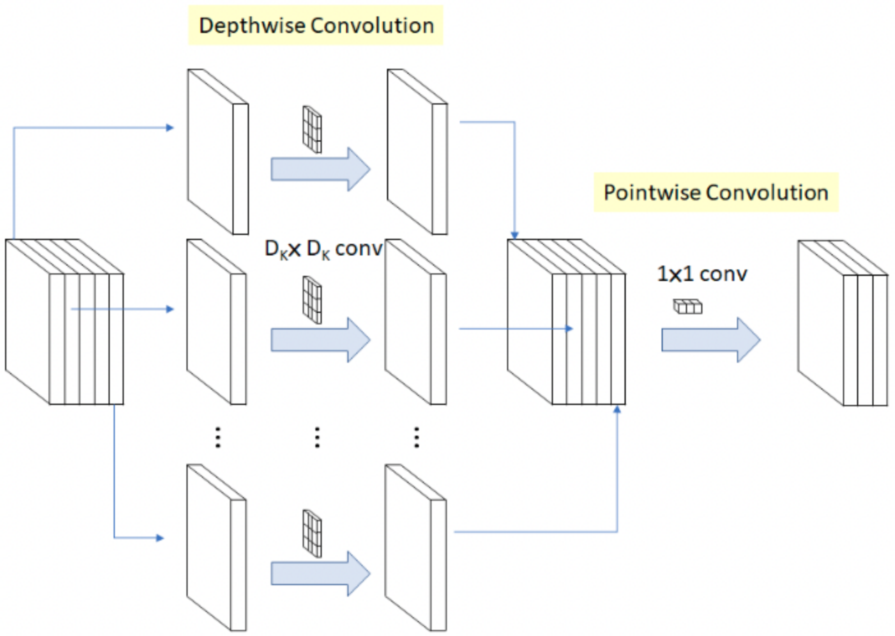

FIGURE 3.2: Depthwise Separable Convolution.

layers if depthwise and pointwise convolutions are counted separately.
A basic MobileNet contains 4.2 million parameters, which can be further decreased by setting the width multiplier hyperparameter properly. The size of the supplied picture is 224 × 224 x 3.

3.1.2 Depthwise Separable Convolution

The depthwise separable convolution 3.2 gets its name from the fact that it deals with not only the spatial dimensions, but also the depth dimension — the number of channels. A depthwise separable convolution divides a kernel into two separate kernels that perform two convolutions: depthwise and pointwise. To apply a single filter to each input channel, we use depthwise convolutions (input depth). The output of the depthwise layer is then linearly combined using pointwise convolution, a simple 11 convolution. For both layers, MobileNets employ batchnorm and ReLU nonlinearities.

3.1.3 Advantages of MobileNet v1

While enhancing spoof detection performance, MobileNet-v1 decreases model size and training/evaluation time. It's a low-latency network that classifies an input fingerprint picture as real or fake in 100 milliseconds, compared to 800 milliseconds for the Inception-v3 network. MobileNet-v1 (4.24 million) has a lower number of model parameters to train than Inceptionv3 (23.2 million) and VGG (138 million), needing less regularisation and data augmentation to avoid overfitting.

3.2 Support Vector Machine

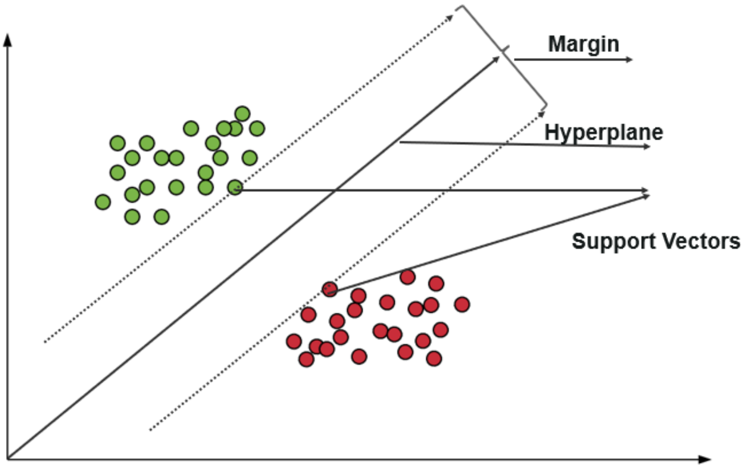

FIGURE 3.3: Support Vector Machine

SVM is a supervised machine learning algorithm that can be used for both classification and regression. The purpose of the SVM method is to discover a hyperplane in an N-dimensional space that unambiguously categorises the input points. The number of features determines the size of the hyperplane. If there are only two input features, the hyperplane is simply a line. There are two types of Margins in SVM, Hard Margin and Soft Margin. In Hard Margin we linearly separate data without misclassifcation provided the data is linearly separable. But if the data is not linearly separable then we use soft margin. In our implementation we have used Soft Margin SVM. SVM learns the W parameters by solving the optimization problem given in the equation below. This equation is known as the primal form problem of L1-SVM, with the standard **hinge loss**.

$$\min_{w} \frac{1}{2} W^T W + C \sum_{i=1}^{p} max(0, 1 - y_i'(W^T X_i + b))$$

Because L1-SVM is not differentiable, a common variant is the L2-SVM, which reduces squared hinge loss.

$$\min_{w} \frac{1}{2} W^T W + C \sum_{i=1}^{p} max(0, 1 - y_i'(W^T X_i + b))^2$$

In the preceding sections, we discussed the capabilities of both tools for picture categorization; however, machine learning relies on additional approaches for feature extraction. Several writers have developed approaches that combine Deep Learning with Machine Learning. In the next part, we go over our process in great depth.

3.2.1 Hinge Loss

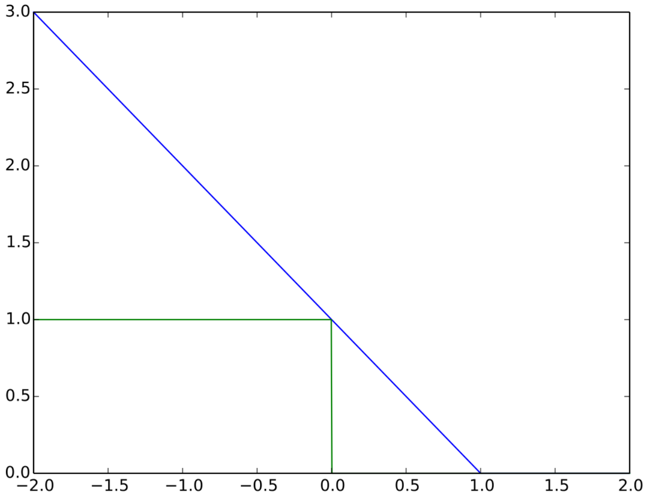

FIGURE 3.4: Hinge Loss

The hinge loss is a form of cost function that is linearly increasing that calculates the cost based on a margin or distance from the classification boundary. Even if additional observations are accurately classified, they may be penalised if the margin from the decision border is insufficient. The hinge loss ensures that the classifier will identify the classification border that is as far away from each of the multiple classes of data points as possible during training. In other words, it determines the categorization border that ensures the greatest possible buffer between the data points of the various classes.

