Presentation Attack Detection in Biometric Security Using Deep Learning Techniques - A Comparative Analysis

MSDSM Project Report

By
Dev Nandan Sarkar

INDIAN INSTITUTE OF TECHNOLOGY INDORE
INDIAN INSTITUTE OF MANAGEMENT INDORE

JUNE 2024

Presentation Attack Detection in Biometric Security Using Deep Learning Techniques - A Comparative Analysis

A PROJECT

Submitted in partial fulfilment of the
requirements for Term VI
of
Master of Science in Data Science and Management

Submitted by
Dev Nandan Sarkar
2204107017

INDIAN INSTITUTE OF TECHNOLOGY INDORE
INDIAN INSTITUTE OF MANAGEMENT INDORE

JUNE 2024

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the project entitled **Presentation Attack Detection in Biometric Security Using Deep Learning Techniques - A Comparative Analysis** in the partial fulfilment of the requirements for term six of **MASTER OF SCIENCE IN DATA SCIENCE AND MANAGEMENT , JOINTLTY OFFERED BY Indian Institute of Technology Indore** and **Indian Institute of Management Indore** is an authentic record of my own work carried out during the time period from April 2024 to June 2024 under the guidance of Associate Professor Dr. Somnath Dey.

The matter presented in this project has not been submitted by me for the award of any other degree of this or any other institute.

19 June 2024

Signature of the student with date

This is to certify that the above statement made by the student is correct to the best of my/our knowledge.

Signature of the Guide of MSDSM Project

(Dr. Somnath Dey)

Dev Nandan Sarkar has successfully given his/her MSDSM Oral Examination held on **June 19 2024**.

Signature(s) of Guide

Date:

Signature of Faculty

Date:

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my project guide Dr. Somnath Dey, Associate Professor, IIT Indore, who has mentored, supported and encouraged me in completion of this work on Presentation Attack Detection in Biometric Security using Deep Learning Techniques -A Comparative Analysis. His guidance was critical for the achievement of the objectives of this work.

I would also like to extend my thanks to Mr. Anuj Rai who has provided valuable insights and hands-on guidance to me for the execution of this work.

I would also like to thank the Indian Institute of Technology Indore, for providing me the opportunity and ecosystem that has helped me complete this work.

Dev Nandan Sarkar
Master of Data Science and Management Batch 2
IIT Indore

DEDICATION

I would like to dedicate this work to my parents, my wife and my daughter.

Abstract

Fingerprint biometrics is frequently used as the unique personal identifier in biometric authentication systems. They are also utilized in multifactor authentication processes. Due to the wide acceptance and usage of fingerprints as a reliable identifier for user authentication it is also most exposed to presentation attacks. Presentation attacks are deployed to fool a system using a forged replica of a biometric identifier, such as a forged fingerprint made from different materials, in order to deceive the system into believing that a genuine user is requesting access. In order for a system to detect that it is under presentation attack it needs capabilities, either in its hardware or in its software (or both), that can distinguish between a genuine biometric artefact and a forged one. In case of fingerprint biometrics, enhancement of hardware would require enhancement of the fingerprint sensor capabilities to detect temperature, blood pressure and moisture presence on the finger. Such enhancements would be capital intensive and we leave the exploration of those techniques as out of the scope of this work. In this work, we have focused on the techniques relating to development and enhancement of software capabilities that can help existing systems distinguish between a genuine fingerprint and a forged one. Previous works have shown successful use of hand-crafted features for spoof detection. In this work, we have focussed on usage of various Deep Learning models, without the use of handcrafted features, on different datasets and have presented a comparative analysis of the findings. The dataset that we have utilized for the purposes of training and testing our models is LivDet-2011. A standard deep learning model that we have studied in this work is VGG16. Apart from this we have studied a few shallow CNN architectures and compared their performance with standard architectures. Our findings reveal that the VGG16 model performs with intra-sensor testing accuracy of 91.79% and a cross-sensor testing accuracy of 58%. In comparison, one of the presented shallow CNN model exceeds this intra-sensor testing accuracy by achieving 94.72% accuracy and the other shallow CNN model falls only slightly short of this cross-sensor testing accuracy by achieving an accuracy of 55.29%. This shows that shallow CNN architectures have the potential to perform at par with the standard deep CNN architectures, like VGG16, in the area of Fingerprint Presentation Attack Detection.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

Acronym	Expansion
FRS	Fingerprint Recognition System
FPAD	Fingerprint Presentation Attack Detection
CNN	Convolution Neural Network
ATM	Automated Teller Machine
ARL	Adversarial Representation Learning
UMG	Universal Material Generator
TDR	True Detection Rate
FDR	Fake Detection Rate
SWIR	Short Wave Infrared
RE	Reconstruction Error
SGD	Stochastic Gradient Descent

Chapter 1

Introduction

1.1 Fingerprint Biometrics

Biometrics is the study of statistical traits and measurements of physical characteristics of human beings. Examples are fingerprints, iris, face, voice, DNA etc. Certain statistical measurements of these physical characteristics can help uniquely identify (and thereby authenticate) a person. Therefore, biometrics finds immense utility in authentication and access control applications across the world.

Fingerprints are most widely used biometric data in such authentication systems given its reliability, robustness, ease of use and relatively low cost of application. Figure 1.1 shows a sample fingerprint.

Figure 1.1: A sample of a fingerprint from LivDet-2011 dataset, Biometrika sensor

1.2 Fingerprint Recognition System

A Fingerprint Recognition System (FRS) [16] works in the following way – users' fingerprint data is collected through sensors and their features are extracted and recorded into a database. When an authentication request comes from a new fingerprint, the FRS [16] extracts its feature vector and tries to match it with the existing records present in the database. If a match is found, the request is authenticated. Figure 1.2 shows a block diagram representation of how a FRS [16] works.

