doi.org/10.24191/jeesr.v19i1.013

Fingerprint Presentation Attack Detection Using Deep Transfer Learning and DenseNet201 Network

Divine S. Ametefe, Suzi S. Seroja, and Darmawaty M. Ali

Abstract— **Fingerprint presentation attack, which involves presenting spoof fingerprints to fingerprint biometric sensors to gain illicit access, is a significant challenge faced by Automatic Fingerprint Identification Systems (AFIS). As a result, various hardware-based and software-based approaches have been posited to help remedy this concern. However, the software-based methods have seen enormous utilisation relative to the hardware-based techniques due to their robust cognitive feature extraction for spoof detection. Nonetheless, most software-based techniques that utilise handcrafted features proffer shallow features for discriminating against spoofs due to their manual feature extraction process, which, as a result, affects the model's robustness. Motivated by this concern, we propose a deep transfer learning approach to automatically learn deep hierarchical semantic fingerprint features as a means of discriminating against spoofs. Experiments were conducted on the LivDet competition standard database, encompassing datasets from LivDet-2009, 2011, 2013, and 2015, resulting in the acquisition of real fingerprints and fake fingerprints fabricated from twelve (12) different spoofing materials. The developed model recorded an average classification accuracy of 99.8%, a sensitivity of 99.73% and a specificity of 99.77%, showcasing a state-of-the-art performance.**

Index Terms—**Presentation Attack; Spoof Detection; Deep Transfer Learning; DenseNet201.**

I. INTRODUCTION

IN recent times, with an unprecedented rise in technology, biometrics has played a vital role in the use of IoT applications, protecting user confidentiality in diverse applications. According to [1]–[5], the fingerprint biometric modality is the most widely used biometric modality for access control functions. Due to its vast international utilisation, it is greatly hampered by various security threats such as identity theft, account hacking, unauthorised access, but to mention a few. However, amongst all forms of threats or attacks faced by Automatic Fingerprint Recognition Systems (AFIS), the most predominant is the fingerprint presentation attack (i.e., fingerprint Spoofing). This form of attack involves the fabrication of fake fingerprints, using various ubiquitous materials like latex, eco-flex, clay, wood glue, and gum to circumvent fingerprint scanners to achieve unauthorised access to the system.

Nevertheless, in a quest to counter this problem, several presentation attack detection approaches have been posited. However, as fingerprint sensing technology advances, so does the spoofing technology, which proliferates organisations' difficulty to have secured AFIS that prevent systems' circumvention. Based on physical observations, it is nearly impossible to discern a clear difference between real and spoofed fingerprints on the sensory imagery (i.e., point 1), as shown in *Figure 1.* Hence, various software-based approaches are used to extract global-level, local-level, or fine-level fingerprint features as determinants for spoof detection.

Traditionally, handcrafted textural features extracted from the fingerprints are used to decipher between real and fake fingerprints [6]. Due to the nature of these kinds of features, high-resolution images and exhaustive feature tuning is necessary. As a result, they are easily influenced by computed features and noise. However, to counter this issue, Menotti et al. developed a CNN-based network called 'SPOOFNET' [7], training on LivDet 2013 dataset. This supervised network learns intrinsic fingerprint features, resulting in a significantly enhanced classification performance. Towing a similar paradigm, [8] and [9] deployed standard CNN networks (VGGNet, AlexNet), pre-trained models on a large ImageNet dataset and fine-tuned on LivDet dataset. Findings from [8],[9] showcased that using pre-trained models with transfer learning progressively enhances performance. Motivated by these works, many CNN-based fingerprint spoof detection methods have been proposed [7],[9]–[12]. However, despite the state-of-the-art performance manifested by these fingerprint spoof detection methods, fingerprint spoof detection models developed using handcrafted features through manual selection and extraction of fingerprint features, usually proffer shallow

This manuscript is submitted on 10th January 2021 and accepted on 19th April 2021. D.S. Ametefe, S.S. Sarnin and D.M. Ali are with the School of Electrical Engineering, College of Engineering, Universiti Teknologi MARA, 40450 Shah Alam, Selangor (e-mail: divinea10@gmail.com, suzis045@uitm.edu.my, darma504@uitm.edu.my)

features of samples for discrimination instances. Also, models developed using the CNN-based paradigm, involving the utilisation of pre-trained-models such as VGGNet, AlexNet etc., [7],[9]–[12] have some shortcomings with memory efficiency, computational efficiency, and feature reusability [14].

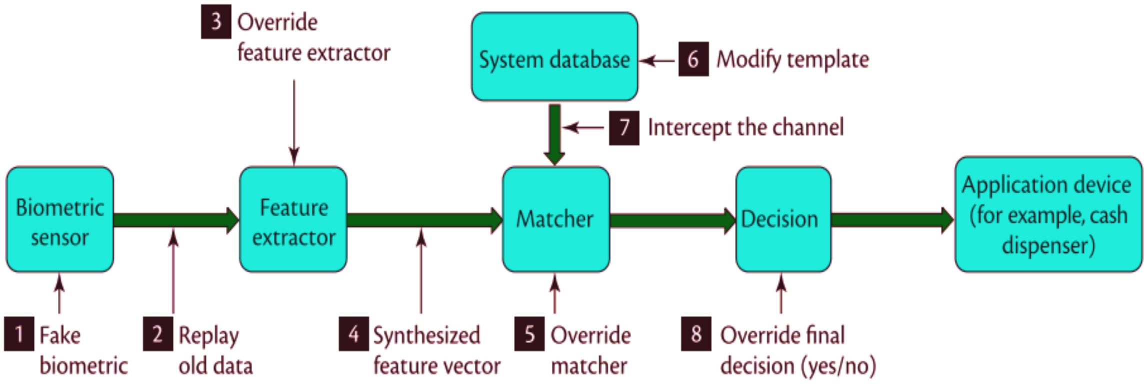

Figure 1: Points of attacks on a generic fingerprint biometric system

Motivated by these concerns, we propose a deep transfer learning approach to automatically learn deep hierarchical semantic fingerprint features as a means of discriminating against spoofs. We also employed DenseNet201 as a pre-trained model to enhance memory efficiency, computational efficiency, and feature reusability. Experiments were conducted on the standard LivDet Competition database (LivDet 2009, 2011, 2013, 2015), involving real fingerprints and fake fingerprints fabricated from twelve (12) distinct spoofing materials. This paper's remainder is structured as follows; Some related literature works are highlighted in section 2. The study's proposed method and evaluation of experimental results are proffered in sections 3 and 4, respectively. Section 5 concludes the study.

II. Related Works

This section briefly highlights existing literature relating to Presentation attack detection, fingerprint image classification schemes, and DenseNets.

Due to the enormous security threat posed by fingerprint spoofing practices, presentation attack detection has become a biometrics priority. Presentation attack detection involves attackers maliciously fabricating fingerprint impressions on ubiquitous materials like gum, gelatin, playdoh, etc., and presenting such spoofed fingerprints to circumvent AFIS [15]. There are two essential presentation attack detection techniques, encompassing hardware-based and software-based approaches. Hardware-based approaches involve using additional sensing devices to garner other inherent live characteristics like blood pressure, blood sugar, skin distortion, or odour etc., [15],[16]. On the other hand, software-based approaches involve extracting various handcrafted features from the sensor image and then classifying them as live or fake [17]–[19]. These handcrafted features can be broadly categorised as exterior anatomical features like ridge strength, pore location and their distribution, continuity, and clarity [20], etc., or physiological features like perspiration patterns [3] and textural based features or statistical features like Weber Local Descriptors (WLD) [21], rotation-invariant Local Phase Quantisation (LPQ) [22] features, or Binarized Statistical Image Features (BSIF) [23] or Local Binary Patterns [24], etc. or a combination of these features.