Hinge Loss Formula

$$max(0, 1 - yt)$$

where y is the actual label and t is the predicted output.

3.3 Principal Component Analysis

Principal component analysis is a technique for extracting features that mixes our input variables in a certain way, allowing us to eliminate the least important variables while keeping the most valuable bits of all the variables. Additionally, following PCA, each of the new variables is independent each other. Because the assumptions of a linear model demand that our independent variables be independent of one another, this is a added benefit that PCA provides. PCA helps to overcome overfitting problem, improves performance of model by eliminating correlated variables, it also speeds up the model training as it reduces the number of dimension.

3.4 Proposed Work

This research proposes a unique approach for detecting fingerprint presentation attacks based on the hybridization of Deep-learning and Machine-learning. When significant characteristics of the pictures are provided to the SVM, it is proved to be a very effective tool for image classification is shown in [6], [7].

Although the extraction of those characteristics involves certain image processing actions. Some of the features are Ridge width smoothness, Local Binary Pattern and Orientation certainty level that has been utilized by various authors in order to detect the spoof fingerprint samples. The main disadvantage of these approaches is that they rely on other pre-processing activities and are confined to a certain data set. When working on a fingerprint acquired with another sensing device, the performance with several features is not the same. The results in [7] clearly show that handmade features combined with SVM do not function well for samples generated with the same sensor. However, this inefficiency is not due to the SVM; rather, it is owing to the limits of the hand-crafted feature. On the other hand, numerous strategies discussed in [3],[2] displays Deep convolution networks capacity to identify presentation assaults. CNN's success can be ascribed to the use of convolution filters, which can extract minute details from fingerprint images.

The textures of the real and false fingerprint samples differ due to natural features such as flexibility, sweat and pores, dampness, and so on. The results show that deep-learning-based approaches are more generic and that sensing device modification has no effect on their performance; nonetheless, they suffer from the difficulty of cross-sensor and cross-material fingerprint presentation attack detection. Looking at the capabilities of both tools, we chose to create a hybrid architecture that combines the strengths of both approaches, with Deep CNN excelling at feature extraction and SVM excelling at classification based on feature values.

The suggested architecture is made up of two parts: the Feature Extractor and the Classifier. In our study, the feature extractor is a MobileNet-v1 Convolutional Neural Network that was trained from scratch on standard fingerprint datasets, and the classifier is a Support Vector Machine. The model is designed to train and update weights as well as classify fingerprint samples from start to finish. However, as proposed in many models, another option to integrate both techniques is to train the CNN first and then extract the feature values to feed the SVM for classification.

This method works effectively for detecting IRIS presentation attacks but not for fingerprints since fingerprint pictures lack texture and other elements. All of the aforementioned factors compelled us to create an end-to-end architecture that incorporates the features of machine learning and deep learning models. The subsections below discuss the intricacies of all the major components, as well as their assembly and coordination toward the objective of detecting phony fingerprint samples.

3.4.1 Static Hybridized Architecture

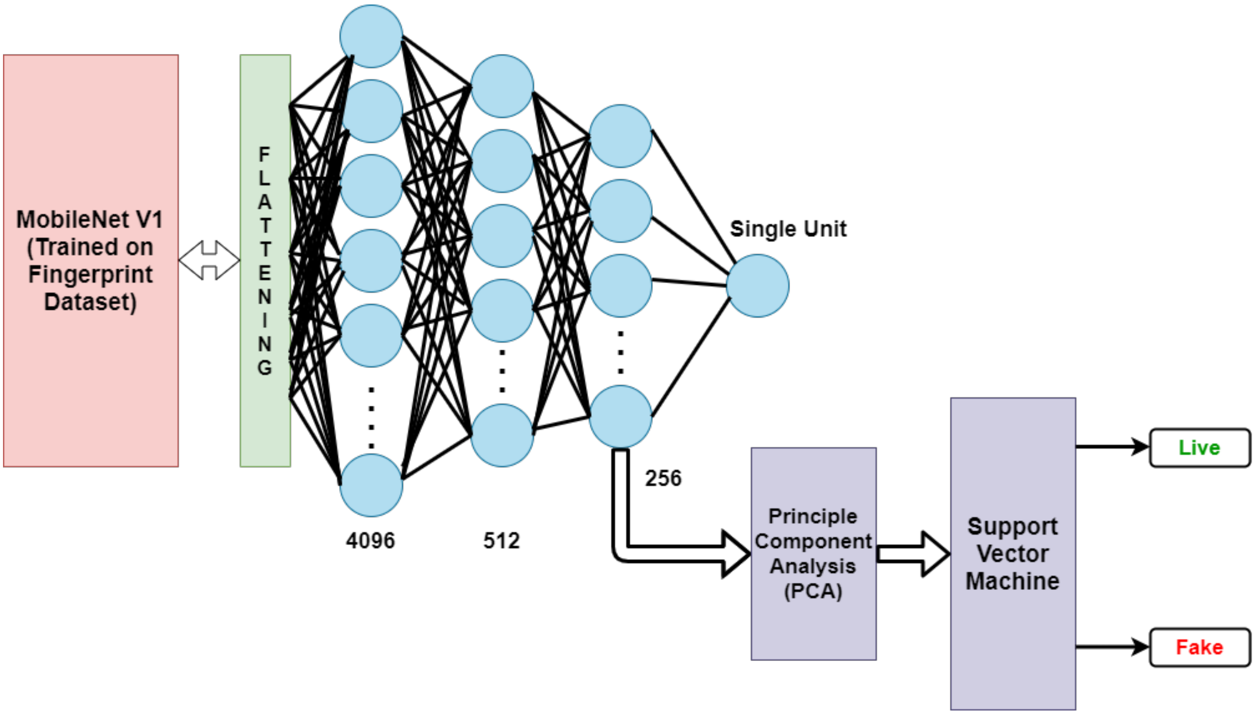
MobileNet V1 (Trained on Fingerprint Dataset)
FLATTENING
4096
512
256
Single Unit
Principle Component Analysis (PCA)
Support Vector Machine
Live
Fake

FIGURE 3.5: Static Collaboration of MobileNet V1 and Support Vector Machine.

In this Architecture, Deep CNN architectures are used as a feature extractor, while machine learning is used as a classifier. Both are taught independently in this scenario, and there is no functional relationship between them. In some work, pre-trained architectural features are tapped at a specific layer of the architecture and given to the SVM for training and testing. This is referred to as static cooperation. The findings obtained by these models are provided. This approach consists of two phases.