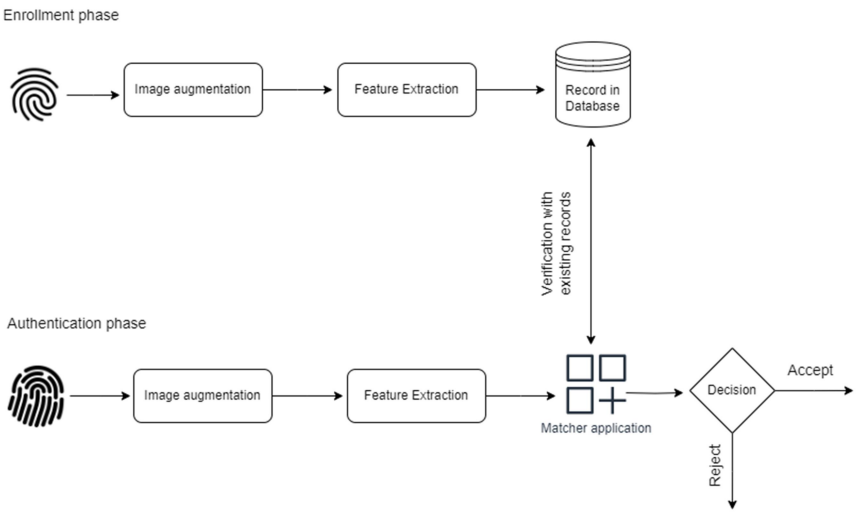

Figure 1.2: Block diagram representation of a Fingerprint Recognition System

FRSs [16] are common-place in today's day and age. It has found applicability in various identity / security systems such as Aadhaar linked systems, Visa processing, Immigration, Automated Teller Machines (ATM) etc. Given its usage in many areas FRSs [16] are also exposed to wide range of identity-fraud attacks. Such attacks are called Presentation Attacks (PA) as an attacker tries to impersonate a genuine enrolled user. PAs are carried out by trying to fool the FRS [16] using a forged fingerprint made from artificial materials like ecoflex, play-doh and wood glue etc. Samples of a genuine fingerprint and forged finger prints are shown in Fig. 1.3 and Fig. 1.4 respectively.

Figure 1.3: A sample of a genuine fingerprint from LivDet-2011 dataset, Biometrika sensor

Figure 1.4: Samples of forged fingerprints from LivDet-2013 dataset, Biometrika sensor, made from various materials

1.3 Presentation Attack Detection

Presentation Attack Detection mechanisms are employed to detect fraudulent or fabricated biometric input. The Fingerprint Presentation Attack Detection (FPAD) can be achieved through both - hardware enhancements as well as smart software. Hardware enhancements are expensive as they involve additional sensors to detect liveliness using moisture, temperature and pulse of the input finger. Solutions using software are cheaper to develop as this approach only needs a handful of good quality images of the fingerprints, which can then be used to create software that can distinguish between a genuine and a fake.

Software solutions are sub-divided into two categories. One is based on mathematically established measures regarding the physical characteristics of fingerprints. These measures are called hand-crafted features. This method uses these hand-crafted features to discriminate between genuine and fake fingerprints. The other method is based on employment of deep learning techniques to learn these discriminating feature vectors. The focus of this work is on the latter.

1.4 Objectives

There are two objectives of this work. First is to use a state-of-the-art CNN model, namely VGG16 [1], to study its performance in discriminating between live and fake fingerprint samples. We will use

the pre-trained model of VGG16 [1] that has been trained on ImageNet [2] dataset. This model will be trained on the fingerprint data from LivDet-2011 [14] and then its performance will be noted on unseen data from the same dataset.

The second objective is to train a light-weight (shallow) CNN model on LivDet-2011 [14] dataset and compare its effectiveness with respect to the state of the art VGG16 [1] model.

1.5 Organization of the Report

Chapter 2: We discuss the related works done by researchers in the area of Fingerprint Presentation Attack Detection using deep learning techniques.

Chapter 3: We study the performance of the standard VGG16 [1] architecture-based models on LivDet-2011 [14] dataset.

Chapter 4: We assess the performance of a shallow CNN architecture-based model custom designed just for the dataset. The model configuration is based on multiple experiments with the number and type of layers to be used.

Chapter 5: Finally, we summarise and conclude the findings of the two approaches.

Chapter 2
Related Work

In this chapter, we discuss some of the deep learning approaches and techniques that have been proposed by researchers for Fingerprint Presentation Attack Detection.

2.1 Technique using Adversarial Representation Learning Coupled with Style Transfer for Cross-Sensor Generalization

Grosz et al. [3] proposed the use of style transfer technique to augment the input data and then use Adversarial Representation Learning (ARL) [13] to train a classifier and help it learn more robust and invariant feature representations.

A style transfer based wrapper, called Universal Material Generator (UMG) [4] that was proposed by Chugh et al. [4], has been used here. This is an input data augmentation technique. In this approach a MobileNet-v1 [5] CNN model learns textural information of images from target domain images and then transfers the learnt style to the images in the source domain by synthetically generating large number of these representations. The features learnt from these images would be sensor invariant.

An ARL [13] model is then trained on these synthetic images. The ARL [13] architecture involves 3 parts – an encoder network, a prediction network and an adversarial network. The encoder is tasked to learn sensor invariant discernable feature representations from the training data. The adversarial network aims finding weakness in the learned representations challenging the primary model and in the process making encoder learn more sensor invariant features.

