Deep Learning-Based Countermeasures for Fingerprint Spoof Detection

Amal Almuarik *amalalmuarik@gmail.com*
Faculty of Computer and Information Sciences
Computer Science Department, King Saud University
Riyadh, Saudi Arabia

Mashael Aldughayem *mashaelAldughayem@gmail.com*
Faculty of Computer and Information Sciences
Computer Science Department, King Saud University
Riyadh, Saudi Arabia

Munirah Alshathri *munirah.alsh3@gmail.com*
Faculty of Computer and Information Sciences
Computer Science Department, King Saud University
Riyadh, Saudi Arabia

Nouf Alrowais *nouf.n1114@gmail.com*
Faculty of Computer and Information Sciences
Computer Science Department, King Saud University
Riyadh, Saudi Arabia

Ouiem Bchir *obchir@ksu.edu.sa*
Faculty of Computer and Information Sciences
Computer Science Department, King Saud University
Riyadh, Saudi Arabia

Abstract

Biometric systems have become a crucial part of modern life, offering secure and seamless authentication. However, as people increasingly share their biometric data, they may not fully understand the associated risks. Fingerprint-based security systems, while popular, are vulnerable to spoofing attacks where counterfeit fingerprints made from materials like latex, silicone, or gelatin can deceive the system. This paper addresses the need for improved security by proposing a new countermeasure for detecting such spoof attacks. The solution utilizes deep learning models to differentiate between genuine and spoof fingerprints. Four models—ResNet, GoogLeNet, DenseNet, and Vision Transformer—were evaluated. The study found that ResNet34 and DenseNet169 consistently outperformed the other models on the ATVS and LivDet2015 datasets, achieving higher True Detection Rates (TDR) at a False Detection Rate (FDR) of 0.05, as well as better F1 scores and Average Classification Error (ACE). Additionally, these models demonstrated lower time complexity, making them both effective and efficient for detecting fingerprint spoofing. This research highlights the importance of incorporating advanced deep learning techniques to enhance the reliability and security of fingerprint authentication systems.

Keywords: Biometric Verification Systems, Fingerprint Spoof Attacks, Genuine Fingerprint, Spoof Fingerprint, Deep Learning, ResNet, GoogLeNet, DenseNet, Vision Transformer.

1 INTRODUCTION

The global adoption of security verification systems is an ever-expanding concern the world is facing (Sandouka et al., 2020). Specifically, biometric systems, which are designed to identify and

analyze physical or behavioral characteristics of an individual for the purpose of recognition, are increasingly adopted as security-based verification systems. These biometric physical characteristics can come in the form of facial features, iris patterns, voice patterns, fingerprints, etc. They are employed in a wide range of applications, including law enforcement, Border control, Healthcare, and the Automotive industry (Guennouni et al., 2020). One of the most adaptive forms of biometrics in verification systems is the fingerprint security system. The latter is widely used to verify the user's identity by analyzing fingerprint patterns (Sharma et al., 2015). Due to their unique and reliable nature, fingerprint-based systems are widely applied in access control, attendance tracking, crime investigation (Ali et al., 2016) and continue to expand into areas such as contactless biometrics (Djara et al., 2015), high-security twin identification (CN et al., 2013), and synthetic fingerprint generation (Bárta, 2016).

However, this increased adaptability makes them vulnerable to spoofing attacks. Specifically, they can be altered using fake fingerprints fabricated using materials such as latex, silicone, or gelatin, to mimic a genuine user fingerprint. These types of security attacks are known as Presentation Attacks (PAs) (Sandouka et al., 2020). The latter are defined as any attempt to present fake biometric data to fool the security system (Sandouka et al., 2020). PAs are among the most predominant security attacks faced by fingerprint security-based verification systems (Lee et al., 2022). This is due to the availability of the fabrication material, the ease of the process, and the success of the replication. These threats compromise the safety of the user information, the stored data, and the application itself.

Hardware devices, particularly optical fingerprint sensors, are commonly used to sense vital cues such as blood flow, temperature, and heart pulses to differentiate genuine fingerprints from spoofed ones. Nevertheless, these devices are hard to implement and expensive (Sandouka et al., 2020). Software countermeasures are alternative solutions to hardware ones. Specifically, they aim to differentiate between spoof and genuine fingerprints through the use of image processing and pattern recognition techniques. As a result, the need for a robust fingerprint spoof attack detection system is imperative to reinforce security and ensure fingerprint authenticity.

In this work, we propose a software-based countermeasure that is able to detect the liveness of the fingerprint. In other words, this countermeasure is designed to distinguish between genuine and spoof fingerprints through the use of image processing and Deep Learning (DL) techniques. For this purpose, we investigate Deep Learning Convolutional Neural Networks (DL CNNs) such as ResNet (Chen et al, 2021), GoogLeNet (Marasco et al., 2016), and DenseNet (Wen et al., 2020), in addition to Visual Transformers (Liu et al., 2022), such as ViT (Dosovitskiy et al., 2020).