The implementation of textural features involving Local Binary Pattern (LBP) and Gabor filters was the approach adopted by [25] because LBP proffers good textural variation for liveliness detection [24]. In the 2015 LivDet competition, [8] obtained state-of-the-art accuracy while utilising LBP and transfer learning for binary classification. In this approach, LBP features were extracted from the fingerprint samples' pre-processing and then classified using pre-trained standard models VGG, AlexNet. A similar technique was harnessed by [9] and [7], obtaining significantly better results than spoof detection using only handcrafted features. However, these forms of approaches resulted in poor cross-sensory and cross material evaluations. Hence, to overcome this, [26] developed a robust spoof detection method exploring the local minutiae-based patches with MobileNet-v1 as the training network [27]. This technique proved efficient, producing state-of-the-art results for intra-sensor, inter-sensor, cross-material, and cross sensor over three datasets (LivDet 2011, 2013, 2015). However, despite resulting in a good performance, this method had some lapses, in that the minutiae point extraction necessitates the use of high-resolution input fingerprint images (> 500dpi). The two-stage training process involved makes the model not operate in an end-to-end fashion like [25].

DenseNet [14] as a network is extensively used for various applications like image classification [28], [29], segmentation [30], image super-resolution [31] etc. These DenseNet features can be attributed to their memory efficiency, computational efficiency, and feature reusability properties. This network alleviates the vanishing gradient problem and strengthens feature propagation. Its feature reusability ensures memory and computational efficiency. Hence, in the quest to counter facial presentation attacks, [32] used DenseNet as the based network. A study by [33] also harnessed DenseNet coupled with LSTM for audio spoof attack detection. However, DenseNet201 has not been used with deep transfer learning for fingerprint presentation attack detection to the best of our knowledge.

III. Proposed Methodology

We propose a deep transfer learning approach, utilising the DenseNet201 network to develop a real fingerprint classifier for countering presentation attacks. This technique facilitates the proposed model to automatically garner and learn inherent fingerprint features for discriminating against spoofs. This approach was implemented on the standard LivDet competition database, encompassing the LivDet 2009, 2011, 2013, and 2015 datasets. Real fingerprints and fake fingerprints made from twelve (12) different spoofing materials were utilised for training the model. All available fake fingerprints from the various spoofing materials were combined into a single fake fingerprints label. Hence, two essential labels (real and fake fingerprint samples) were realised and used to develop a fingerprint binary classifier, as shown in *Figure 2*.

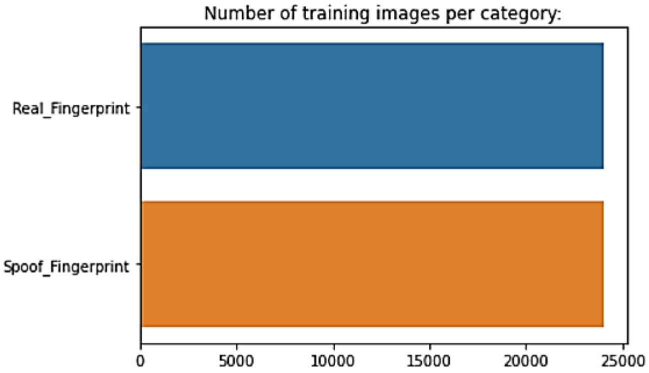

Figure 2: Real and fake fingerprint dataset classes

A. *Fingerprint Categorisation*

One of the critical processes essential in deep transfer learning is the categorisation of the available datasets. Three important categorisations of the available datasets were implemented, encompassing the training datasets, validation datasets, and testing datasets. The training dataset is used for the model's cognitive training to possess the dexterity of discerning between real and fake fingerprints. The validation dataset is used as a pre-test dataset during the training process to determine the trained model's efficacy across the various epochs. Finally, the test dataset is used to probe the practical robustness of the model. A set of holdout datasets not used for training, validation, and testing was also created to further examine the finalised model's practical effectiveness. A total of 60,000 fingerprint datasets were obtained and used for the development of the model. However, since there isn't a standardised dataset categorisation paradigm available, we adopted a dataset split ratio of 8:1:1 analogous to [34] across the training, validation, and testing samples, respectively, as highlighted in TABLE *1*.

TABLE I
PROPOSED CATEGORISATION OF FINGERPRINT DATASETS

Fingerprint Classes	No. Training Dataset	No. Validation Dataset	No. Testing Dataset	Total Dataset	Holdout Dataset
Real Fingerprnts	24,000	3,000	3,000	30,000	10
Fake (Spoof) Fingerprints	24,000	3,000	3,000	30,000	10

The fake fingerprint datasets consisted of twelve (12) fingerprint spoofing materials encompassing Body-Double, Ecoflex, Gelatin, Latex, Liquid_Ecoflex, Modasil, OOMOO, Playdoh, RTV, Silgum, Silicone, WoodGlue, as represented by *Figure 3*.

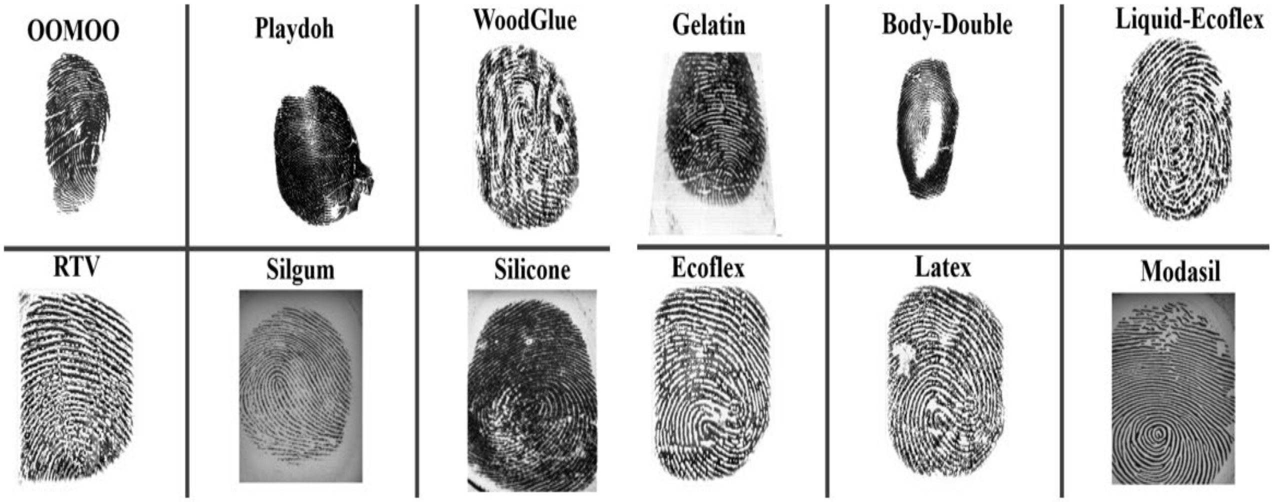

Figure 3: Fake fingerprints made from twelve spoof materials used for the study

B. Dataset Pre-Processing

The fingerprint images have random sizes due to the variation of fingerprint sensing devices used by LivDet Competition to capture fingerprint impressions. Hence, to ensure best practice and enhanced training experience, a suitable image scaling size was adopted. A scaling size of 224 x 224 was implemented by [34] for attaining state-of-the-art image classification. Hence, this study also harnessed a similar scaling size (224 x 224), as shown in *Figure 4*.