The researchers were able to get an improvement of 0.2% in True Detection Rate (TDR) and Fake Detection Rate (FDR) over the UMG [4] only model by testing on the LivDet-2015 [15] dataset.

2.2 Convolution Auto-encoders on Short Wave Infrared (SWIR) Images of Fingerprints

Kolberg et al. [6] proposed a single-class Presentation Attack Detection (PAD) method based on anomaly detection technique in using auto-encoders. An auto-encoder is nothing but a neural network that is optimized to reconstitute the original input data. An auto-encoder consists of two parts, namely – an encoder part, that creates a representation of an input, and the decoder, that tries to reconstruct the original input from this representation. This auto-encoder is designed in such a way that the output has lower dimension than that of the input. This would require the auto-encoder to extract the most relevant features from the input.

After the model is trained it can be used to encode any input and get a reconstructed output. The model will perform poorly to reconstruct a piece of data that is significantly unfamiliar or dissimilar from the training dataset. The reconstruction error (RE) will be huge. This is the premise of this FPAD approach proposed by Kolberg et al. [6]. The auto-encoder will be trained only on genuine fingerprint samples. A fake fingerprint, when passed through this trained auto-encoder, will generate high reconstruction error. Out of the 3 convolution auto-encoder architectures proposed (Conv-AE, Pooling-AE, Dense-AE) [6] it was found that Dense-AE performs better than the other two. Fingerprint dataset used in this training was collected by the research team manually.

2.3 Feature Denoising through Suppression of Noise Channels

Feng Liu et al. [7] used a novel approach to reduce the impact of noisy features from the images by suppressing certain channels from the feature map which contribute most to the noise data. This approach aims to suppress noisy features so that deep learning networks can focus on the visually most meaningful regions of an image and learn the feature map. This technique of focussing on the meaningful regions of an image is similar to the Attention Mechanisms [8-12] used in computer vision where attention modules are directed to focus on specific areas of interest.

This approach consists of primarily 3 steps. The first is evaluation of the importance of each channel. This is done by first generating a feature map consisting of all channels. A distance measure is then calculated of each channel with a high distance value indicating a noisy channel where as a low distance value indicating an important channel.

The second step is suppression of noisy channels to mitigate the propagation of noise in the learning. This is done by setting the noisy channels to zero. This step performs the “denoising” action.

The third step is domain-wise channel alignment through designing a PA-Adaptation Loss [7]. Since the features of live and fake fingerprints of various PAs vary randomly it was found difficult to define a single decision boundary for fake and live fingerprints. To tackle this the authors proposed a technique called PA-Adaptation loss [7] that can redefine the feature distribution for live and fake fingerprints such that fake fingerprints of different attack types are considered as separate classes from each other. The result of using this technique is that live fingerprints are clustered together and away from the fake fingerprints. Another outcome is that the fake fingerprints of same attack types are clustered together. The cross sensor and cross material performance of a model using this technique was found

to be impressive on the LivDet-2017 dataset. One drawback of this method is that since it is a multi-model based method its training exercise is very complex and time intensive and therefore it is not suitable for efficient deployment in practical scenarios.

Chapter 3
Exploring VGG16 Model

3.1 VGG16 Architecture

This CNN architecture consists of 16 convolution layers, 3 fully connected layers and 5 max pooling layers. This network learns the relevant features from the input layer and performs classification through its last layer which is a Softmax layer. We have taken the approach of transfer learning where we have utilized the pre-trained model (trained on ImageNet [2] dataset). We have modified the model by removing the top layer and also added a Dense layer since we plan to train the model further on the LivDet-2011 [14] dataset to perform binary classification.

We have trained the model on the Sagem sensor data from the LivDet-2011 [14] dataset.

3.2 Training on LivDet-2011 Dataset

The LivDet-2011 [14] (Sagem sensor) dataset has been augmented using horizontal and vertical flips randomly. Further augmentation has been added in form of zoom and feature normalization. All the pre-processing augmentations, shown in Table 3.1, have been done to both the training as well as testing data.

The model has been trained till 40, 50 and 60 epochs progressively and at each of these milestones the testing accuracy for cross sensor data has been recorded.

Table 3.1: Hyper-parameters for VGG16 model

Hyper-parameters	**Value**
image_size	224 × 224 pixels
batch_size	16
rescale	1/255
zoom_range	0.2
horizontal_flip	True
vertical_flip	True
samplewise_center	True
featurewise_std_normalization	True

Figure 3.1 shows the training / validation accuracy and loss curve. We notice that the validation loss is very volatile. The validation accuracy worsens after 50 epochs.