2 LITERATURE REVIEW

2.1 Fingerprint Presentation Attack Detection Systems Based on ResNet

In 2023, Rai et al. proposed a fingerprint presentation attack detection system based on dual network (ResNet) architecture called Slim-ResCNN. Slim-ResCNN employs an improved version of residual blocks. The improved residual blocks, ("block_b") don't incorporate an activation function (ReLU) at the second convolution as the original residual block ("block_a"). This change helps maintain the representation of the low-dimensional layers. Moreover, the convolution kernels adopted in the improved residual ("block_b") are broadened by a factor of two, and a dropout layer is inserted between each pair of convolutional layers. This modification enhances the model's generalization by preventing it from relying heavily on local features. For high-dimensional layers, an improved residual ("block_d") is used, replacing the 1x1 convolutional layer with a zero-padding channel layer as in ("block_c"). This change does not introduce additional parameters or reduce the efficiency of reverse gradient flow.

In 2019, Zhang et al. proposed an approach that begins by extracting the foreground region from the entire fingerprint image using a statistical histogram of both rows and columns. This allows the identification of relevant fingerprint patterns. Within the identified foreground region, local patches are segmented using the center of gravity (CoG). This method ensures that local patches are selected intelligently, considering the distribution of pixel values within the fingerprint ridges

and valleys. The training data is augmented by applying data augmentation techniques. This involves flipping and rotating the local patches before feeding them into the Slim-ResCNN model. During the testing stage, the trained Slim-ResCNN model is used to assess the liveness of the input fingerprint image. As such, it predicts a score with respect to each extracted patch to determine whether the corresponding fingerprint is from a genuine "alive" entity or a "spoof" artifact. The Slim-ResCNN approach won the first prize in the Fingerprint Liveness Detection Competition 2017 with an overall accuracy of 95.25% (Mura, V., 2018).

Similarly, ResNet has been adopted in the EXPRESSNET framework proposed by Rai et al., (2023).In this approach, the input fingerprint image is first resized to 512 × 512 pixels and then passed through a heatmap generator block, which forms the core of the proposed architecture. Specifically, the heatmap generator block is responsible for generating heatmaps to aid in the classification process. The latter includes two main components: the Encoder-Decoder component and the Channel Attention component. The encoder-decoder component localizes spatially relevant features. While the Channel Attention component focuses on relevant information in the image. As such, the heatmap reflects the importance of each pixel in the fingerprint image. This heatmap is then fed into the modified ResNet (Slim ResNet) for final classification. In the Slim ResNet model, the depth of the network is reduced without compromising its spatial properties to reduce computational complexity. The original ResNet architecture consists of four building blocks, each incorporating three convolutional layers. Specifically, the first block is repeated 3 times, the second one 4 times, the third one 6 times, and the fourth one 3 times, whereas, in the case of Slim ResNet, the number of block repetitions is reduced to 2 times for the first block, 3 times for the second, 4 times for the third, and 2 times for the fourth block.

In 2017, Yuan et al. introduced an approach that integrates the Principal Component Analysis (PCA) technique into the convolutional neural network model for fingerprint spoofing detection. The PCA modules aim at addressing the issues related to feature extraction, overfitting, and dimensionality reduction. Specifically, PCA is performed on each feature map. The obtained features from the Convolutional Neural Network (CNN) are subjected to dimensionality reduction via Principal Component Analysis (PCA), are then fused, and fed to a Support Vector Machine (SVM) classifier (Brereton, 2010). The latter categorizes the extracted features and determines whether a fingerprint is genuine or counterfeit.

2.2 Fingerprint Presentation Attack Detection Systems Based on CNN-Inception v3

In 2017, Chugh et al. combined theCNNInception-v3 (GoogLeNet) model with minutiae-based local patches. By leveraging local patches of size p × p (with p = 96), 48 patches are extracted from each fingerprint image. These yields capture relevant information that distinguishes between spoof and genuine fingerprints. During the offline training phase, the system performs minutiae detection, extracts local patches centered around minutiae locations, and trains CNN models on these patches. Alternatively, in the online testing phase, GoogLeNet predicts a spoofing score for each patch constituting the input image. The spoof detection decision is made by computing the average of the spoofing scores learned with respect to these patches.

Similarly, Tabassi et al. (2018) employed the Inception-v3 Convolutional Neural Network model (CNN) on extracted local patches centered around fingerprint minutiae. In this approach, the Inception-v3 CNN is employed twice. The first instance is trained on the whole input fingerprint image to classify it as genuine (valid) or spoofed (altered). Fingerprints identified as spoofed are then examined using a patch-based strategy, as proposed by Chugh et al. (2017), to identify altered regions within the fingerprint. Furthermore, data augmentation techniques—such as the use of a Generative Adversarial Network (GAN)—are employed to generate synthetically altered fingerprint samples.

2.3 Fingerprint Presentation Attack Detection Systems Based on DenseNet

In2020, Zhang et al., adopted a DenseNet architecture to devise a new network architecture named FLDNet. As part of the preprocessing, the foreground region is extracted from the fingerprint image, and five fixed-size local patches centered around the center of gravity are

selected. These patches are subsequently used as training samples for the FLDNet. Compared to the original DenseNet, FLDNet incorporates a novel architectural component—referred to as D&R—that integrates both the residual and densely connected paths. Specifically, the channel-wise concatenation in the middle of the original dense block is replaced by an element-wise addition, followed by an average pooling layer.