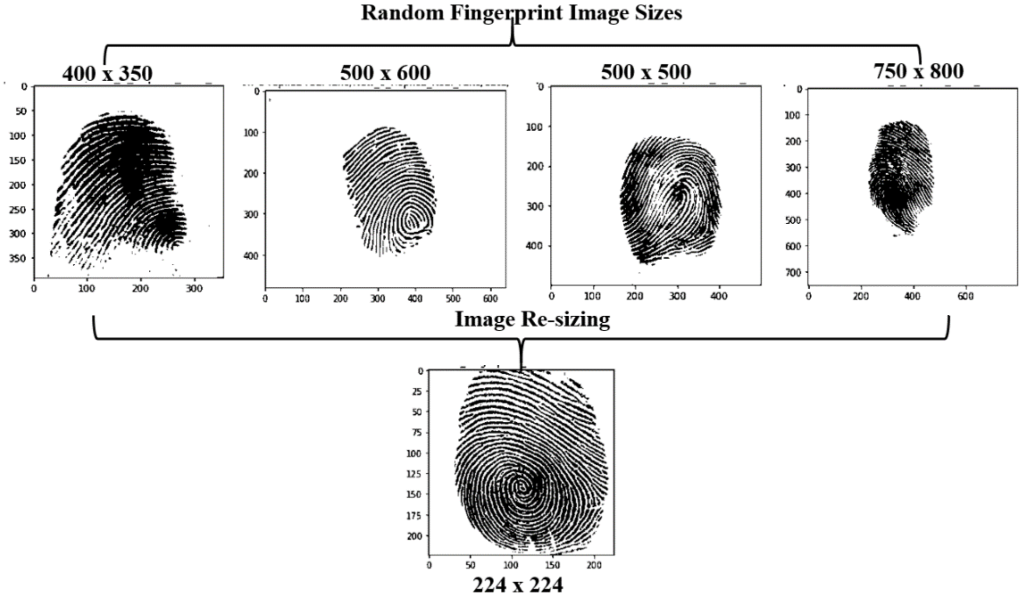

Figure 4: Rescaling of Fingerprint Images

The rescaling implementation resulted in low memory consumption, reduced computational time, and the fingerprint profile's complete usage without losing fingerprint details. Dataset normalisation was also introduced to scale the magnitude of fingerprint image attributes in the range of 0 and 1, which served as an image filter (feature extraction). The proposed fingerprint binary classification model was developed through an automatic exploration of two (2) salient features through the implementation of deep transfer learning, encompassing the global-level and local-level features as showcased by *Figure 5*.

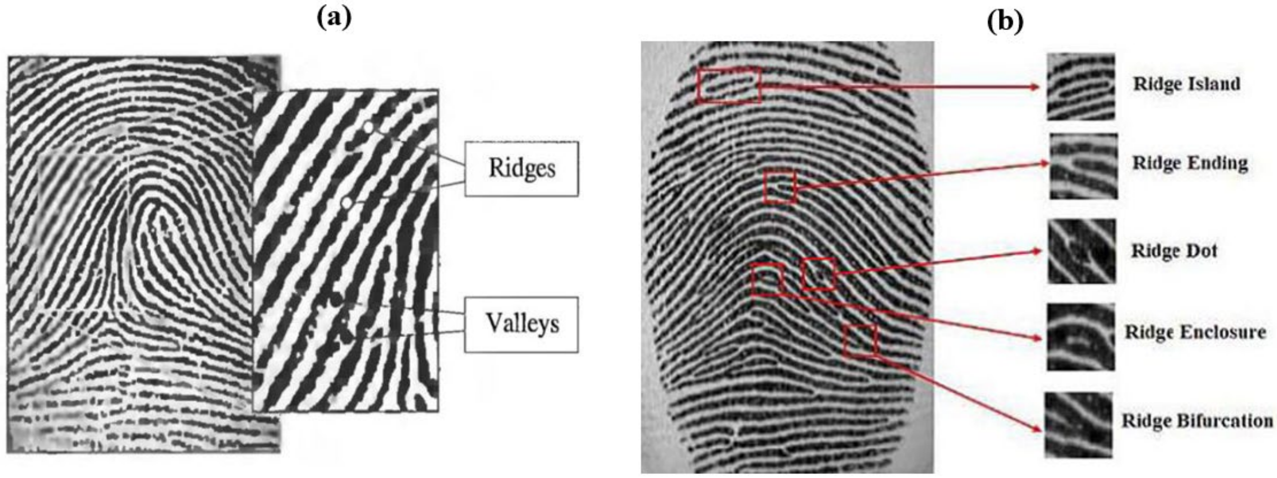

Figure 5: Global-level and Local-level fingerprint features

C. Deep Transfer Learning Process

The transfer learning framework is hinged on three critical parameters, encompassing the domains, tasks, and marginal probabilities [35]. This framework can be defined as:

A domain, $\mathbf{D}$, is defined as a two-element tuple consisting of feature space, $\boldsymbol{x}$**,** and marginal probability**,** $\boldsymbol{P}(\mathbf{X})$**,** where $\mathbf{X}$**,** is a sample data point. Thus, its mathematical representation is given as $\mathbf{D} = \{\boldsymbol{x}, \boldsymbol{P}(\mathbf{X})\}$**.**

A Domain consists of two components: $\mathbf{D} = \{\boldsymbol{x}, \boldsymbol{P}(\mathbf{X})\}$**.**

Feature space: $\boldsymbol{x}$
Marginal distribution: $\boldsymbol{P}(\mathbf{X}), \mathbf{X} = \{x_1, \ldots\ldots\ldots x_n\}, x_i \in \boldsymbol{x}$.
Here $\boldsymbol{x_i}$ represents a specific vector as described in the above depiction. A task, $\mathbf{T}$, on the other hand, can be defined as a two-element tuple of the label space, $\boldsymbol{y}$, and objective function, $\boldsymbol{\eta}$. The objective function can also be denoted as $\mathbf{P}\ (\boldsymbol{y}|\ \mathbf{X})$ from a probabilistic viewpoint.
For a given domain $\mathbf{D}$, a Task is defined by two components:

$$\mathbf{T} = \{\boldsymbol{y}, \boldsymbol{P}(\boldsymbol{Y}|\mathbf{X})\} = \{\boldsymbol{y}, \boldsymbol{\eta}\} \quad \boldsymbol{Y} = \{\boldsymbol{y_1}, \ldots\ldots\ldots \boldsymbol{y_n}\}, \boldsymbol{y_i}\ \in \boldsymbol{y}$$

A label space: $\boldsymbol{y}$
A predictive function $\boldsymbol{\eta}$ learned from feature vector/label pairs $(\boldsymbol{x_i}, \boldsymbol{y_i})$, $x_i \in x$, $\boldsymbol{y_i}\ \in \boldsymbol{y}$
for each feature vector in the domain, n predicts its corresponding label: $\boldsymbol{\eta}\ (\boldsymbol{x_i}) = \boldsymbol{y_i}$
Hence,
Given a source domain $\boldsymbol{D_S}$, a corresponding source task $\boldsymbol{T_S}$ as well as a target domain $\boldsymbol{D_T}$ and a target task $\boldsymbol{T_T}$, the objective of transfer learning now is to enable us to learn the target conditional probability distribution $\mathbf{P}(\boldsymbol{Y_T}|\ \boldsymbol{X_T})$ in $\boldsymbol{D_T}$ with the information gained from $\boldsymbol{D_S}$ and $\boldsymbol{T_S}$ where $\boldsymbol{D_S}\ \neq \boldsymbol{D_T}$ OR $\boldsymbol{T_S} \neq \boldsymbol{T_T}$.

The implementation of deep transfer learning necessitates the use of pre-trained models. Hence, the DenseNet (DenseNet201) was selected as the pre-trained model due to its characteristics. DenseNet is much esteemed in the paradigm of image classification, image segmentation, and image super-resolution [36], resulting in optimal performances. Its implementation has been attributed to memory efficiency, computational efficiency, and feature reusability capabilities [25]. Its utilisation also supports the mitigation of vanishing gradients' concerns and strengthens feature propagation [36]. Furthermore, its essential property of feature reusability ensures memory and computational efficiency.

For an adequate implementation of a pre-trained model, the selected execution method plays a significant role. Under the scope of deep transfer learning, four (4) principal execution approaches [37] are shown in *Figure 6*. Before selecting any of the principles, two (2) parameters are essential and help choose the datasets' size and similarity to the pre-trained model datasets. However, a dataset is tagged small and large when the available datasets are less and more than 1,000, respectively [34]. Also, a dataset is considered similar if the available dataset is comparable to the pre-trained datasets. For instance, a pre-trained model used to classify blood cells is identical when used to classify cancerous cells. On the other hand, a dataset is dissimilar if the available dataset is not like the pre-trained datasets. For instance, a pre-trained model used for classifying flowers is distinct when used to classify automobiles.