The current Mobile-net v1 model is initially trained for 100 iterations on standard fingerprint datasets, and the weights of filters and dense layers are kept for the epoch with the greatest classification accuracy. In the second stage, the network is tapped on the second last layer while sending the training and testing data to this model, resulting in the production of CSV files for the training and testing datasets. The data values from the previous stage are supplied into the SVM to train and test it in the final step. For the dimensionality reduction of the input data, we used Principle Component Analysis (PCA). 3.5 depicts the block diagram of this approach.

3.4.2 Dynamic Hybridized Architecture

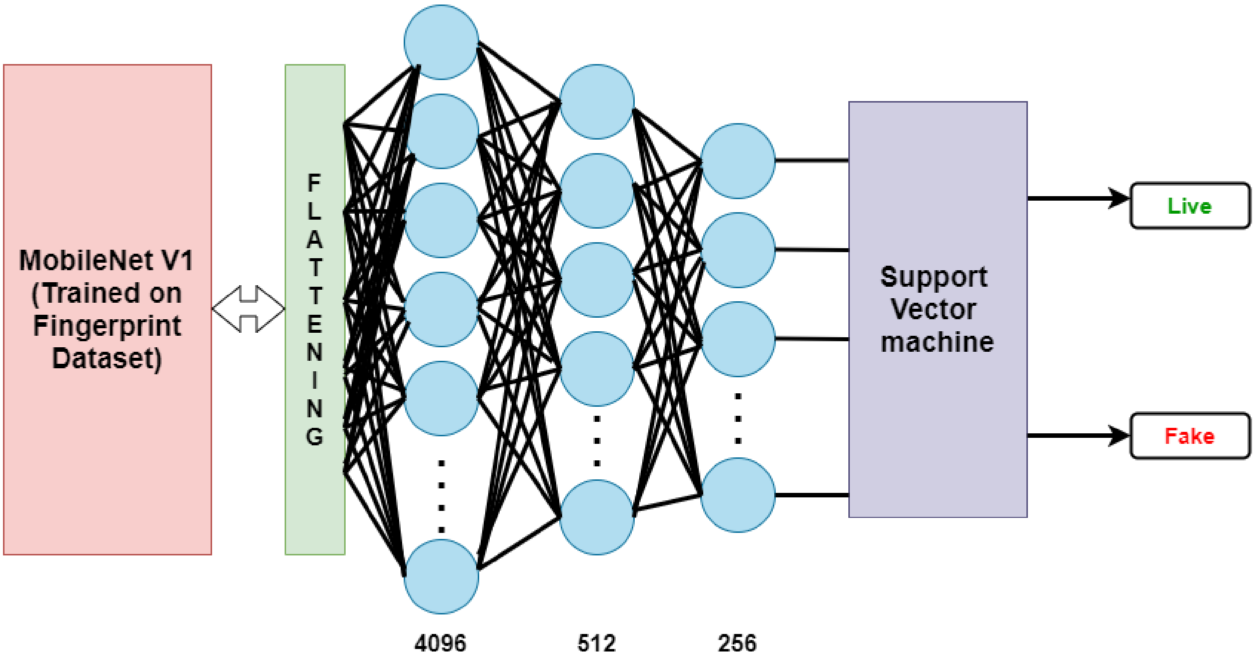

FIGURE 3.6: Dynamic Collaboration of MobileNet V1 and Support Vector Machine.

We propose a dynamic combination of MobileNet V1 and the Support Vector Machine in this work. It is a complete architecture in which both tools are educated simultaneously. End-to-end (E2E) learning is the method of training a potentially complex learning system with a single model (usually a Deep Neural Network) that represents the whole target system, without the intermediary layers observed in traditional pipeline designs.

As a feature extractor, MobileNet v1 harvests minute features and trains according to the loss estimated by the binary Cross entropy function (in the static method) and Hinge loss (In dynamic approach). The spoofness score is then calculated using the retrieved feature values. Both models are trained in the static manner, however only the deep CNN is trained in the dynamic approach by the loss caused by the SVM.

By taking characteristics from a deep CNN architecture and putting them into an SVM, we may combine them. Some other projects have used pre-trained deep CNN architectures for feature extraction and then fed them to machine learning classifiers in the IRIS biometric system to identify presentation attacks. However, while these solutions function better for IRIS biometric systems, they do not perform as well for fingerprint biometric systems. Apart from this constraint, these approaches also have the drawback of not being end-to-end in nature. Keeping all of these considerations in mind, In the above Figure 3.6. We proposed a unique architecture that works end-to-end and has higher performance for detecting fingerprint presentation attacks.

Chapter 4

Experimental Results

4.1 Dataset

The algorithm's performance is evaluated in this section using three datasets totaling ten sensors. The performance of the proposed model is assessed using the LivDet liveness detection fingerprint data sets from 2013, 2015, and 2017. These fingerprint samples are separated into training and testing sets. The sensing equipment, the quantity of training and testing live and false samples, as well as the name of spoofing materials are all given in the 4.1.

Datasets	Sensor	Spoofing Materials
LivDet 2013	Biometrika	Ecoflex, Gelatin, Latex, Silicone, Wood Glue
	Italdata	Ecoflex, Gelatin , latex, Silicone, Wood Glue
	Crossmatch	Body Double, Latex, , Play-Doh ,Wood Glue ,
LivDet 2015	Crossmatch	Body Double, Eco Flex, Play-Doh,OOMOO ,Gelatin
	Digper	Ecoflex, Latex, Gelatine, Wood Glue, Liquid Ecoflex, RTV
	Greenbit	Ecoflex, Latex, Gelatine, Wood Glue, Liquid Ecoflex, RTV
	Biometrika	Ecoflex, Latex, Gelatine, Wood Glue, Liquid, Ecoflex, RTV
LivDet 2017	Oracnathus	Body Double, Ecoflex,Latex, Wood Glue , Gelatine, Liquid ecoflex
	Digper	Body Double, Ecoflex, Wood Glue, Gelatine, Gelatine, Latex, Liquid, Ecoflex
	Greenbit	Body Double, Ecoflex, Latex , Wood Glue , Gelatine ,Latex ,Liquid ,Ecoflex

TABLE 4.1: Datasets LivDet 2013, LivDet 2015, LivDet 2017 with Corresponding Sensor and spoof material used

4.2 Performance Metrics

The Attack Presentation Classification Error Rate (APCER) measures the proportion of misclassified spoof fingerprint photos, whereas the Bonafide Presentation Classification Error Rate (BPCER) measures the proportion of misclassified real fingerprint images.
Average classification error (ACE), which is the averaged total of APCER and BPCER, is used to evaluate the system's overall performance. The following is an equation that represents the calculation of ACE.

$$ACE = \frac{APCER + BPCER}{2}$$

We can derive accuracy by : Average Accuracy = 100 - ACE

4.3 Implementation Details

4.3.1 Hardware Specification

The proposed model is implemented using Keras library. All training and testing has been done over NVIDIA TESLA P100 GPU. The Cross-sensor validation is performed by using the testing data captured with the sensor other than which is used for the preparation of training data.