Similarly, Wenet al. (2020) employed a DenseNet-based architecture along with ROI extraction to eliminate the background and noise from the fingerprint image. Specifically, one-pixel-wide ridgelines are utilized to reduce offset issues caused by thicker ridgelines. Moreover, short ridgeline segments are removed to minimize offset stemming from noisy data. Concerning DenseNet, both structural and non-structural hyperparameters are optimized using a genetic algorithm to search for optimal solutions. As such, a DenseNet structure candidate is represented using four gene fragment chromosomes on which operations like crossover, mutation, and selection are employed. Over successive generations, the classification accuracy of the population gradually improves, ultimately resulting in an optimized network configuration: {(0.1,0.4,20) -(12,8,10)-(5,5,7)-(96,32,96)}.

2.4 Fingerprint Presentation Attack Detection Systems Based on Attention Mechanism

In 2023, Kothadiya et al. proposed a model that combines five convolution layers with sequential attention modules. In particular, the convolution layers are based on the ResNet50 architecture (Vaswani et al., 2017), integrating residual modules, skip connections, and batch normalization to avoid the vanishing gradient problem, speed up the convolution operation, and normalize features before applying to ReLU's activation function. The attention modules employed in He et al. (2016) are used to enhance the feature learning process. The attention mechanism consists of a Channel Attention (CA) block, a Spatial Attention (SA) block, and a mixed attention block. In particular, the CA block computes the relevant features across distinct channels, while the SA block provides global contextual information such as texture and background.

In 2023, Grosz et al. proposed a fingerprint presentation attack detection (PAD) system based on a pre-trained Vision Transformer (ViT) model that focuses on the local features of the fingerprint images. As such, a 16-patch ViT is employed to encode local patches of the fingerprint image. Specifically, a small ViT of depth 12, including 6 attention heads, is adopted. The output of each layer is conveyed to a multilayer perceptron (MLP) classifier with two intermediate layers to categorize genuine or spoofed fingerprints. The purpose of employing a classifier at each layer is to empirically determine the optimal number of layers. This PAD system is jointly used with another ViT for the purpose of fingerprint recognition in a unified framework.

In the same year, Zhang et al. (2023) proposed a lightweight fingerprint presentation attack detection model based on ResNet and a self-attention mechanism. First, as a preprocessing step, foreground segmentation is performed to discard the background. Then, local patches from the foreground are extracted. Upon that, they are conveyed to the deep learning model. The latter is derived from the Slim-ResCNN network (Rai et al.,2023). Moreover, to enhance the model's attention to the fingerprint ridge structure, a self-attention mechanism is incorporated. As such, ResNet extracts abstract information through convolution layers, and the self-attention layers summarize and process high-level semantics in the obtained feature maps. In order to provide the system with the ability to generalize to new or unfamiliar spoof materials, a CycleGAN-based method proposed by Zhu et al. (2017) is employed. It aims at transferring style information from known spoof materials to synthesize convincing fake fingerprint images.

Recently, Vurity et al. (2025) introduced a hybrid finger photo PAD architecture called ColFigPhotoAttnNet1 that combines MobileNetV3-based feature extraction with Swin Transformer window attention followed by residual blocks initialized from ResNet34. The model processes features from three color spaces and applies dynamic quantization to reduce parameter count. While the approach reports strong intra- and inter-capture performance across datasets from various mobile devices, the reliance on multi-color fusion and quantization introduces a trade-off between generalization and model compactness. Furthermore, the evaluation remains specific to

finger photo datasets and does not explore classical fingerprint PAD scenarios involving sensors or LivDet benchmarks.

2.5 Overfitting and Interpretability in Fingerprint Presentation Attack Detection Systems

In 2024, Reza & Jung (2023) proposed an ensemble learning approach with activation perturbation, where noise is added to class activation map regions during training to diversify the learning process and enhance model robustness. In the same year, Yuan et al. introduced a Siamese Attention Residual Convolutional Neural Network (Res-CNN) designed to analyze the fundamental differences between genuine and spoof fingerprints, focusing on ridge continuity features. This model improves interpretability by providing visual cues that help distinguish between genuine and spoof fingerprints, thereby improving the system's reliability.

More recently, Carta et al. (2025) explored the use of dimensionality reduction methods to project high-dimensional data into two-dimensional spaces. This technique facilitates visual inspection and aids in identifying weaknesses in the decision boundaries learned during training. Such approaches contribute not only to model transparency but also to improving understanding of classification behavior and potential vulnerabilities in fingerprint PAD systems.

3 DISCUSSION

In related work, various deep learning approaches have been employed to address fingerprint spoofing attacks. Specifically, the most frequently used convolutional neural network (CNN) models are ResNet (Chen et al., 2021), Inception-V3 (Marasco et al., 2016), and DenseNet (Wen et al., 2020). ResNet has been applied to local patches in studies by Rai et al. (2023) and Zhang et al. (2019) and to full fingerprint images in the study by Yuan et al. (2017). Notably, Rai et al. (2023) used an enhanced version of residual blocks, while Yuan et al. (2017) incorporated principal component analysis (PCA) for feature extraction from each feature map. Inception-V3 was employed on local patches by Chugh et al. (2017) and on full fingerprint images by Tabassi et al. (2018). Similarly, DenseNet was implemented on local patches by Zhang et al. (2020), while Wen et al. (2020) applied it to full fingerprint images. Attention mechanisms have also been explored in recent studies. Specifically, spatial and channel attention blocks are incorporated between convolutional layers in the model proposed by Kothadiya et al. (2023). Zhang et al. (2023) introduced a self-attention mechanism after convolutional layers to enhance sensitivity to ridge structures. Furthermore, Grosz et al. (2023) employed the Vision Transformer (ViT) model (Dosovitskiy et al., 2020), while Vurity et al. (2025) proposed a hybrid model combining CNN with the Swin Transformer architecture.