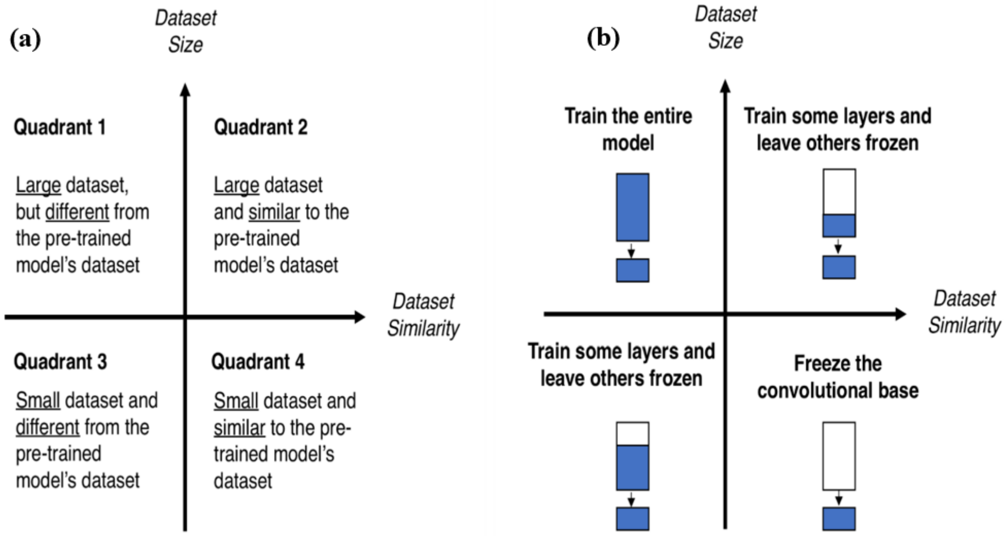
(a)
Dataset Size
Quadrant 1
Large dataset, but different from the pre-trained model's dataset
Quadrant 2
Large dataset and similar to the pre-trained model's dataset
Dataset Similarity
Quadrant 3
Small dataset and different from the pre-trained model's dataset
Quadrant 4
Small dataset and similar to the pre-trained model's dataset
(b)
Dataset Size
Train the entire model
Train some layers and leave others frozen
Dataset Similarity
Train some layers and leave others frozen
Freeze the convolutional base

Figure 6: (a) Size-Similarity Matrix and (b) Decision Map For Fine-Tuning Pre-Trained Models

Since the study has over 1,000 available datasets that are not similar to that of the pre-trained model, the first quadrant was implemented and involved training the entire model (i.e., convolutional base and classifier) as depicted by *Figure 6(b).*

D. Implementation of Deep Transfer Learning Using DenseNet201

The execution of pre-trained models is generally hinged on the principles of Convolution Neural Networks (CNN). Its implementation architecture is stratified into two essential divisions [37], encompassing the convolutional base and classifier, as showcased by *Figure 7*. The convolutional base is stacked with image filters that aid in feature extraction, while the classifier is used for making predictions based on the kind of features extracted.

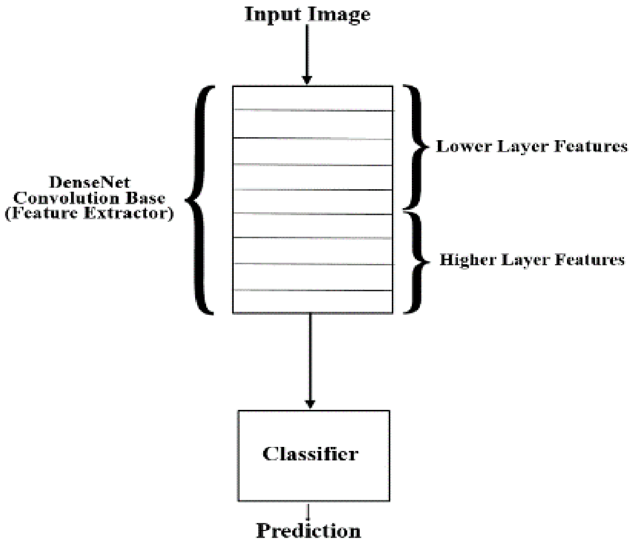

Figure 7: The Architecture Of Proposed Model Based Convolutional Neural Network

As indicated by *Figure 7*, the convolutional based comprises two (2) primary partitions, involving the lower and higher layers. In deep transfer learning, the learning process is automatically realised hierarchically, starting and ending from lower to higher layers. The lower layer computes general features and could be reused for different problem domains. In contrast, the higher layer computes specialised features, dependent on the type of datasets. The specialised features we used for developing our model is the local fingerprint features (textural features). However, despite having lower layers and higher layers, a group of transitional layers (hidden layers) is sandwiched between them, as shown in *Figure 8*.

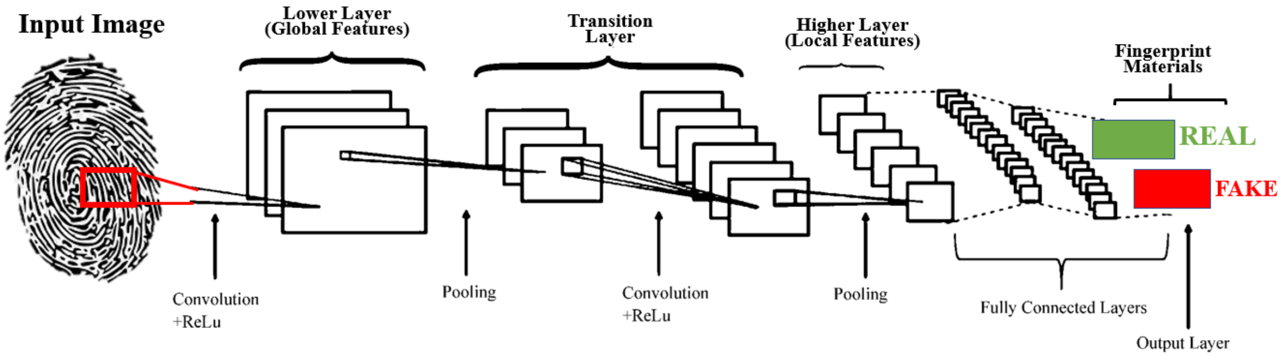

Figure 8: Deep Transfer Learning Process

The purpose of these convolutional layers is to derive a feature map (i.e., a trainable classifier), as shown in *Figure 9*. Traditionally, a low number of filters are used at low-level for feature detection. However, as the progressive learning process goes deeper into the CNN, additional filters are engaged to detect and extract high-level features. Feature maps are birth by scanning the input dataset with filters of varying sizes coupled with matrix computations.