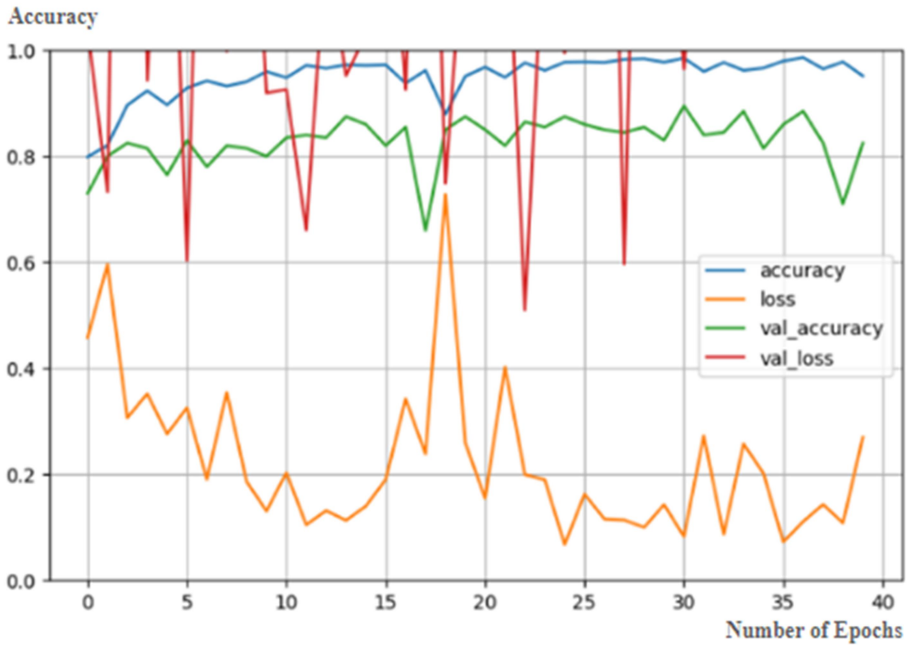

Figure 3.1: Training / validation accuracy and loss curve for VGG16 on LivDet-2011 dataset (Sagem sensor)

3.3 Test Results for VGG16 Model

The testing scores for both inter-sensor and intra-sensor data suggest that the model accuracy peaks at around 50 epochs of training with the current hyper-parameter set.

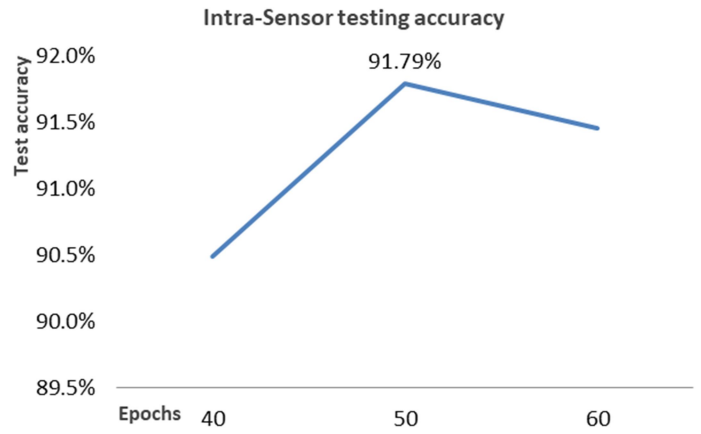

Figure 3.2: Intra-sensor testing accuracy of VGG16 model on LivDet-2011 dataset (Sagem sensor)

Figure 3.2 shows that the testing accuracy of unseen data within the same sensor dataset (Sagem) peaks at 91.79% at 50 epochs. It declines slightly after that in the next 10 epochs.

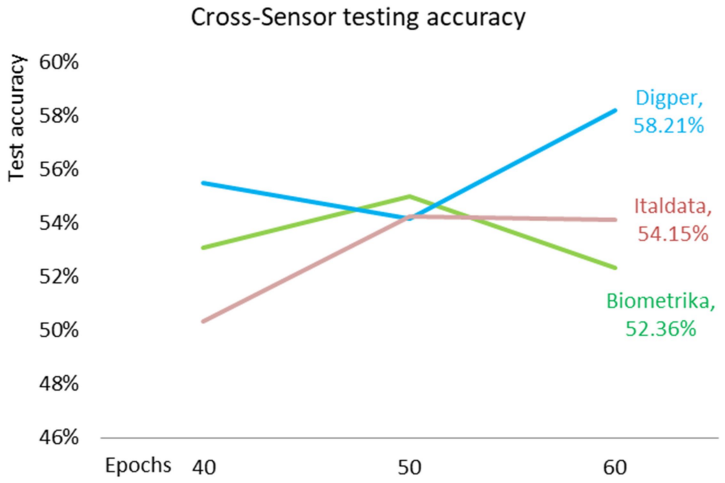

Figure 3.3: Cross-sensor testing accuracy of VGG16 model trained on LivDet-2011 Dataset (Sagem sensor)

Figure 3.3 shows the testing accuracy of unseen data from 3 other sensors – Digper, Italdata and Biometrika. From Table 3.2, we have also noticed that the testing accuracy varies differently for each sensor as the numbers of training epochs are increased. Testing accuracy on Digper sensor data set responds favourably to increase in number of training epochs particularly.

Table 3.2: Testing accuracy for VGG16 model trained on LivDet-2011 Dataset (Sagem sensor)

Sensor	**Epochs**	**Testing Accuracy**	**Testing Loss**
Sagem	40	90.49%	0.2642
	50	91.79%	0.2944
	60	91.45 %	0.2629
Italdata	40	50.34%	2.2060
	50	54.27%	2.3960
	60	54.15%	2.6188
Digper	40	55.52%	2.2849
	50	54.17%	2.6181
	60	58.21%	2.0889
Biometrika	40	53.08%	2.0990
	50	55.01%	2.2844
	60	52.36%	2.6758

Chapter 4
Exploring Shallow CNN Models

We have created two shallow CNN architectures and trained one model on LivDet-2011 Dataset from Sagem sensor. We have trained the other model on LivDet-2011 Dataset from Digper sensor. Input image were resized before processing to size 300×300 pixels. The intra-sensor testing accuracy on unseen data for the former model capped out at 88.95% whereas the same for the latter model reached 94.72%. However, the best cross sensor testing accuracy for the former model was higher at 55.29% while the latter was at 54.01%. The next few sections cover the details of these models and their result scores.