Most of the reported works adopted accuracy as a performance measure, except the work by Tabassi et al. (2018) which used the False Discovery Rate (FDR) and the False Positive Rate (FPR). Moreover, most of the related work assessed their systems on different versions of LivDet dataset [Mura, V. (2018); Marcialis et al., 2009;Yambay et al., 2012; Ghiani et al., 2013; Mura et al., 2015]. However, the ATVS (Galbally et al., 2010) has also been utilized. Nonetheless, as one can notice, the reported works did not adopt the same samples as the number of instances used is different.

3.1 Deep Learning Paradigms for Fingerprint Spoof Detection

The need to reinforce biometric security against presentation attacks in a world full of cyberattacks is imperative. As such, there have been research efforts to investigate deep learning models and image processing techniques to automatically detect the impersonation of a fingerprint. Specifically, most of the studies have focused on Convolutional Neural Network (CNN) models, and limited attention has been given to the Vision Transformer (ViT) architecture. Moreover, despite the various approaches that have been proposed, there is still a need for a comprehensive study comparing the performance of the CNN models and the ViT model in the context of fingerprint presentation attack detection. In fact, although CNN models proved to be successful across a range of image-related tasks, it's undeniable that ViT has demonstrated remarkable results on the well-known ImageNet benchmark dataset (Russakovsky et al.,2015).

Furthermore, fingerprint presentation attacks haven't been thoroughly investigated in a high level of abstraction that incorporates contextual information and mining a large range of dependencies. In fact, only Zhang et al. (2023) alleviated this depth of analysis through the adaptation of the recently proposed ViT model, and two others, Kothadiya et al. (2023) and Zhang et al. (2023), have only integrated the attention mechanism unit into the CNN models.

This paper aims to explore these aspects by examining the feasibility of designing a fingerprint presentation attack countermeasure based on the ViT model. The latter has demonstrated remarkable results on well-known benchmark datasets, which makes it a promising choice for improving biometric security. Besides, this work intends to compare the performances of the ViT-based model with the well-known CNN models. These are ResNet (He et al., 2016), GoogleNet (Szegedy et al., 2015), and DenseNet (Huang et al., 2017).

3.2 ResNet

Residual Network (ResNet) (He et al., 2016) is a deep neural network architecture based on convolutional layers. It is constructed by stacking multiple residual blocks together. ResNet introduced the concept of residual blocks to address the vanishing gradient problem. The residual blocks are based on skip connections. In fact, contrary to traditional convolution layers, where each layer is conveyed to the following one, the residual block layer is conveyed to the following layer and straightway to the one next to it using skip connections. They enable connecting directly one layer to deeper layers by skipping some intermediate layers in between. This allows training very deep networks, which was challenging due to vanishing gradients.

3.3 GoogLeNet (Inception V3)

Inception-V3 is an improved version of GoogLeNet (Szegedy et al., 2015). It introduced several architectural optimizations to reach a balance between computational efficiency and accuracy in deep neural networks. A major characteristic of GoogLeNet is its unique architectural components, which include inception modules and reduction blocks, serving as the fundamental building blocks of the network. The structure of an inception module incorporates three convolution layers. These modules perform parallel feature maps' extraction through the employment of 1x1, 3x3, and 5x5 convolutional kernels, along with max pooling, to enhance the network's capabilities and performance. The 1x1 Convolutional Layer is employed for dimensionality reduction, capturing local features, and effectively reducing the number of parameters and computational load. Alternatively, the 3x3 convolutional layer captures spatial information and extracts features to recognize patterns of moderate complexity. On the other hand, the 5x5 Convolutional Layer captures larger spatial patterns and more intricate features to enhance the network's ability to recognize a diverse range of objects and patterns. Additionally, the Max-Pooling Layer downsamples the feature maps and performs feature selection. The initial phase of the Inception V3 architectural model involves a series of convolutional layers that extract intricate features from the input image (Ghojogh & Ghodsi, 2023). Upon that, a combination of average pooling and max-pooling operations is incorporated. This combination serves to downsample the feature maps while preserving the most salient information.

3.4 DenseNet

Densely Connected Convolutional Network (DenseNet) (Huang et al., 2017)is a deep convolutional network that incorporates shorter connections between layers close to the input and those closer to the output, ensuring maximal information flow between the network's layers. In fact, unlike conventional feed-forward networks, where each layer is only connected to the subsequent one, DenseNet introduces the concept of dense connectivity. Specifically, DenseNet feeds every layer with feature maps from all preceding layers, thereby facilitating extensive information exchange. The output of each convolutional layer is processed using a composite function comprising batch normalization (BN), rectified linear units (ReLUs), and a 3x3 convolution operation. Moreover, the growth rate hyperparameter determines the number of preceding feature maps contributing to the current layers. The output of the convolutional layers x_0, x_1, x_2, and $x_3$contributes to the output of x_4. Furthermore, a transition layer is integrated between dense blocks. It includes batch normalization, 1x1 convolution, and 2x2 average pooling.