4.3.2 Static Hybridisation of MobileNet and SVM

We used keras library for all our implementations. For static hybridisation we followed our proposed model architecture 3.5.
We started by importing MobileNet architecture availbale in Keras and removing the top layer which was meant for 1000 classes ImageNet problem. We kept half of the layers of MobileNet as non-trainable i.e. with the weights it has from being trained on ImageNet dataset and half of the layers as trainable i.e. they can change their weights according to our model. Thereafter we added the Dense and Dropout layers as shown in 3.5. On our dataset of LivDet we separated them into two folders of training and testing both having fake and live sub-folders and fed them using Image Data Generator from Keras. This automatically assigns labels to the fake and live images. For the training dataset we used Image Augmentation and preprocessing techniques. The model above is trained using Adam Optimiser and Binary Cross Entropy loss. The model is trained for over 100 epochs and then the dense layer with 256 neurons is tapped for feature extraction. We do feature extraction by using keras methodology of extracting output from any named layer. The output from the dense layer is fed to PCA(Principal Component Analysis) so as to retain the important nodes and remove the unimportant ones and the dimesion is reduces to 100. After this we use Sklearn scikit library and pass the output from PCA into a SVM Classifier for training. The accuracies obtained by this method is reported in the results section table for LivDet 2013, LivDet 2015 and LivDet 2017 dataset.

4.3.3 Dynamic Hybridisation of MobileNet and SVM

All of our implementations used the Keras library. We used our proposed model architecture for dynamic hybridization 3.6.
We began by importing the available MobileNet architecture in Keras and eliminating the top layer, which was intended for the 1000-class ImageNet challenge. We left half of MobileNet's layers non-trainable, i.e. with the weights it learned from the ImageNet dataset, and the other half trainable, i.e. able to adjust their weights according to our model. After that, as indicated in 3.6, we added the Dense and Dropout layers. For introducing SVM in our CNN model we added a Dense layer with hinge loss and the kernel regularizer as L2 regularizer. The final layer of our CNN would act as a linear SVM for binary classification of the images as live or fake.

4.3.4 Image Augmentation

We augmented the given training images by providing our Image Data Generator with parameters. We performed rotation, horizontal flipping, scaling, zca whitening, zoom and shear operations. We also shuffled our images so as to introduce more randomness that would make the model more generic.

4.4 Results

The resulting accuracies from both of the above approaches were computed completely for all three datasets over ten sensors. Their performance is compared to existing State-of-the-Art approaches. We tested the suggested model's performance in three different scenarios. Training and testing fingerprint samples may be arranged in two ways under these conditions: intra-sensor same material and intra-sensor cross-material. All of these eventualities are described in detail below:

4.4.1 Intra-Sensor and Known Spoof Material

The fake samples for both the training and testing sets are created with the same fabrication materials, and the training and testing sets use the same sensing equipment. **LivDet 2013** data set that we used for validation is part of this setup, in which training and testing false samples are constructed utilising the materials, i.e. the spoofing materials' names. An other data-set used by us, **LivDet 2015** also falls into this category, although only about two-thirds (66%) of the training and testing phoney samples are made with the same spoof components.

Results for LivDet 2013 Dataset

The Graphs for loss and accuracy for all the sensors of LivDet 2013 dataset are provided in this section. We also provide a comparitive analysis with the State-of-the-art methods for LivDet 2013 Dataset.

S.no.	Method name and reference	Biometrika	Crossmatch	Italdata	Average
1	C. Yuan et al. [11]	99.25	97.53	99.40	98.72
2	Y. Zhang et al.[12]	99.53	-	96.99	98.26
3	H. Jung et al. [13]	94.12	99.56	97.92	96.02
4	R. Nogueira et al. [14]	99.20	96.71	97.7	98.45
5	C. Gottsc hlich et al. [15]	96.10	-	98.30	97.2
	Proposed Method	**99.59**	**97.67**	**99.39**	**98.88**

TABLE 4.2: Comparison with the State-of-the-art Methods on LivDet 2013.

Dataset	Sensor	Mobilenet	Static Hybrid Model	Dynamic Hybrid Model
2013	Biometrika	96.49	92.40	99.59
	Italdata	92.95	85.50	99.39
	Crossmatch	87.41	88.08	97.67
	Average	**92.11**	**88.66**	**98.88**