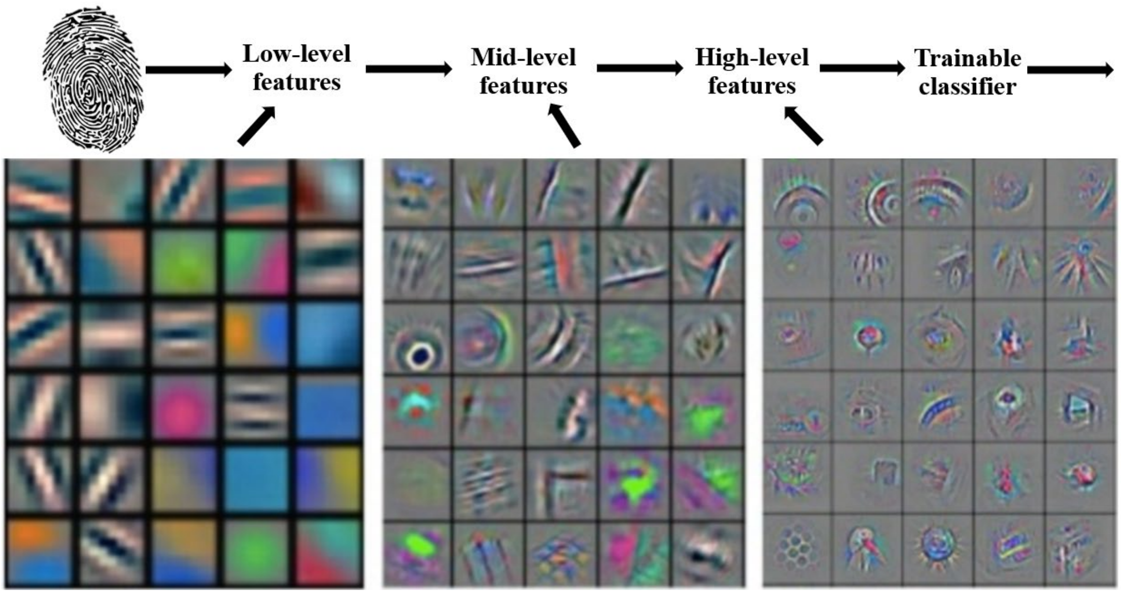

Figure 9: Feature Extraction in Deep Transfer Learning Process

The activation function adopted for the fingerprint classification model was the Rectified Linear Unit (ReLu) function, as shown in *Figure 10*. ReLu activation function was selected because of its efficiency when implemented for binary classification. The evaluation of the ReLu activation function has a max(0, z).

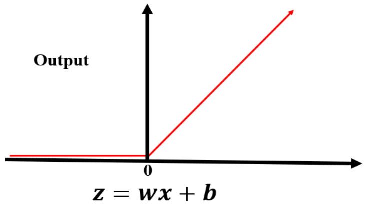

Figure 10: Rectified Linear Unit (ReLu) function

In summary, *Figure 11* shows the deep transfer learning process executed for the study. The initial step involved obtaining datasets, which were then cleaned (pre-processed) to suit the network's requirements. The pre-processed datasets were after divided into the train (validation inclusive) and test split. The model's training was initiated using the available training datasets with some fine-tuning of some optimisation parameters (hyper-parameters). After a satisfying training experience, the model was tested on a set of test datasets. Finally, the holdout dataset was used to evaluate the model's functional performance. It should be noted that training and validation were implemented simultaneously.

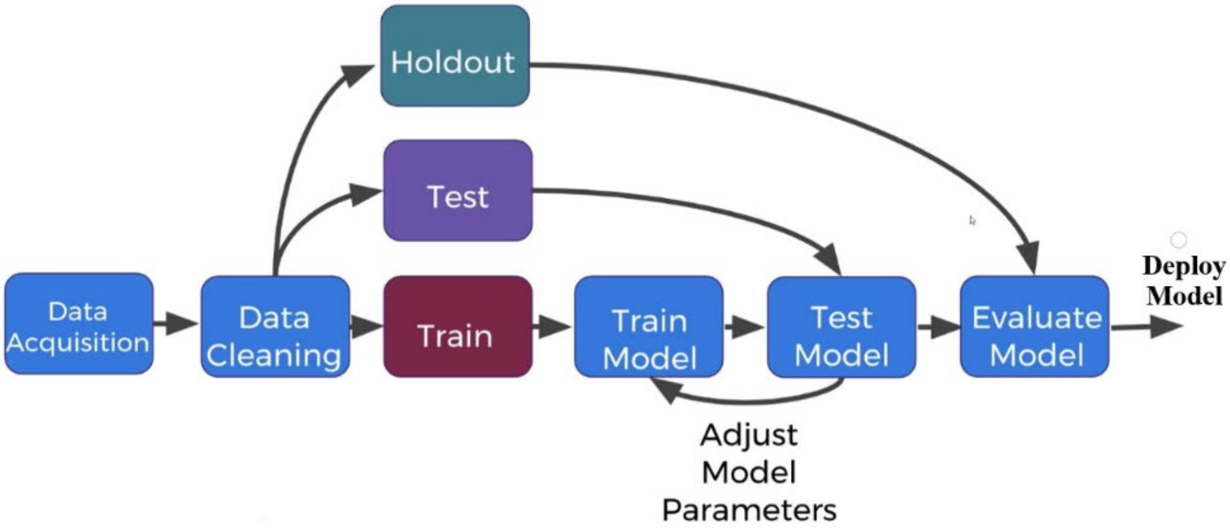

Figure 11: Proposed Model Development Process

E. *Dataset*

The proposed model's efficacy is evaluated on four public datasets encompassing the LivDet Competition dataset 2009, 2011, 2013, and 2015. LivDet competition is held every two years since 2009 and served as a platform for researchers to showcase their methodologies of averting fingerprint presentation attacks. Datasets contained fake fingerprints fabricated from cooperative and non-cooperative methods. Datasets obtained from LivDet2009 consist of four distinct datasets involving real fingerprints and spoof fingerprints made from silicone, playdoh, and gelatin. The LivDet2009 [38] datasets were captured with three fingerprint sensing devices encompassing Biometrika, CrossMatch, and Identix. LivDet2011 [39] consists of real fingerprints and fake fingerprints made from spoof materials such as Ecoflex, Gelatin, Latex, Silgum, Woodglue, and Playdoh. The resultant LivDet2011 datasets involved four fingerprint readers: Biometrika, Digital, Ital, and Sagem. The LivDet2013 [40] used the Biometrika and Ital-data reader to obtain six datasets, including real fingerprints and fake fingerprints fabricated from spoof materials ecoflex, gelatine, latex, modasil, and wood glue. LivDet2015 [41] utilised three fingerprint readers: Biometrika, Digital Persona, and GreenBit. The LivDet2015 consisted of real fingerprints and ten fake fingerprints encompassing ecoflex, gelatine, OOMOO, liquid-ecoflex, RTV, woodglue, body-double, and latex. The various salient properties of the datasets obtained from the LivDet 2009, 2011, 2013, and 2015 are summarised in TABLE *2*.

.

TABLE II
DETAILS OF LIVDET COMPETITION DATASETS UTILISED IN THE STUDY (THE TRAINING SETS WERE RESCALED TO 224 X 224, AND TWELVE (12) FINGERPRINT SPOOF MATERIALS WERE OBTAINED AND INCLUDED IN THE STUDY)

Dataset	Sensor	Image Size	Fingerprint Acquisition Approach	Real Fingerprints Available	Spoof Material
LivDet2009	Biometrika*	312 x 372	Cooperative	Yes	Silicone, Gelatin, Playdoh
	CrossMatch*	640 x 480			
	Identix*	720 x 720			
LivDet2011	Biometrika*	312 x 372	Cooperative	Yes	Ecoflex, Gelatin, Latex, Silgum, WoodGlue, Playdoh
	Digital Persona*	355 x 391			
	Ital_Data*	640 x 480			
	Sagem*	352 x 384			
LivDet2013	Biometrika*	312 x 372	Non-Cooperative	Yes	Ecoflex, Gelatine, Latex, Modasil, WoodGlue
	Ital_Data*	640 x 480			
LivDet2015	Biometrika*	1000 x 1000	Cooperative	Yes	Ecoflex, Gelatine, Latex, Liquid Ecoflex, RTV, WoodGlue, BodyDouble, Play-doh, OOMOO.
	Digital Persona*	252 x 324			
	GreenBit*	500 x 500			
	CrossMatch	640 x 480			

The study employed a stochastic gradient descent approach [36] in training the models with a batch size of 64 samples at a learning rate of $5\mathrm{e}^{-5}$ (0.00001). Due to the selected batch size, 750 iterations per epoch was executed for 50 epochs. Approximately 8 hours were spent on the model's training and validation, which involved 54,000 fingerprint samples. KAGGLE API was used, and its cloud-based resources were harnessed, involving its Graphical Processing Unit (GPU) and Central Processing Unit (CPU) of 16GB and 13GB, respectively.