3.5 Vision Transformer (ViT)

Vision Transformer (ViT) is an extension of the Transformer architecture that is specialized in image classification tasks. It employs the self-attention mechanism to learn context correlations within the image. ViT comprises N encoder transformer blocks. Each block contains multi-head self-attention, multi-layer perceptron, and residual connections (Dosovitskiy et al., 2020).

Typically, Transformers integrate an attention mechanism that computes the association between two tokens. While these tokens consist of words in the context of NLP problems, they consist of pixels in the context of image classification. Nevertheless, measuring the correlation of each pair of pixels is exceedingly expensive. Consequently, ViT measures the correlation between image patches, where each patch is a small block of the image. The patches are flattened and ordered in a sequence. Upon that, they are embedded. The obtained vector is combined with the position embedding and conveyed to the transformer. ViTs can come in different versions, including Swin (Liu et al., 2021), T2T-ViT (Yuan et al., 2021), and DeiT (Touvron et al., 2020).

We propose to compare the performance of different CNN models, including ResNet (He et al., 2016), Inception V3 (Szegedy et al., 2015), and DenseNet (Huang et al., 2017), as well as the performance of the Vision Transformer (ViT) (Dosovitskiy et al., 2020), to determine if the Vision Transformer (ViT) outperforms them in categorizing fingerprint images as genuine or spoof. To achieve this, we will train each model using images of both spoof and genuine fingerprints.The considered CNN models and the Vision Transformer will be evaluated using the test set. The different models will be assessed to determine the best-performing model. The model that performs the best will be chosen to build the required system.

4 EXPERIMENTS

4.1 Dataset Description

4.1.1 LivDet

The LivDet 2015 (Mura et al., 2015) dataset is a publicly available—accessible upon agreement—dataset for evaluating the performance of fingerprint liveness detection algorithms. The dataset contains a large collection of genuine and spoof fingerprint images collected using various sets of sensors, including CrossMatch, Digital Persona, Greenbit, and Hi-Scan sensors. The dataset includes 9,000 genuine and 10,198 spoofed fingerprint instances. The training dataset contains 4500 genuine and 4250 spoof images. Whereas testing data contains 4500 genuine and 5948 spoof images that are fabricated using different materials: Ecoflex, gelatin, latex, etc. In addition to the materials used in the training dataset, the testing dataset also contains spoof images fabricated using two new materials: liquid Ecoflex and RVT.

The LivDet 2015 dataset version is challenging due to the variance in the considered material and in the adopted sensors. Furthermore, the data was collected with two fabrication methods: (i) cooperative and (ii) non-cooperative. The cooperative fabrication process involves molding a live subject's finger, while the non-cooperative enhances latent prints on surfaces and prints them on a transparent sheet for mold creation.

4.1.2 ATVS

The ATVS dataset is a publicly available—accessible upon agreement—resource containing genuine and spoofed fingerprint images gathered by the Biometric Recognition Group at Universidad Autónoma de Madrid, Spain (Galbally et al., 2010). It consists of two subsets: cooperative and non-cooperative, each containing both genuine and spoofed fingerprint images in bitmap format.

In the cooperative dataset, spoof fingerprints were generated with user cooperation by capturing index and middle finger samples from both hands of 17 users using three different sensors: a flat optical sensor by Biometrika, a sweeping thermal sensor by Yubee with Atmel's FingerChip, and a flat capacitive sensor by Precise Biometrics (model Precise 100 SC). This subset comprises 816 genuine fingerprint images and an equal number of spoofed samples.

The non-cooperative dataset includes spoof fingerprints collected without user cooperation, with samples from 16 users. Each fingerprint (genuine and spoofed) was captured using the same three sensors as in the cooperative dataset during a single acquisition session. This subset comprises 768 genuine images and an equal number of spoofed samples.

4.2 Experiment 1: Fingerprint spoof attacks detection using ResNet

In the first experiment, four versions of the ResNet(He et al., 2016) model were considered: ResNet-18, 34, 50, and 101. These architectures differ in depth and complexity, with 18 being the most lightweight and 101 being the most complex. The performance of each variant can vary depending on the dataset characteristics and task-specific requirements. All models were trained using the training set of the LivDet 2015 dataset (Mura et al., 2015)and the ATVS dataset (Galbally et al., 2010). The models were assessed using TDR at FDR 5%, ACE, and F1-score. Moreover, we adopted three configuration scenarios per model for hyperparameter selection. Each configuration varied the learning rate, optimizer, and training duration (epochs). For example, we tested learning rates of 0.01, 0.001, and 0.0001, with optimizers such as SGD with momentum 0.9 and Adam. Also, we maintained a batch size of 32 across all configurations to ensure consistency. Tables 1–3 present the full hyperparameter configurations used for ResNet-34, ResNet-50, and ResNet-101 across both datasets.

Dataset	Configuration (C#)	Learning Rate	Optimizer	Momentum	Number of Batches	Number of Epochs
ATVS	C1	0.01	SGD	0.9	32	15
	C2	0.001	SGD	0.9	32	20
	C3	0.0001	SGD	0.9	32	32
LivDet 2015	C1	0.01	Adam	-	32	15
	C2	0.001	Adam	-	32	20
	C3	0.0001	Adam	-	32	32

TABLE 1: Hyperparameter configurations considered for training ResNet-34.