TABLE 4.3: Accuracy of Mobilenet , Static and Dynamic Models on LivDet 2013

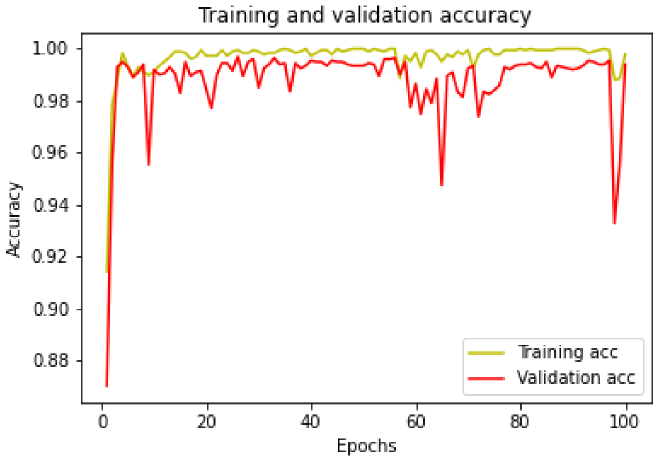

FIGURE 4.1: LivDet 2013 Biometrika Accuracy

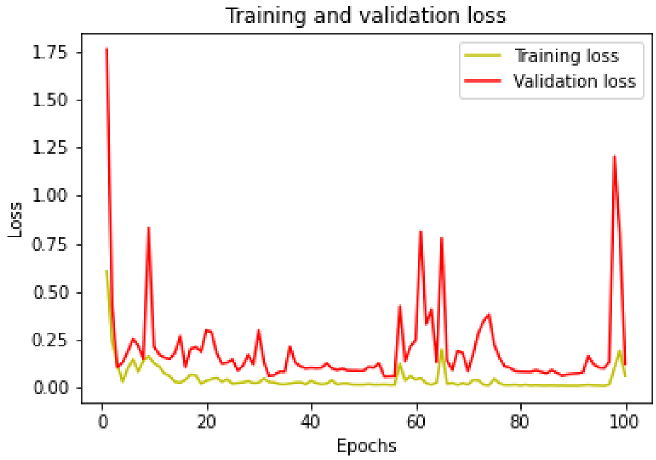

FIGURE 4.2: LivDet 2013 Biometrika Loss

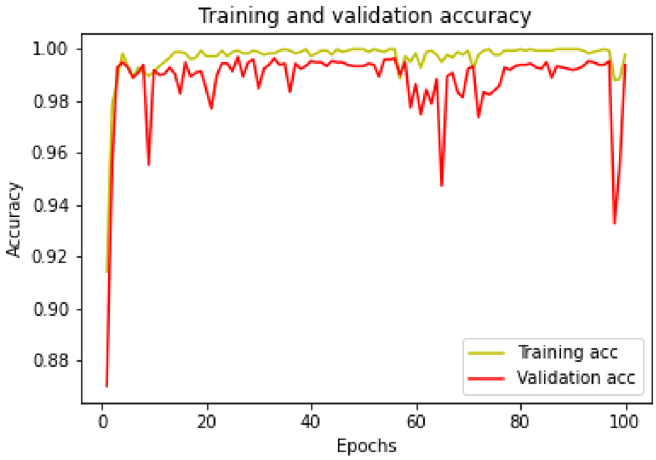

FIGURE 4.3: LivDet 2013 Italdata Accuracy

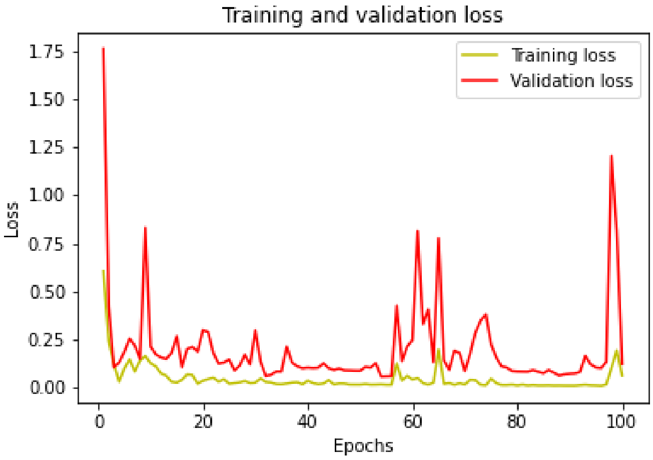

FIGURE 4.4: LivDet 2013 Italdata Loss

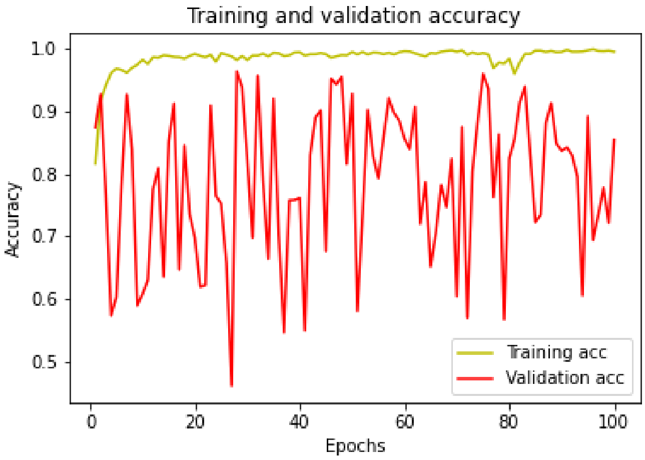

FIGURE 4.5: LivDet 2013 CrossMatch Accuracy

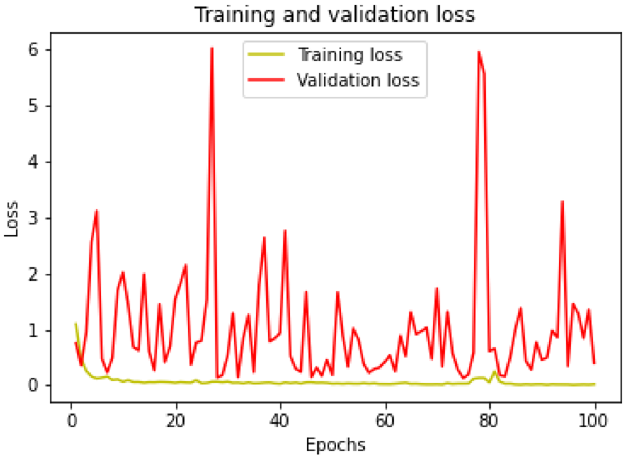

FIGURE 4.6: LivDet 2013 CrossMatch Loss

LivDet 2015

The Graphs for loss and accuracy for all the sensors of LivDet 2015 dataset are provided in this section. We also provide a comparitive analysis with the State-of-the-art methods for LivDet 2015 dataset.

Sr. no.	Method name and reference	Biometrika	Greenbit	Crossmatch	Digper	Average
1	Yuan et al. [11]	-	-	-	-	88.99
2	Kim et al. [16]	-	-	-	-	86.39
3	Xia et al. [17]	90.36	89.18	86.28	95.47	90.32
4	G. Huang et l. [18]	95.63	96.32	93.33	89.10	93.59
5	Chugh et al. [3]	-	-	-	-	**99.03**
	Proposed Method	**96.67**	**96.75**	**98.42**	**95.11**	**96.75**

TABLE 4.4: Comparison with the State-of-the-art Methods on LivDet 2015

Dataset	Sensor	Mobilenet	Static Hybrid Model	Dynamic Hybrid Model
2015	Biometrika	94.75	84.64	96.67
	Digper	91.90	83.88	95.11
	Greenbit	94.52	86.77	96.75
	Crossmatch	96.48	89.42	98.42
	Average	**94.41**	**86.17**	**96.75**