4.1 Training of Shallow CNN on LivDet-2011 Dataset (Sagem Sensor)

We have used a shallow CNN architecture consisting of 8 layers in total. There are 2 convolution layers, 2 max-pooling layers, 2 dense layers, 1 flatten layer and a last Softmax layer for this model. Table 4.1 shows the model summary for this shallow CNN architecture. No data augmentation was used. The optimizer used was Adam with default learning rate. Figure 4.1 shows that the model started reaching 100% training accuracy at around 8 epochs and validation accuracy peaked at about 80%, which means validation loss stabilized at a minimum of 20%.

Table 4.1: Shallow CNN architecture for LivDet-2011 (Sagem sensor)

Layer (type)	Output Shape	Param #
conv2d_18 (Conv2D)	(None, 298, 298, 64)	1,792
max_pooling2d_18 (MaxPooling2D)	(None, 99, 99, 64)	0
conv2d_19 (Conv2D)	(None, 97, 97, 24)	13,848
max_pooling2d_19 (MaxPooling2D)	(None, 32, 32, 24)	0
flatten_12 (Flatten)	(None, 24576)	0
dense_39 (Dense)	(None, 36)	884,772
dense_40 (Dense)	(None, 18)	666
dense_41 (Dense)	(None, 2)	38
Total params: 901,116 (3.44 MB) Trainable params: 901,116 (3.44 MB) Non-trainable params: 0 (0.00 MB)		

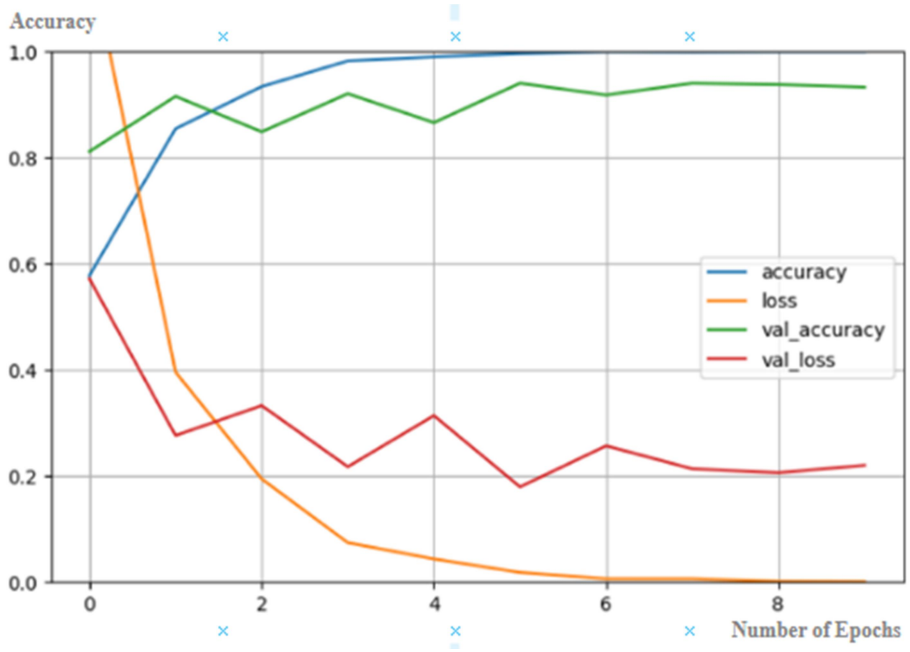

Figure 4.1: Training and validation accuracy and loss for shallow CNN (Sagem sensor) using ADAM optimizer

This model performed with an intra-sensor testing accuracy of 84% on unseen data. Increasing the number of training epochs made the validation-loss increase thereby ultimately decreasing the testing accuracy.

At this point we utilized the SGD optimizer (Stochastic Gradient Descent) with default parameters. The decline of validation loss was at a very gradual pace and therefore we had to train the model up to 65 epochs. Figure 4.2 shows the training and validation accuracies for this training phase. By the end of 65 epochs the validation loss plateaued at 11%. This improvement was significantly represented in the intra-sensor testing accuracy which increased from the earlier 84% to 88.95%.

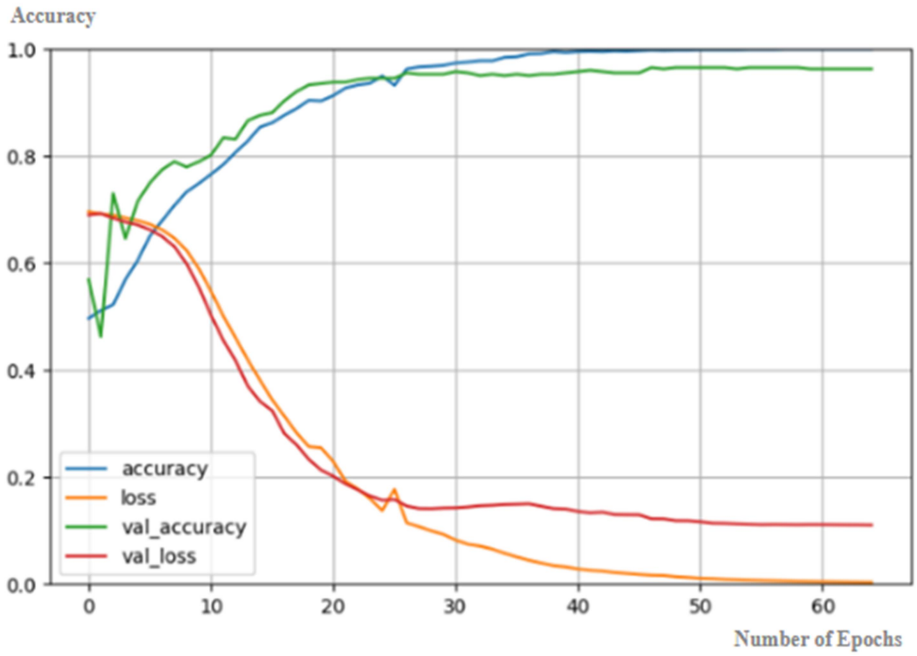

Figure 4.2: Training and validation accuracy and loss for shallow CNN (Sagem sensor) using SGD optimizer