Dataset	Configuration (C#)	Learning Rate	Optimizer	Momentum	Number of Batches	Number of Epochs
ATVS	C1	0.01	SGD	0.9	32	32
	C2	0.001	SGD	0.9	32	22
	C3	0.0001	SGD	0.9	32	42
LivDet 2015	C1	0.01	SGD	0.9	32	35
	C2	0.001	SGD	0.9	32	29
	C3	0.0001	SGD	0.9	32	29

TABLE 2: Hyperparameter configurations considered for training ResNet-50.

Dataset	Configuration (C#)	Learning Rate	Optimizer	Momentum	Number of Batches	Number of Epochs
ATVS	C1	0.01	SGD	0.9	32	17
	C2	0.001	SGD	0.9	32	12
	C3	0.0001	SGD	0.9	32	27
	C1	0.01	SGD	0.9	32	19

LivDet 2015	C2	0.001	SGD	0.9	32	15
	C3	0.0001	SGD	0.9	32	20

TABLE 3: Hyperparameter configurations considered for training ResNet-101.

Considering the configuration that allowed achieving the best result for each model, Figure1 displays the performance results on the ATVS validation set with respect to ResNet-18, ResNet-34, ResNet-50, and ResNet-101. Similarly, Figure2 displays the performance results on the ATVS test set with respect to ResNet-18, ResNet-34, ResNet-50, and ResNet-101. As it can be seen, ResNet-34 achieved the best result.

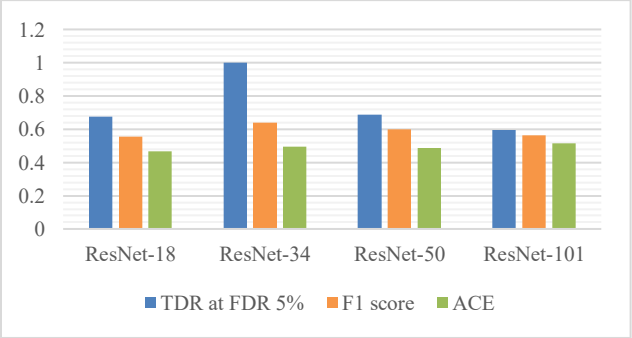

FIGURE 1: Performance results on the ATVS validation set with respect to ResNet-18, ResNet-34, ResNet-50, and ResNet-101.

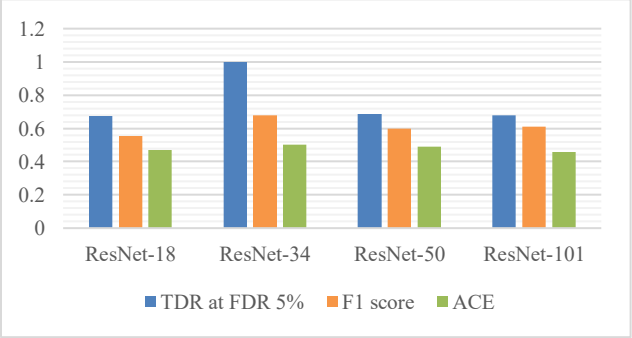

FIGURE 2: Performance results on the ATVS test set with respect to ResNet-18, ResNet-34, ResNet-50 and ResNet-101.

The results of the ATVS validation and test sets for the ResNet model that achieved the optimal results on the validation set (ResNet-34) are presented in Table 4. As it can be seen, there is a slight drop in the performance when using the test set. This shows that the model generalizes well.

	TDR at FDR 5%	F1-Score	ACE
Validation set	1.000	0.640	0.500
Test set	1.000	0.680	0.500

TABLE 4: Performance results on ATVS validation and test sets for the best ResNet architecture with respect to the best configuration 2.

In consideration of the configuration that permitted the optimal outcome for each model, Figure3 presents the performance outcomes on the LivDet 2015 validation set in relation to ResNet-18, ResNet-34, ResNet-50, and ResNet-101. Figure4 presents the performance results on the ATVS

test set with respect to ResNet-18, ResNet-34, ResNet-50, and ResNet-101. As can be observed, ResNet-34 achieved the optimal result.

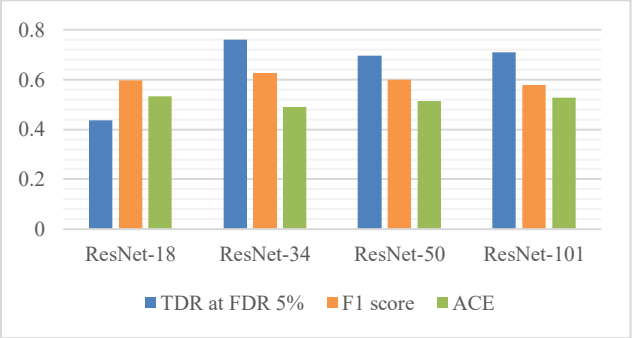

FIGURE 3: Performance results on the LivDet 2015 validation set with respect to ResNet-18, ResNet-34, ResNet-50, and ResNet-101.