TABLE 4.5: Accuracy of Mobilenet , Static and Dynamic Models on LivDet 2015

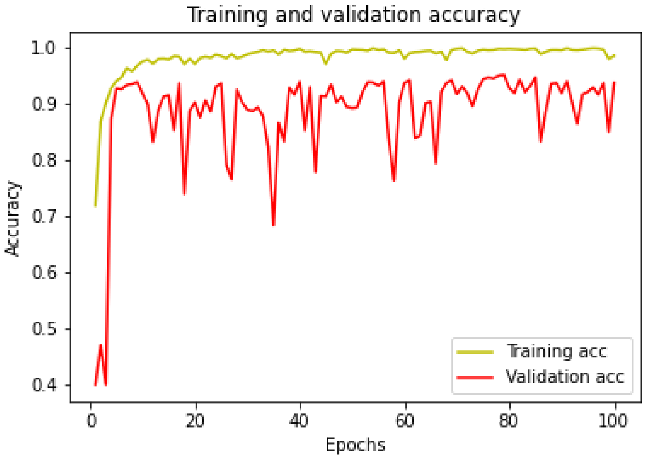

FIGURE 4.7: LivDet 2015 Biometrika Accuracy

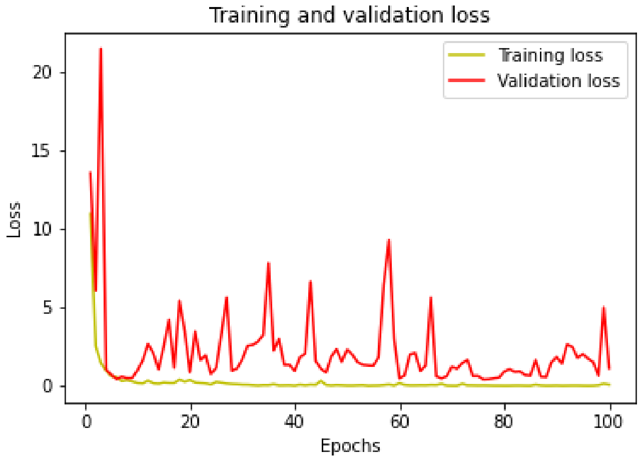

FIGURE 4.8: LivDet 2015 Biometrika Loss

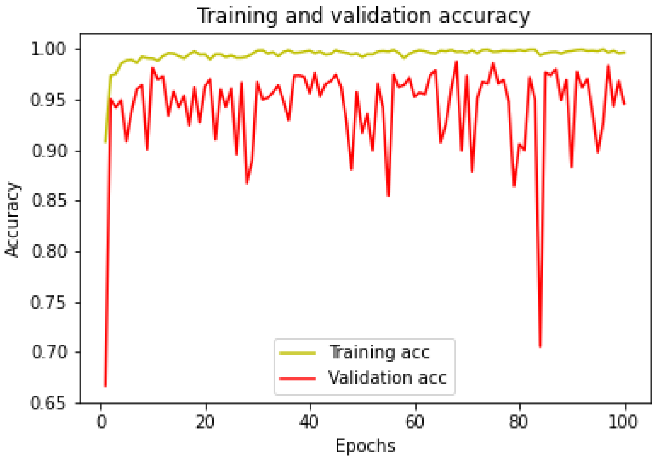

FIGURE 4.9: LivDet 2015 CrossMatch Accuracy

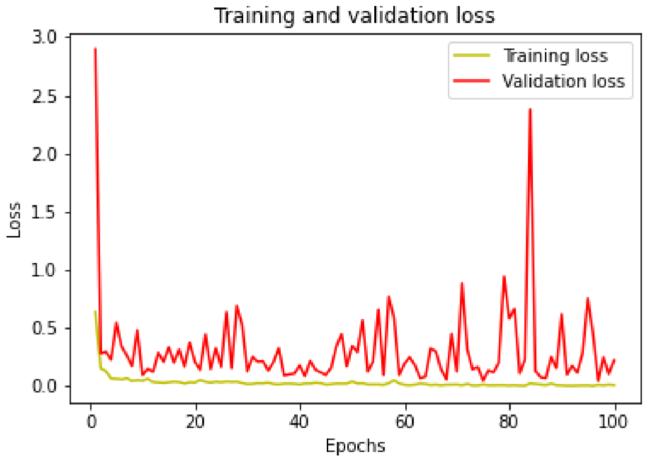

FIGURE 4.10: LivDet 2015 CrossMatch Loss

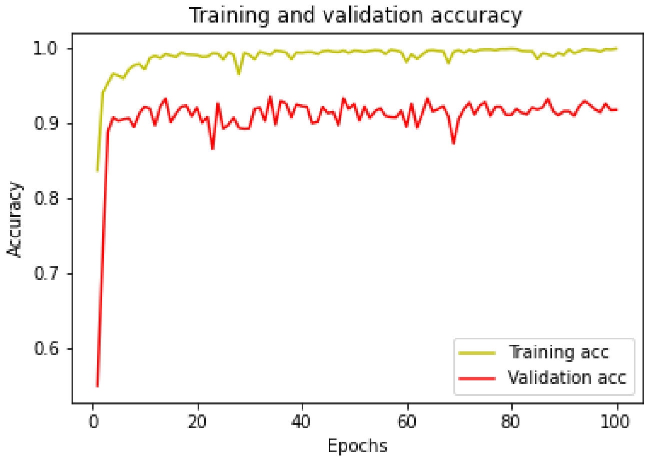

FIGURE 4.11: LivDet 2015 Digper Accuracy

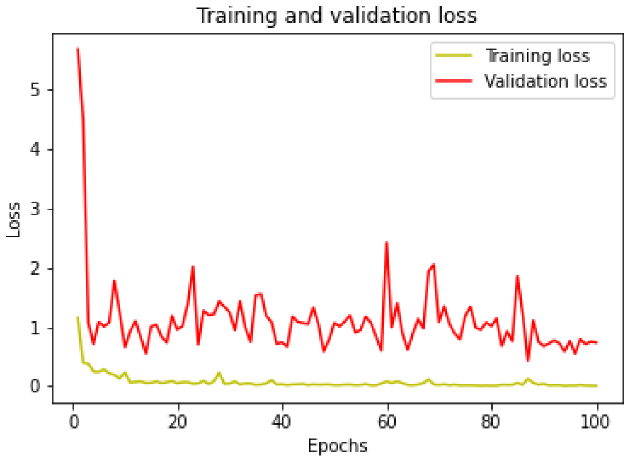

FIGURE 4.12: LivDet 2015 Digper Loss

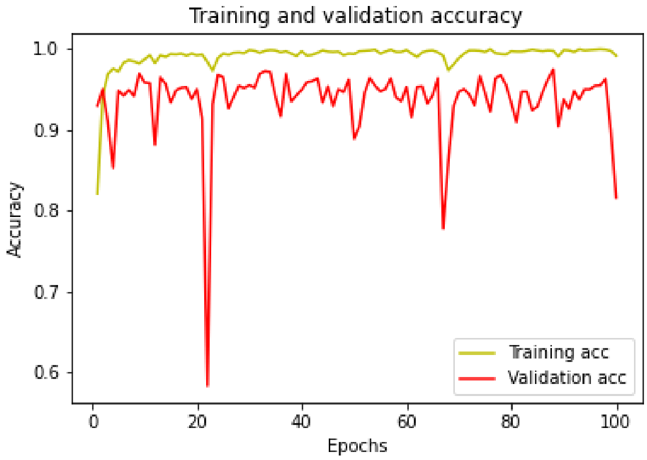

FIGURE 4.13: LivDet 2015 GreenBit Accuracy

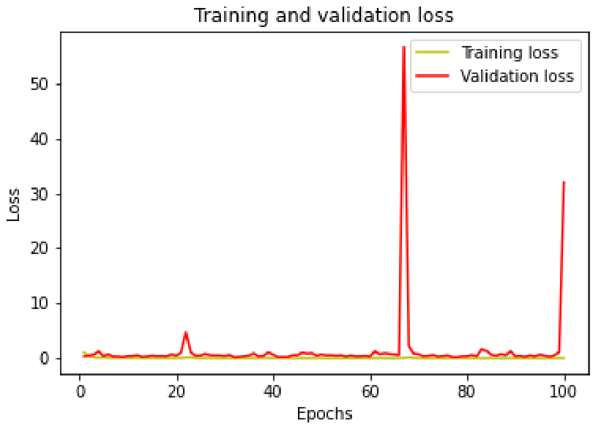

FIGURE 4.14: LivDet 2015 GreenBit Loss

4.4.2 Intra-Sensor and Unknown Spoof Material

In this approach, training and testing samples are collected using the same sensing equipment, but bogus fingerprint samples for the training and testing data sets are created using different spoofing materials. In this setup, any FPAD model's performance should be better since it shows the model's ability to detect fake samples in a real-world setting where new and affordable spoofing materials are discovered every day. The **LivDet 2015** has one-third (33%) of the training and testing fake samples in the training and testing set are made with different spoof materials, this data-set falls into this category. Another set of data was utilised to test the suggested model. Since the training and testing fake samples are made of different materials, **LivDet 2017** belong under this category altogether.

LivDet 2017

The Graphs for loss and accuracy for all the sensors of LivDet 2017 dataset are provided in this section. We also provide a comparitive analysis with the State-of-the-art methods for LivDet 2017 dataset.

Sr. no.	Method name and reference	GreenBit	Orcanthus	DigPer	Average
1	T. Chung et al. [3]	96.68	94.51	95.12	**95.44**
2	Y. Zhang et al. [12]	95.20	93.93	92.89	**94.01**
	Proposed Method	94.16	95.00	95.55	**94.90**

TABLE 4.6: Comparison of Proposed Dynamic Model on Dataset 2017 with State-of-the-art Methods.

Dataset	Sensor	Mobilenet	Static Hybrid Model	Dynamic Hybrid Model
2017	Oracnathus	88.71	80.79	95.00
	Digper	91.64	81.63	95.55
	Greenbit	90.94	83.67	94.16
	Average	**90.43**	**82.03**	**94.90**