4.2 Test Results for Shallow CNN Model (Sagem Sensor)

Table 4.2 shows the testing results of the model on LivDet-2011 unseen data. The model performs with 88.95% intra-sensor testing accuracy. Its cross sensor testing with LivDet-2011 (Digper sensor)

data yields a testing accuracy of 55.29%. This is the highest cross-sensor testing accuracy of this model. Table 4.3 shows the precision and recall values for this model against both for both fake and live classes. Figure 4.3 shows the precision and recall plots for each sensor data.

Table 4.2: Testing accuracy for shallow CNN model (Sagem sensor)

Sensor	**Testing Accuracy**	**Testing Loss**
Sagem	88.95%	0.3857
Digper	55.29%	4.2902
Biometrika	50.05%	8.2198
Italdata	46.42%	6.3409

Table 4.3: Model Classification report for shallow CNN model (Sagem sensor)

Sensor	**Class**	**Precision**	**Recall**
Digper	fake	0.78	0.11
	live	0.52	0.97
Italdata	fake	0.06	0.01
	live	0.48	0.92
Biometrika	fake	0	0
	live	0.5	1
Sagem	fake	0.88	0.89
	live	0.89	0.87

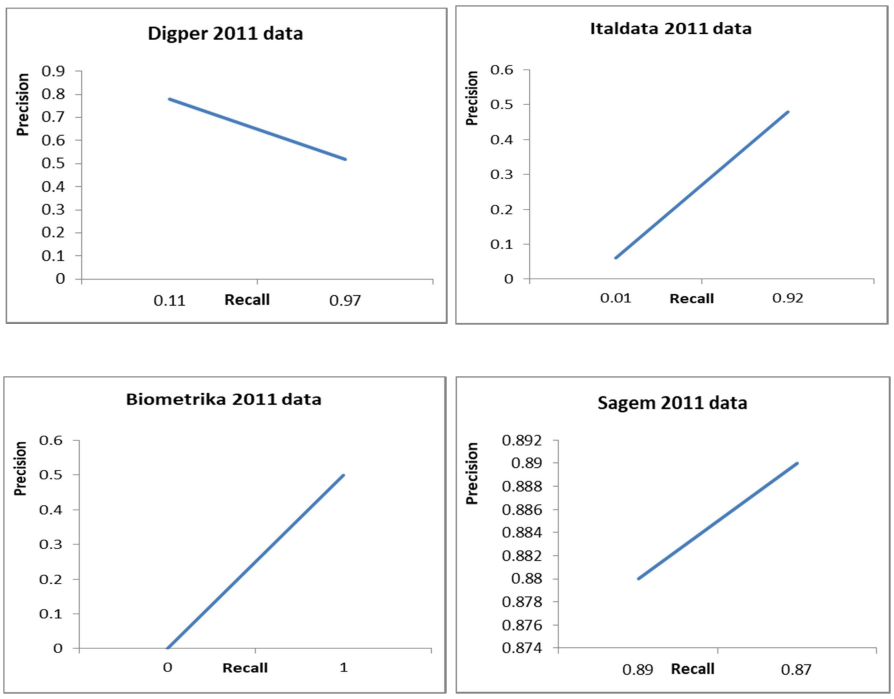

Figure 4.3: Precision vs Recall for shallow CNN model (Sagem sensor)

4.3 Training Shallow CNN on LivDet-2011 Dataset (Digper Sensor)

We have used a shallow CNN architecture consisting of 9 layers in total. There are 2 convolution layers, 2 max-pooling layers, 3 dense layers, 1 flatten layer and a last Softmax layer for this model. Table 4.4 shows the model summary.

No data augmentation was used. The optimizer used was Adam with default learning rate. The optimal number of epochs for best training and validation accuracy was experimentally determined to be 17 epochs. Training accuracy and validation accuracy capped out at 97.68% and 98.00% respectively at the end of 17 epochs.

Table 4.4: Model summary of the shallow CNN on LivDet-2011 (Digper Sensor)

Layer (type)	**Output Shape**	**Param #**
conv2d_4 (Conv2D)	(None, 298, 298, 64)	1,792
max_pooling2d_4 (MaxPooling2D)	(None, 149, 149, 64)	0
conv2d_5 (Conv2D)	(None, 147, 147, 32)	18,464
max_pooling2d_5 (MaxPooling2D)	(None, 73, 73, 32)	0
flatten_2 (Flatten)	(None, 170528)	0
dense_8 (Dense)	(None, 128)	21,827,712
dense_9 (Dense)	(None, 48)	6,192
dense_10 (Dense)	(None, 12)	588
dense_11 (Dense)	(None, 2)	26
Total params: 21,854,774 (83.37 MB) Trainable params: 21,854,774 (83.37 MB) Non-trainable params: 0 (0.00 MB)		

Figure 4.4 shows the training and validation progress curve for the training phase. We notice that the validation accuracy and validation loss have fluctuated a lot during the training. However, at the end of 17 epochs the training accuracy and validation accuracy converge near 98% mark. Similarly the training loss and validation loss both show convergence around the 7% mark.