IV. Experimental Results and Discussion

Five sets of evaluation parameters were utilised to probe the efficacy of the proposed model: these include training accuracy, validation accuracy, training loss, validation loss, and testing accuracy. Training and validation accuracies were parameters used to determine the accuracy level across the entire epoch during the model's training and validation. On the other hand, the training and validation losses were used for measuring the extent to which the model deviates from an optimal accuracy level across the various epochs. The testing accuracy was used to determine the model's general accuracy, utilising a confusion matrix to evaluate the proposed model's classification efficiency.

The training and validation processes were executed simultaneously across all the epochs, with the validation accuracy the most critical parameter. The validation dataset serves as the pre-test dataset that computes the model's efficiency during the training process. The resulting validation accuracy is an essential parameter because it is used to determine how effective each trained epoch performs. Also, validation accuracy adequately renders the model's robustness across the epochs. As a result, a checkpoint parameter predefined as "save-the-best-model-to-sign-classifier.h5" was set to save the epoch that records the best validation accuracy during the progressive training process. The best validation accuracy was recorded at "epoch 38".

At the best epoch, the model recorded a training accuracy of 0.9998 and validation accuracy of 0.9983, as shown in *Figure 12*. As demonstrated by *Figure 13*, training and validation losses of $9.0537e^{-4}$ and 0.0085 were recorded, respectively. Experimental results showcased that the model had relatively low losses, which resulted in state-of-the-art training and validation accuracies.

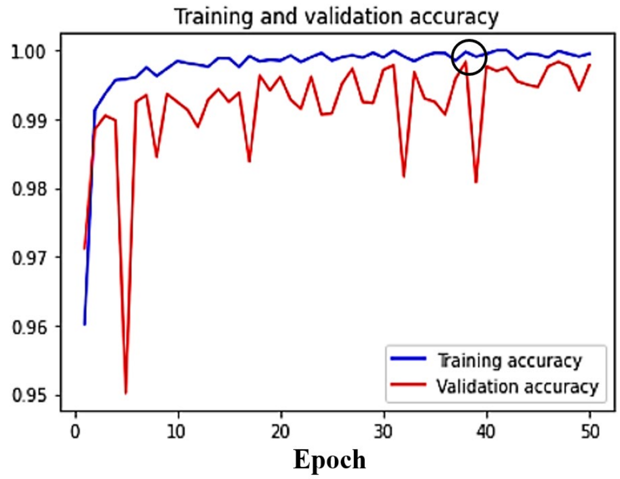

Figure 12: Training and Validation Accuracy of Model

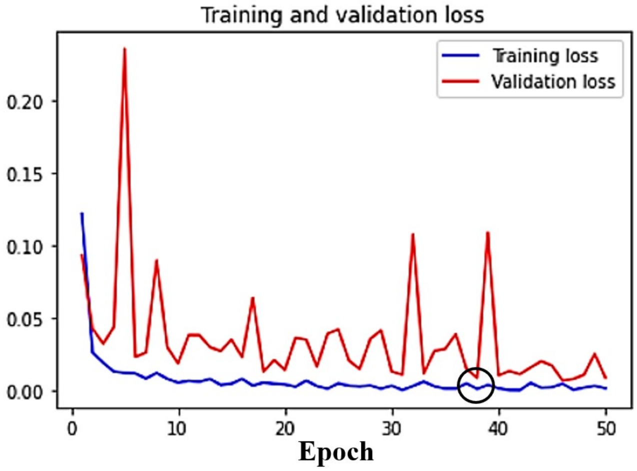

Figure 13: Training and Validation Accuracy of Model

As illustrated by the confusion matrix in *Figure 14*, the model successfully classified 2,992 real fingerprints and 2,993 fake fingerprints. Hence, only misclassifying 8 real fingerprints and 7 fake fingerprint samples.

The model's sensitivity (True Positive Rate (TPR)) and specificity (true negative rate (TNR)) in classifying real fingerprints and spoof fingerprints, respectively, were evaluated. These salient parameters were assessed using various variables: True-positive - test results that are classified as real fingerprints and are genuinely real fingerprints; False-positive - test results that are classified as real fingerprints but are fake fingerprints; True negative - test results that are fake fingerprints and genuinely fake fingerprints; and False-negative - test results that are classified as fake fingerprints but are real fingerprints.

True positives (**TP**) = 2992
False positive (**FP**) = 7
True negatives (**TN**) = 2993
False negative (**FN**) = 8

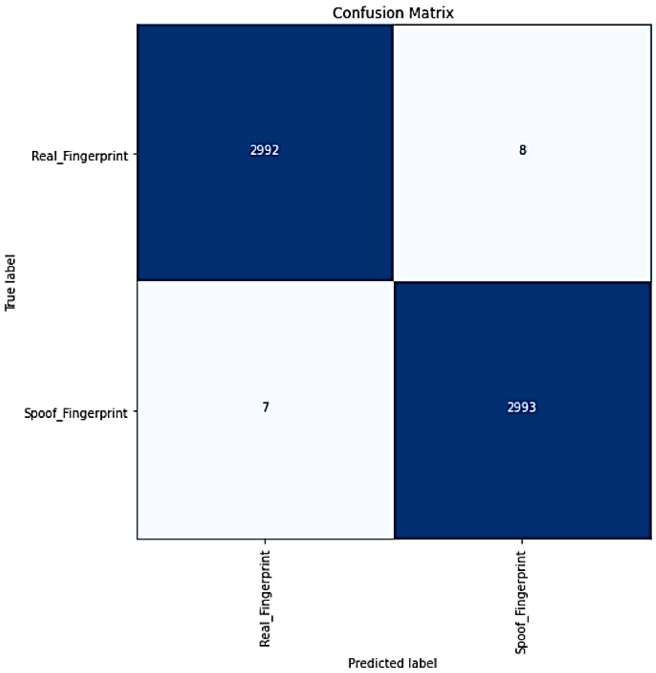

Figure 14: Confusion Matrix Evaluation of Proposed Model on Test Dataset

The following equation is used to calculate the model sensitivity and specificity:

$$\boldsymbol{Model\ Sensitivity} = \frac{\boldsymbol{TP}}{\boldsymbol{TP + FN}}\ \mathbf{x\ 100\%} \ldots\ldots\ldots\ldots\ldots.(\mathbf{1})$$

$$\boldsymbol{Model\ Specificity} = \frac{\boldsymbol{TN}}{\boldsymbol{TN + FP}}\ \mathbf{x\ 100\%}\ \ldots\ldots\ldots\ldots\ldots..(\mathbf{2})$$

The proposed model had a resultant sensitivity of 99.73% and a specificity of 99.77%, indicating the model's robustness.

To further validate the model's robustness, a group of fingerprint samples tagged as the holdout dataset, not used for the training, validation, or testing process, was introduced to verify its practical robustness. The model had a 100% classification accuracy on the holdout datasets, effectively classifying all fingerprint samples as depicted by *Figure 15*.

V. Conclusion

One of the severe threats battled by Automatic Fingerprint Identification Systems (AFIS) is the issue of presentation attack, which involves the malicious fabrication of synthetic fingerprints to circumvent AFIS. However, various approaches have been introduced to help remedy the threat of presentation attacks on AFIS through diverse hardware-based and software-based approaches. The hardware-based methods primarily deal with integrating specialised sensors to capture salient live human traits such as pulses, blood pressure, odour, etc. Nevertheless, these hardware-based approaches are easily bypassed with fabricated thin-layered spoofs. On the other hand, the software-based method eradicates additional sensors and instead engages in fingerprint feature extraction and cognitive learning schemes.