TABLE 4.7: Accuracy of Mobilenet , Static and Dynamic Models on LivDet 2017

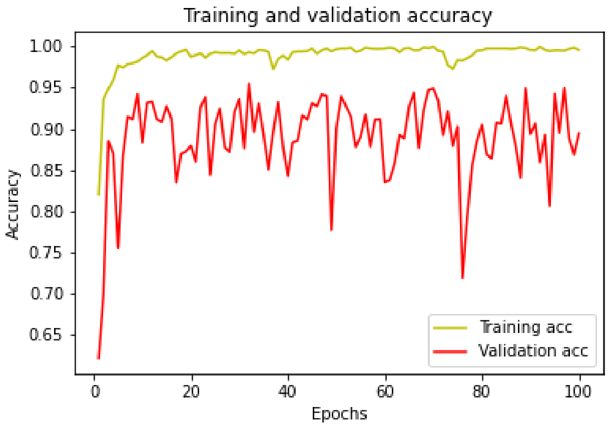

FIGURE 4.15: LivDet 2017 Orcanthus Accuracy

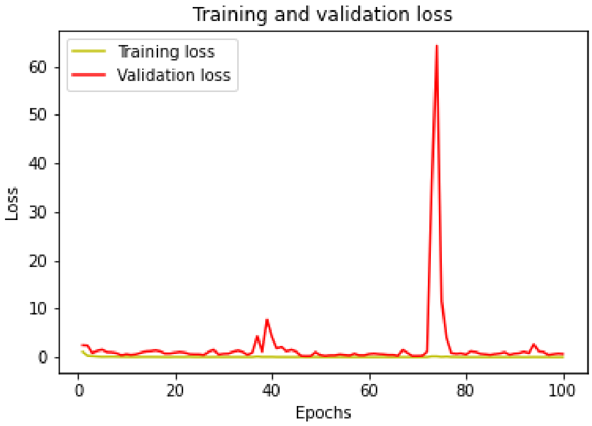

FIGURE 4.16: LivDet 2017 Orcanthus Loss

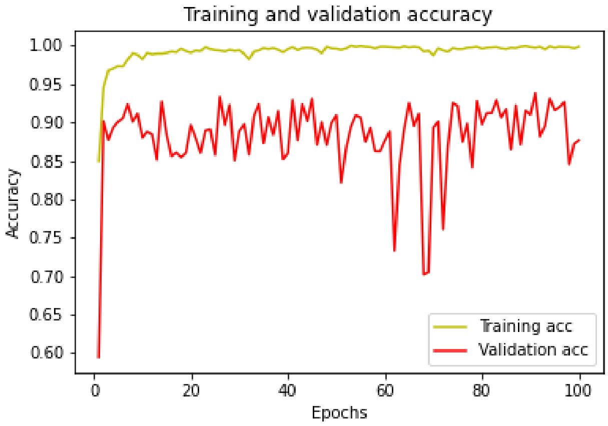

FIGURE 4.17: LivDet 2017 Digper Accuracy

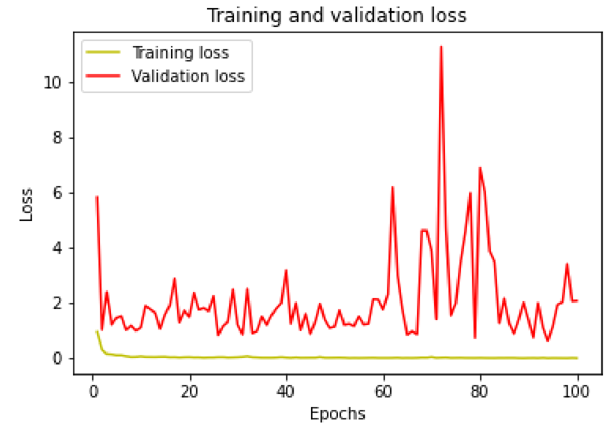

FIGURE 4.18: LivDet 2017 Digper Loss

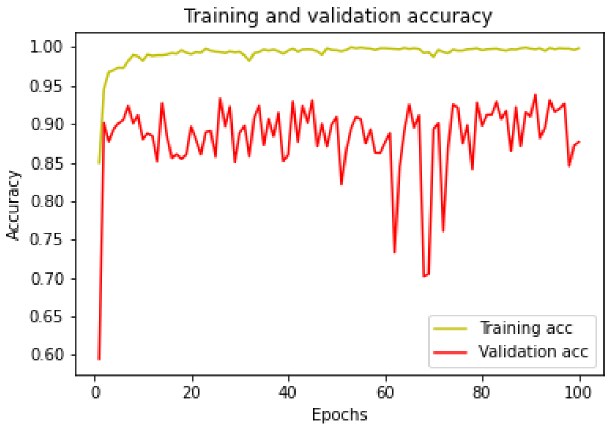

FIGURE 4.19: LivDet 2017 GreenBit Accuracy

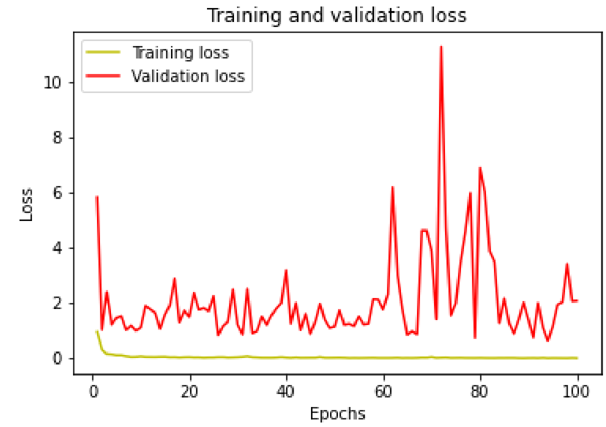

FIGURE 4.20: LivDet 2017 GreenBit Loss

Chapter 5

Conclusion and Future Work

Starting with the analysis phase, understanding the need from real-world use cases, the in-depth technical knowledge required to develop the machine learning model, preparing the dataset for training the model, understanding and identifying the features that are required for training the model, selecting the right algorithm for preparing the model, and evaluating the machine learning model, this report has assisted us in understanding the process required to build a deep learning model.

In this project, we have proposed a new method of Presentation Attack Detection in Fingerprint Biometrics. Our Dynamic Hybridisation of SVM and MobileNet outperforms the many models in LivDet 2013, 2015 and 2017 datasets.

Future work in this direction would be to use some other classifiers like Random Forest instead of SVM and see the performance of the model.
The calculation for 2019 LivDet and cross-sensor paradigm will also be performed.
Also as we have proposed an end-to-end architecture we will develop a GUI application that can be used for Presentation Attack Detection.

Bibliography

[1] Shefali Arora. “Fingerprint Spoofing Detection to Improve Customer Security in Mobile Financial Applications Using Deep Learning”. In: *Arabian Journal for Science and Engineering* 45 (Oct. 2019). DOI: 10.1007/s13369-019-04190-1.

[2] B. Anusha, Sayan Banerjee, and Subhasis Chaudhuri. “DeFraudNet:End2End Fingerprint Spoof Detection using Patch Level Attention”. In: (Feb. 2020).

[3] Tarang Chugh and Kai Cao. “Fingerprint Spoof Buster: Use of Minutiae-Centered Patches”. In: *IEEE Transactions on Information Forensics and Security* PP (Mar. 2018), pp. 1–1. DOI: 10.1109/TIFS.2018.2812193.

[4] Rodrigo Nogueira, Roberto Lotufo, and Rubens Machado. “Evaluating software-based fingerprint liveness detection using Convolutional Networks and Local Binary Patterns”. In: (Aug. 2015). DOI: 10.1109/BIOMS.2014.6951531.

[5] Rohit Kumar Dubey, Jonathan Goh, and Vrizlynn L. L. Thing. “Fingerprint Liveness Detection From Single Image Using Low-Level Features and Shape Analysis”. In: *IEEE Transactions on Information Forensics and Security* 11.7 (2016), pp. 1461–1475. DOI: 10.1109/TIFS.2016.2535899.