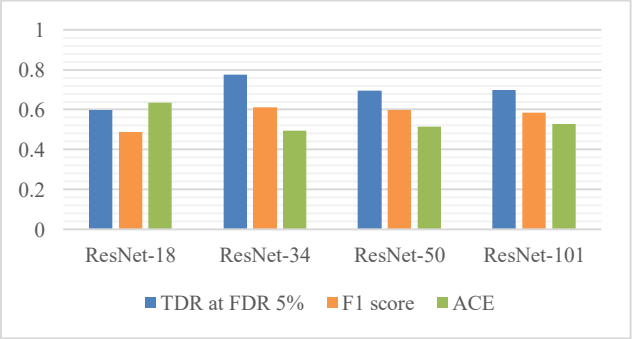

FIGURE 4: Performance results on the LivDet 2015 test set with respect to ResNet-18, ResNet-34, ResNet-50 and ResNet-101.

Table 5 shows the performance results on validation and test sets for the ResNet model that achieved the best results on the LivDet 2015 validation set (ResNet-34). As evident from the results, there is only a marginal decline in performance when transitioning from the validation set to the test set. This negligible drop suggests that the model's performance generalizes well beyond the training data, indicating that overfitting is unlikely to be a significant issue.

	TDR at FDR 5%	F1-Score	ACE
Validation set	0.760	0.630	0.490
Test set	0.770	0.610	0.490

TABLE 5: Performance results on LivDet2015 validation and test sets for the best ResNet architecture with respect to the best configuration 2.

The superior performance of ResNet-34 on the ATVS and LivDet 2015 datasets compared to ResNet-18, ResNet-50, and ResNet-101 may result from several factors. Firstly, the smaller model ResNet-34 is inherently less prone to overfitting and thus achieves better performance. In fact, larger models like ResNet-50 and ResNet-101 have more parameters, making them more prone to overfitting. ResNet-34, being smaller, may benefit from inherent regularization effects, helping it generalize better to unseen data. Furthermore, the feature representation learned by each model variant could play a pivotal role. It's possible that the feature representation learned by ResNet-34 is more discriminative or better aligned with the dataset's underlying patterns, leading to superior performance.

Although ResNet-34 demonstrated a high performance in fingerprint spoof detection, a subset of enrolled fingerprints resulted in misclassification. Figure 5 shows examples of misclassifications made by ResNet-34. In Figure 5(a), a genuine fingerprint image is incorrectly predicted as a spoof. This misclassification can be attributed to the poor quality of the image, evident in its blurred appearance. Alternatively, Figure 5(b) depicts a spoof image misclassified as genuine. One plausible explanation for this error could be the low resolution of the image and discontinuity.

(a) (b)

FIGURE 5: ResNet-34 misclassifications examples predicted (a) as spoof, ground truth is genuine, (b) as genuine, ground truth is spoof.

4.3 Experiment 2: Fingerprint spoof attacks detection using DenseNet

The primary objective of this experiment is to determine whether the DenseNet is able to categorize the presented fingerprint image as “Genuine” or “Spoof”. In particular, we try to find the best hyper-parameter configuration settings for DenseNet-121, DenseNet-169, DenseNet-161, and DenseNet-201 models when used to detect fingerprint spoofing attacks. The models considered in this experiment were trained on two datasets, LivDet 2015 and ATVS. The training data was obtained using a cross-validation data-splitting technique, where we split the data into 60% for training, 20% for validation, and 20% for testing. Moreover, we employed different hyperparameter settings per model. Each configuration involved varying key parameters such as the learning rate (0.01, 0.001, and 0.0001), the optimizer (SGD with momentum 0.9 or Adam), and the number of training epochs. To maintain consistency across experiments, the batch size was fixed at 32 for all configurations. Tables 6-9 depict the hyperparameter configurations considered for training DenseNet variants on the ATVS dataset and the LivDet 2015 dataset.

Dataset	Configuration (C#)	Learning Rate	Optimizer	Momentum	Number of Batches	Number of Epochs
ATVS	C1	0.01	SGD	0.9	32	40
	C2	0.001	SGD	0.9	32	31
	C3	0.0001	SGD	0.9	32	45
LivDet 2015	C1	0.01	SGD	0.9	32	48
	C2	0.001	SGD	0.9	32	29
	C3	0.0001	SGD	0.9	32	31

TABLE 6: Hyperparameter configurations considered for training DenseNet-121.

Dataset	Configuration (C#)	Learning Rate	Optimizer	Momentum	Number of Batches	Number of Epochs
ATVS	C1	0.01	Adam	-	32	20
	C2	0.001	Adam	-	32	25
	C3	0.0001	Adam	-	32	29
LivDet 2015	C1	0.01	Adam	-	32	27
	C2	0.001	Adam	-	32	21
	C3	0.0001	Adam	-	32	20

TABLE 7: Hyperparameter configurations considered for training DenseNet-169.

Dataset	Configuration (C#)	Learning Rate	Optimizer	Momentum	Number of Batches	Number of Epochs
ATVS	C1	0.01	SGD	0.9	32	20
	C2	0.001	SGD	0.9	32	23
	C3	0.0001	SGD	0.9	32	17
LivDet 2015	C1	0.01	Adam	-	32	30
	C2	0.001	Adam	-	32	23
	C3	0.0001	Adam	-	32	26

TABLE 8: Hyperparameter configurations considered for training DenseNet-161.