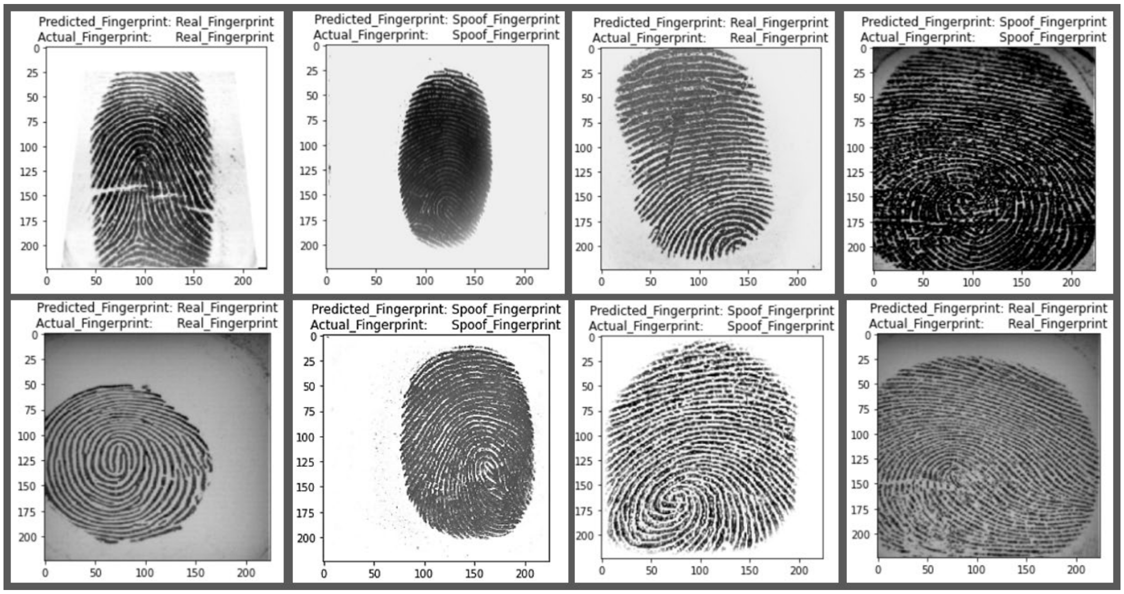
Predicted_Fingerprint: Real_Fingerprint
Actual_Fingerprint: Real_Fingerprint
Predicted_Fingerprint: Spoof_Fingerprint
Actual_Fingerprint: Spoof_Fingerprint
Predicted_Fingerprint: Real_Fingerprint
Actual_Fingerprint: Real_Fingerprint
Predicted_Fingerprint: Spoof_Fingerprint
Actual_Fingerprint: Spoof_Fingerprint
Predicted_Fingerprint: Real_Fingerprint
Actual_Fingerprint: Real_Fingerprint
Predicted_Fingerprint: Spoof_Fingerprint
Actual_Fingerprint: Spoof_Fingerprint
Predicted_Fingerprint: Spoof_Fingerprint
Actual_Fingerprint: Spoof_Fingerprint
Predicted_Fingerprint: Real_Fingerprint
Actual_Fingerprint: Real_Fingerprint
0
25
50
75
100
125
150
175
200
0
50
100
150
200

Figure 15: Proposed Model Classification of Holdout Dataset

Several software-based approaches have been posited to counter the issue of fingerprint presentation attacks. Most of these methods involve manual selection and extraction of handcrafted features, resulting in shallow features for spoof discrimination. Also, the CNN-based approaches, involving pre-trained-models such as VGGNet, AlexNet, MobileNet, Xception, ResNet etc., [7],[9]–[12], have some limitations on memory efficiency, computational efficiency, and feature reusability [14]. Hence, motivated by these concerns, we developed a spoof fingerprint detection model using a deep transfer learning approach, utilising the DenseNet201 network as a pre-trained model to discriminate against spoof fingerprints. Experiments were carried out on the LivDet competition standard database, encompassing the combination of datasets from LivDet 2009, 2011, 2013, and 2015, resulting in the acquisition of real fingerprints and fake fingerprints fabricated from twelve (12) different spoofing materials.

Experimental results manifested an adequate training accuracy of 0.9998 with training loss of $9.0537e^{-4}$ at the best epoch, indicating a state-of-the-art training performance. The validation accuracy that happens to be the principal parameter of interest for selecting the best epoch recorded a validation accuracy of 0.9983 and a validation loss of 0.0085. Cumulatively, 60,000 datasets were utilised to develop the model, with a dataset split ratio of 8:1:1 for training, validation and testing, respectively. The developed model showcased an average classification accuracy of 99.8%, indicating state-of-the-art classification accuracy. The model manifested a sensitivity of 99.73% and specificity of 99.77%. Further test on the holdout dataset (i.e., independent of training, validation, and test datasets) validated the model's robustness, with the model effectively classifying all holdout dataset accurately.

REFERENCES

[1] C. Sousedik and C. Busch, "Presentation attack detection methods for fingerprint recognition systems : a survey," *IET Biometrics*, no. November, pp. 219–233, 2013, doi: 10.1049/iet-bmt.2013.0020.

[2] L. Ghiani, D. A. Yambay, V. Mura, G. L. Marcialis, F. Roli, and S. A. Schuckers, "Review of the Fingerprint Liveness Detection (LivDet) Competition Series: 2009 to 2015," *Image Vis. Comput.*, 2016, doi: 10.1016/j.imavis.2016.07.002.

[3] E. Marasco and C. Sansone, "Combining perspiration- and morphology-based static features for fingerprint liveness detection," *Pattern Recognit. Lett.*, vol. 33, no. 9, pp. 1148–1156, 2012, doi: 10.1016/j.patrec.2012.01.009.

[4] C. Gottschlich, A. Y. Yang, and U. C. Berkeley, "Fingerprint Liveness Detection based on Histograms of Invariant Gradients," *IEEE Int. Jt. Conf. biomet- rics, pages 1–7. IEEE, 2014*, 2015.

[5] M. Espinoza and C. Champod, "Using the Number of Pores on Fingerprint Images to Detect Spoofing Attacks," *IEEE*, 2011.

[6] C. Yuan *et al.*, "Semi-supervised stacked autoencoder-based deep hierarchical semantic feature for real-time fingerprint liveness detection," *J. Real-Time Image Process.*, vol. 17, no. 1, pp. 55–71, 2020, doi: 10.1007/s11554-019-00928-0.

[7] D. Menotti *et al.*, "Deep Representations for Iris, Face, and Fingerprint Spoofing Detection," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 4, pp. 864–879, 2015, doi: 10.1109/TIFS.2015.2398817.

[8] R. F. Nogueira, R. D. A. Lotufo, and R. C. Machado, "Fingerprint Liveness Detection using Convolutional Neural Networks," *IEEE Trans. Inf. Forensics Secur.*, vol. 6013, no. c, 2016, doi: 10.1109/TIFS.2016.2520880.

[9] F. Pala and B. Bhanu, "Deep triplet embedding representations for liveness detection," *Adv. Comput. Vis. Pattern Recognit.*, vol. PartF1, pp. 287–307, 2017, doi: 10.1007/978-3-319-61657-5_12.

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.

[11] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 1800–1807, 2017, doi: 10.1109/CVPR.2017.195.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.

[13] S. Woo, J. Park, J. Lee, and I. S. Kweon, "CBAM: Convolutional Block Attention Module," *Eccv*, 2018.

[14] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 2261–2269, 2017, doi: 10.1109/CVPR.2017.243.

[15] E. Park, X. Cui, T. Hai, B. Nguyen, and H. Kim, "Presentation Attack Detection Using a Tiny Fully Convolutional Network," *IEEE Trans. Inf. Forensics Secur.*, vol. PP, no. c, p. 1, 2019, doi: 10.1109/TIFS.2019.2907184.

[16] Z. Xia, R. Lv, Y. Zhu, P. Ji, and H. Sun, "Fingerprint liveness detection using gradient-based texture features," *Signal, Image Video Process.*, 2016, doi: 10.1007/s11760-016-0936-z.