[6] Ajita Rattani and Arun Ross. “Automatic adaptation of fingerprint liveness detector to new spoof materials”. In: *IEEE International Joint Conference on Biometrics*. 2014, pp. 1–8. DOI: 10.1109/BTAS.2014.6996254.

[7] Ram Sharma and Somnath Dey. “Fingerprint image quality assessment and scoring using minutiae centered local patches”. In: *Journal of Electronic Imaging* 28 (Jan. 2019), p. 1. DOI: 10.1117/1.JEI.28.1.013016.

[8] Javier Galbally, Sébastien Marcel, and Julian Fierrez. “Image Quality Assessment for Fake Biometric Detection: Application to Iris, Fingerprint, and Face Recognition”. In: *IEEE Transactions on Image Processing* 23.2 (2014), pp. 710–724. DOI: 10.1109/TIP.2013.2292332.

[9] Zhihua Xia et al. “Fingerprint liveness detection using gradient-based texture features”. In: *Signal, Image and Video Processing* 11 (Feb. 2017). DOI: 10.1007/s11760-016-0936-z.

[10] Ajita Rattani, Walter J. Scheirer, and Arun Ross. “Open Set Fingerprint Spoof Detection Across Novel Fabrication Materials”. In: *IEEE Transactions on Information Forensics and Security* 10.11 (2015), pp. 2447–2460. DOI: 10.1109/TIFS.2015.2464772.

[11] Chengsheng Yuan, Xingming Sun, and Q. M. Wu. “Difference Co-Occurrence Matrix Using BP Neural Network for Fingerprint Liveness Detection”. In: *Soft Comput.* 23.13 (2019), 5157–5169. ISSN: 1432-7643. DOI: 10.1007/s00500-018-3182-1. URL: https://doi.org/10.1007/s00500-018-3182-1.

[12] Yongliang Zhang et al. “Slim-ResCNN: A Deep Residual Convolutional Neural Network for Fingerprint Liveness Detection”. In: *IEEE Access* 7 (2019), pp. 91476–91487. DOI: 10.1109/ACCESS.2019.2927357.

[13] Ho Yub Jung, Yong Heo, and Soochahn Lee. “Fingerprint Liveness Detection by a Template-Probe Convolutional Neural Network”. In: *IEEE Access* PP (Aug. 2019), pp. 1–1. DOI: 10.1109/ACCESS.2019.2936890.

[14] Rodrigo Frassetto Nogueira, Roberto de Alencar Lotufo, and Rubens Campos Machado. “Fingerprint Liveness Detection Using Convolutional Neural Networks”. In: *IEEE Transactions on Information Forensics and Security* 11.6 (2016), pp. 1206–1213. DOI: 10.1109/TIFS.2016.2520880.

[15] Carsten Gottschlich et al. “Fingerprint Liveness Detection based on Histograms of Invariant Gradients”. In: *IJCB 2014 - 2014 IEEE/IAPR International Joint Conference on Biometrics* (Sept. 2014). DOI: 10.1109/BTAS.2014.6996224.

[16] Wonjun Kim. “Fingerprint Liveness Detection Using Local Coherence Patterns”. In: *IEEE Signal Processing Letters* 24.1 (2017), pp. 51–55. DOI: 10.1109/LSP.2016.2636158.

[17] Zhihua Xia et al. “A Novel Weber Local Binary Descriptor for Fingerprint Liveness Detection”. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 50.4 (2020), pp. 1526–1536. DOI: 10.1109/TSMC.2018.2874281.

[18] G. Huang et al. “Densely Connected Convolutional Networks”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, 2017, pp. 2261–2269. DOI: 10.1109/CVPR.2017.243.