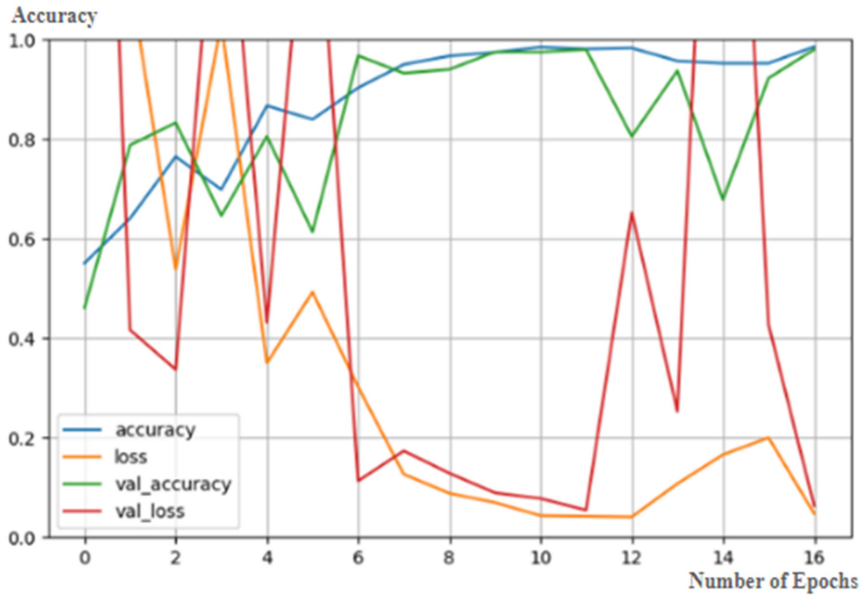

Figure 4.4: Training and validation accuracy and loss for shallow CNN (Digper sensor)

4.4 Test Results for Shallow CNN Model (Digper Sensor)

Table 4.5 shows that the classification report of the training phase of this model. Note the high precision and recall values of this model. Table 4.6 shows the highest intra-sensor testing accuracy achieved is 94.72% and the highest cross sensor testing accuracy achieved is 54.01%. Table 4.7 shows the classification report of the testing phase. Figure 4.5 shows the precision and recall plots for each sensor.

Table 4.5: Model classification report in the training phase

Class	Precision	Recall	F1-score	Support
fake	0.96	0.94	0.95	1000
live	0.94	0.96	0.95	1000
Accuracy			0.95	2000
macro avg	0.95	0.95	0.95	2000
weighted avg	0.95	0.95	0.95	2000

Table 4.6: Testing accuracy for shallow CNN model (Digper sensor)

Sensor	Testing Accuracy	Testing Loss
Digper	94.72%	0.2485
Biometrika	54.01%	8.1761
Italdata	51.50%	10.938
Sagem	51.90%	13.8565

Table 4.7: Model classification report in the testing phase

Sensor	Class	Precision	Recall
Digper	fake	0.96	0.94
	live	0.94	0.96
Italdata	fake	0.5	0.99
	live	0.36	0.01
Biometrika	fake	0.51	0.99
	live	0.82	0.06
Sagem	fake	0.51	1
	live	0	0

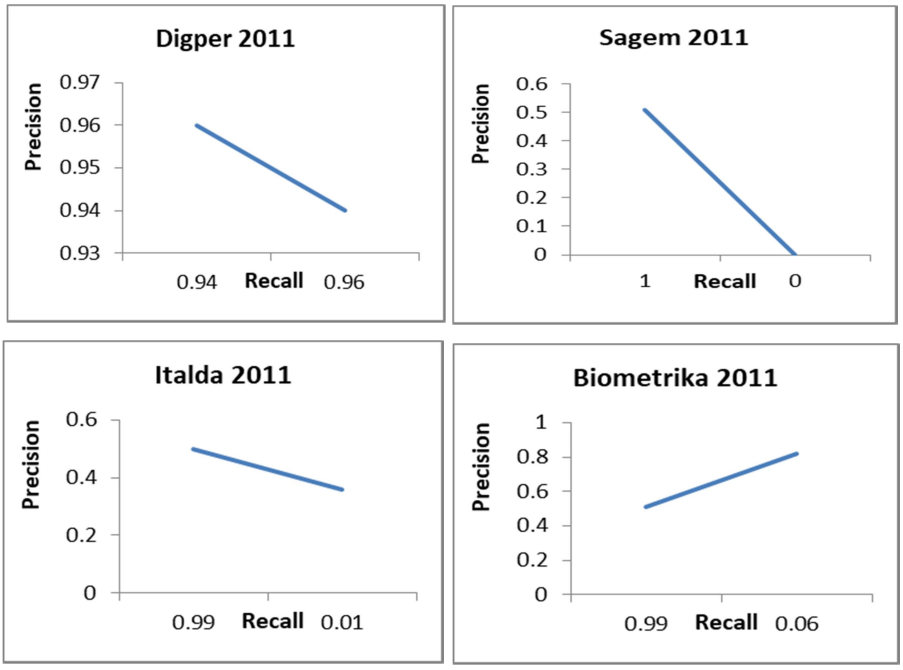

Figure 4.5: Precision vs Recall for shallow CNN model (Digper sensor)

Chapter 5

Conclusion and Future Work

5.1 Conclusions

This study of VGG16 [1] model's performance using pre-trained model weights, of a different dataset, and using no handcrafted features at best gives an intra-sensor testing accuracy of 91.79% and a cross sensor testing accuracy of 58%.