[17] I. Goicoechea-telleria, K. Kiyokawa, J. L. I. U. Jimenez, and R. Sanchez-reillo, "Low-Cost and Efficient Hardware Solution for Presentation Attack Detection in Fingerprint Biometrics Using Special Lighting Microscopes," *IEEE Access*, vol. 7, pp. 7184–7193, 2019, doi: 10.1109/ACCESS.2018.2888905.

[18] T. Chugh, K. Cao, and A. K. Jain, "Fingerprint Spoof Buster : Use of Minutiae-centered Patches," *IEEE Trans. Inf. FORENSICS Secur.*, vol. 6013, no. c, pp. 1–13, 2018, doi: 10.1109/TIFS.2018.2812193.

[19] R. K. Dubey, J. Goh, and V. L. L. Thing, "Fingerprint Liveness Detection From Single Image Using Low Level Features and Shape Analysis," *IEEE Trans. Inf. Forensics Secur.*, vol. 6013, no. c, 2016, doi: 10.1109/TIFS.2016.2535899.

[20] G. L. Marcialis, F. Roli, and A. Tidu, "Analysis of fingerprint pores for vitality detection," *Proc. - Int. Conf. Pattern Recognit.*, pp. 1289–1292, 2010, doi: 10.1109/ICPR.2010.321.

[21] D. Gragnaniello, G. Poggi, C. Sansone, and L. Verdoliva, "Fingerprint liveness detection based on Weber Local image Descriptor," *2013 IEEE Work. Biometric Meas. Syst. Secur. Med. Appl. BioMS 2013 - Proc.*, pp. 46–50, 2013, doi: 10.1109/BIOMS.2013.6656148.

[22] L. Ghiani, G. L. Marcialis, and F. Roli, "Fingerprint liveness detection by local phase quantisation," *Proc. - Int. Conf. Pattern Recognit.*, pp. 537–540, 2012.

[23] L. Ghiani, A. Hadid, G. L. Marcialis, and F. Roli, "Fingerprint Liveness Detection using Binarised Statistical Image Features," *IEEE 6th Int. Conf. Biometrics Theory, Appl. Syst. BTAS 2013*, 2013, doi: 10.1109/BTAS.2013.6712708.

[24] T. Ojala, M. Pietikäinen, and D. Harwood, "Performance evaluation of texture measures with classification based on Kullback discrimination of distributions," *Proc. - Int. Conf. Pattern Recognit.*, vol. 3, pp. 582–585, 1994, doi: 10.1109/ICPR.1994.576366.

[25] S. Banerjee and S. Chaudhuri, "DeFraudNet : End2End Fingerprint Spoof Detection using Patch Level Attention," *2020 IEEE Winter Conf. Appl. Comput. Vis.*, pp. 2695–2704, 2020.

[26] T. Chugh, K. Cao, and A. K. Jain, "Fingerprint spoof buster: Use of minutiae-centred patches," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 9, pp. 2190–2202, 2018, doi: 10.1109/TIFS.2018.2812193.

[27] A. G. Howard *et al.*, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv*, 2017.

[28] K. Zhang, Y. Guo, X. Wang, J. Yuan, Z. Ma, and Z. Zhao, "Channel-Wise and Feature-Points Reweights Densenet for Image Classification," *Proc. - Int. Conf. Image Process. ICIP*, vol. 2019-Septe, pp. 410–414, 2019, doi: 10.1109/ICIP.2019.8802982.

[29] K. Zhang, Y. Guo, X. Wang, J. Yuan, and Q. Ding, "Multiple feature reweight DenseNet for image classification," *IEEE Access*, vol. 7, pp. 9872–9880, 2019, doi: 10.1109/ACCESS.2018.2890127.

[30] Y. Yuan *et al.*, "Prostate Segmentation With Encoder-Decoder Densely Connected Convolutional Network (Ed-DenseNet)," *2019 IEEE 16th Int. Symp. Biomed. Imaging (ISBI 2019)*, no. Isbi, pp. 434–437, 2019.

[31] L. Wang, L. Qiu, W. Sui, and C. Pan, "Reconstructed Densenets For Image Super-Resolution," *2018 25th IEEE Int. Conf. Image Process.*, no. 61773377, pp. 3558–3562, 2018.

[32] A. George and S. Marcel, "Deep pixel-wise binary supervision for face presentation attack detection," *Int. Conf. Biometrics, number CONF, 2019*, 2019.

[33] C.-M. P. Lian Huang, "Audio Replay Spoof Attack Detection Using Segment-Based Hybrid Feature And DenseNet-LSTM Network," *ICASSP 2019-2019 IEEE Int. Confer- ence Acoust. Speech Signal Process.*, pp. 2567–2571, 2019.

[34] "Deep Transfer Learning for Image Classification | by Vegard Flovik | Towards Data Science." https://towardsdatascience.com/deep-transfer-learning-for-image-classification-f3c7e0ec1a14 (accessed Nov. 03, 2020).

[35] S. Panigrahi, A. Nanda, and T. Swarnkar, "A Survey on Transfer Learning," *Smart Innov. Syst. Technol.*, vol. 194, pp. 781–789, 2021, doi: 10.1007/978-981-15-5971-6_83.

[36] Y. Zhang, D. Shi, X. Zhan, D. I. Cao, and K. Zhu, "Slim-ResCNN : A deep Residual Convolutional Neural Network for Fingerprint Liveness Detection," *IEEE Access*, vol. PP, p. 1, 2019, doi: 10.1109/ACCESS.2019.2927357.

[37] "Transfer learning from pre-trained models | by Pedro Marcelino | Towards Data Science." https://towardsdatascience.com/transfer-learning-from-pre-trained-models-f2393f124751 (accessed Nov. 03, 2020).

[38] G. L. Marcialis *et al.*, "First International Fingerprint Liveness Detection Competition — LivDet 2009," pp. 1–12, 2009.

[39] D. Yambay, L. Ghiani, P. Denti, G. L. Marcialis, F. Roli, and S. Schuckers, "LivDet 2011 – Fingerprint Liveness Detection Competition 2011," pp. 1–8, 2011.

[40] L. Ghiani *et al.*, "LivDet 2013 Fingerprint Liveness Detection Competition 2013," *2013 Int. Conf. Biometrics*, pp. 0–5, 2013.

[41] V. Mura, L. Ghiani, G. L. Marcialis, F. Roli, D. A. Yambay, and S. A. Schuckers, "LivDet 2015 Fingerprint Liveness Detection Competition 2015," 2015.

Divine S. Ametefe received his B.Tech. Degree in Telecommunication Engineering from Ghana Technology University College (GTUC), Ghana, 2017. He was honoured with an MSc. Degree in Telecommunication and Information Engineering by Universiti Teknologi MARA (UiTM) Malaysia, in 2019. He is currently pursuing a PhD program in Electrical Engineering at Universiti Teknologi MARA (UiTM), Malaysia. His interest areas include biometrics, fingerprint spoof detection, pattern recognition, neural networks, deep learning, and the Internet of Things (IoT).

Suzi S. Sarnin received her Bachelor of Electrical and Electronics (B.Eng.) in Communication from the Universiti Teknologi Malaysia, Skudai, Malaysia, in 1999. She completed her Master in Microelectronics (MSc) from the Universiti Kebangsaan Malaysia in 2005. She has a PhD in Electrical Engineering from Universiti Teknologi MARA, Shah Alam, Malaysia, and is currently a senior lecturer at the Universiti Teknologi MARA and collaborates actively in several disciplines of Electrical Engineering.

Darmawaty M. Ali is an Associate Professor at Universiti Teknologi MARA (UiTM), Selangor, Malaysia. She obtained her PhD in 2012 from Universiti Malaya, Malaysia. She received her Master of Electrical Engineering in 2002 from Universiti Teknologi Malaysia. Previously, she earned her first degree from Universiti Kebangsaan Malaysia with Honours in Electrical, Electronic, and System, graduating in 1999. She is the head of Wireless Communication Technology (WiCOT) Research Interest Group (RIG), and her research interests include Wireless Access Technology and Quality of Service in Wireless Broadband.