In comparison, a shallow CNN network (Digper sensor) exceeds this intra-sensor testing accuracy by achieving a 94.72% accuracy result. The cross-sensor testing accuracy of the shallow CNN network (Sagem sensor) is only slightly less than VGG16 [1] at 55.29%.

Also, the shallow networks needed 70%-80% less time and number of epochs to train to arrive at this comparable performance. Therefore, it might be worthwhile to invest more research into designing and fine-tuning shallow CNN architectures for applications in Fingerprint Presentation Attack Detection.

We notice in Chapter 4 that changing the optimizer from Adam to SGD increased the validation accuracy from 84% to almost 89%. SGD poses problems of vanishing gradient in deep neural networks. However, with a shallow network this effect will be minimal. This gives us more leg-room to fine-tune and train our shallow models better.

5.2 Future Work

As discussed in Section 5.1, more research into shallow CNN architectures for FPAD is warranted. In a future state, the use of a number of efficient small (shallow) CNN models clubbed as part Decision Tree / Random Forest architecture to work as an ensemble classifier is something that could be researched into. Shallow models

can find applicability in devices with low computing power such as mobile devices. Such devices may not be able to run large and complex models which require a lot of computing power. Therefore, shallow models may find utility here.

References

[1] S. Liu and W. Deng. Very deep convolutional neural network based image classification using small training sample size. In Proceedings of the 3rd IAPR Asian Conference on Pattern Recognition (ACPR), pp. 730–734, 2015 Kuala Lumpur, Malaysia.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255, 2009, Miami, Florida, USA.

[3] S. A. Grosz, T. Chugh and A. K. Jain. Fingerprint Presentation Attack Detection: A Sensor and Material Agnostic Approach. In Proceedings of the IEEE International Joint Conference on Biometrics (IJCB), 2020, Houston, Texas, USA.

[4] T. Chugh and A. K. Jain. Fingerprint spoof generalization. IEEE Transactions on Information Forensics and Security, Volume: 16, pp. 42 – 55, 2020

[5] T. Chugh, K. Cao, and A. K. Jain. Fingerprint spoof buster: Use of minutiae-centered patches. IEEE Transactions on Information Forensics and Security, Volume: 13, Issue: 9, pp. 2190–2202, 2018.

[6] J. Kolberg, M. Grimmer, M. Gomez-Barrero and C. Busch. Anomaly detection with convolutional autoencoders for Fingerprint Presentation Attack Detection. In IEEE Transactions on Biometrics, Behavior, and Identity Science, 2021.

[7] F. Liu, Z. Kong, H. Liu, W. Zhang and L. Shen. Fingerprint Presentation Attack Detection by Channel-wise Feature Denoising. IEEE Transactions on Information Forensics and Security, Volume: 17, pp. 2963 – 2976, 2022.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is All you Need. In Proceedings of the Advances in Neural Information Processing Systems, pp.5998–6008, 2017, Long Beach, California, USA.

[9] J. Hu, L. Shen, and G. Sun. Squeeze-and-Excitation Networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7132–7141, 2018, Salt Lake City, Utah, USA.

[10] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu. Dual attention network for scene segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3146–3154, 2019, Long Beach, California, USA.

[11] S. Chaudhari, V. Mithal, G. Polatkan, and R. Ramanath. An attentive survey of attention models. ACM Transactions on Intelligent Systems and Technology, Volume 12, Issue 5, pp. 1–32, 2021.

[12] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: neural image caption generation with visual attention. In Proceedings of the 32nd International Conference on Machine Learning, PMLR, pp. 2048–2057, 2015. Lille, France.

[13] Q. Xie, Z. Dai, Y. Du, E. Hovy, and G. Neubig. Controllable invariance through adversarial feature learning. In Proceedings of the Advances in Neural Information Processing Systems 30, pages 585–596, 2017, Long Beach, California, USA.

[14] D. Yambay, L. Ghiani, P. Denti, G. Marcialis, F. Roli, S. Schuckers. LivDet 2011 - Fingerprint liveness detection competition 2011. In Proceedings of 5th IAPR International Conference on Biometrics (ICB), 2011, New Delhi (India).

[15] V. Mura, L. Ghiani, G. Marcialis F. Roli, D. Yambay, S. Schuckers. LivDet 2015 - Fingerprint liveness detection competition 2015. In the Proceedings of IEEE 7th International Conference on Biometrics Theory, Applications and Systems (BTAS), 2015, Arlington, Virginia, USA.

[16] M. Ali, V. Mahale, P. Yannawar, A. Gaikwad. Overview of fingerprint recognition system. In the Proceedings of International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), 2016, Chennai, India.

Finalversion

ORIGINALITY REPORT

9%	6%	6%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Indian Institute of Technology Indore Student Paper	1%
2	export.arxiv.org Internet Source	1%
3	Tyler, Jeramey. "Spatial Location of Binaural Signals Using Cepstral Analysis", Rensselaer Polytechnic Institute, 2024 Publication	1%
4	Submitted to Harrisburg University of Science and Technology Student Paper	1%
5	Submitted to University of Surrey Student Paper	1%
6	Submitted to George Bush High School Student Paper	<1%
7	Md. Mamun Hossain, Md. Moazzem Hossain, Most. Binoee Arefin, Fahima Akhtar, John Blake. "Combining State-of-the-Art Pre-Trained Deep Learning Models: A Noble	<1%