Dataset	Configuration (C#)	Learning Rate	Optimizer	Momentum	Number of Batches	Number of Epochs
ATVS	C1	0.01	SGD	0.9	32	34
	C2	0.001	SGD	0.9	32	10
	C3	0.0001	SGD	0.9	32	18
LivDet 2015	C1	0.01	Adam	-	32	33
	C2	0.001	SGD	0.9	32	30
	C3	0.0001	SGD	0.9	32	20

TABLE 9: Hyperparameter configurations considered for training DenseNet-201.

Considering the configuration that achieved the best result for each model, Figure6 displays the performance results on the ATVS validation set with respect to DenseNet-121, DenseNet-169, DenseNet-161, and DenseNet-201. Likewise, Figure7 displays the performance results on the ATVS test set with respect to DenseNet-121, DenseNet-169, DenseNet-161, and DenseNet-201. As it can be seen, DenseNet-169 achieved the best result.

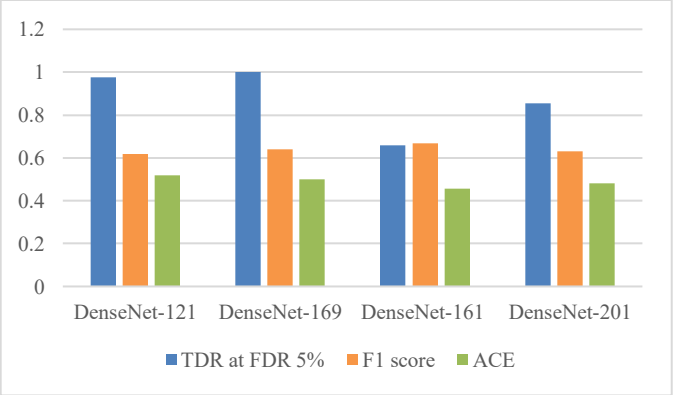

FIGURE 6: DenseNet performance results on the ATVS validation set.

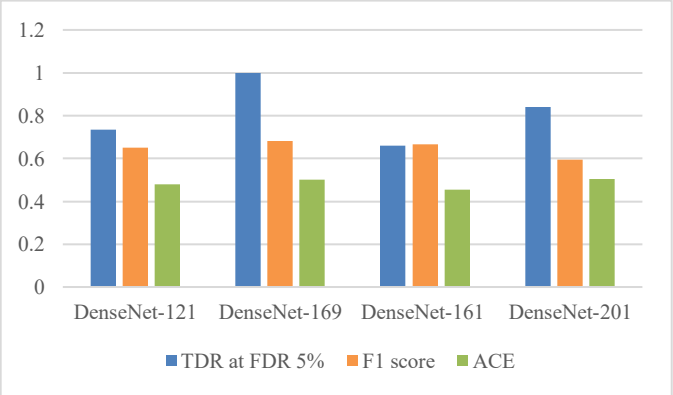

FIGURE 7: DenseNet performance results on the ATVS test set.

The results of the ATVS validation and test sets for the DenseNet model that achieved the optimal results on the validation set (DenseNet-169) are presented in Table 10.

	TDR at FDR 5%	F1-Score	ACE
Validation set	1.000	0.639	0.500
Test set	1.000	0.682	0.500

TABLE 10: Performance results on LivDet2015 validation and test sets for the best DenseNet-169 architecture with respect to the best configuration.

DenseNet-169 stood out as the best choice among other variants, DenseNet-161, DenseNet-201, and DenseNet-121 for the ATVS dataset, as shown in Figure8, due to its balanced complexity and performance. While DenseNet-201 is deeper, it might risk overfitting on smaller datasets. DenseNet-161 and DenseNet-121, though efficient, might not capture all nuances of the data. DenseNet-169 achieves an optimal balance, offering both complexity and efficiency. It showed superior performance during testing, making it the preferred choice.

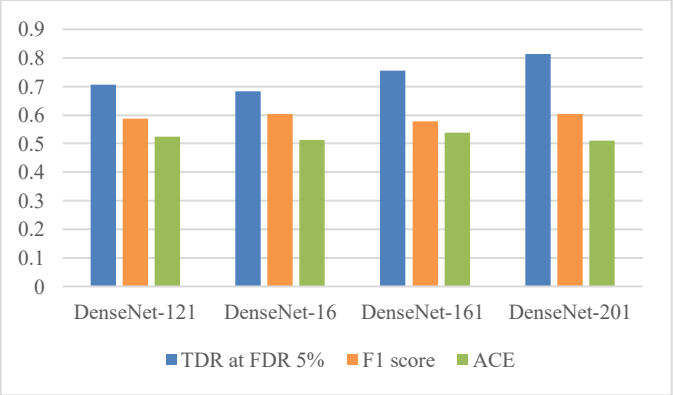

FIGURE 8: DenseNet performance results on the LivDet 2015 validation set.

The results of the LivDet 2015 validation and test sets for the DenseNet model that achieved the optimal results on the validation set (DenseNet-201) are presented in Table 11.

	TDR at FDR 5%	F1-Score	ACE
Validation set	0.813	0.604	0.510
Test set	0.839	0.595	0.504

TABLE 11: Performance results on LivDet2015 validation and test sets for the best DenseNet-201 architecture with respect to the best configuration.

For the LivDet 2015 dataset, DenseNet-201 stood out as the best choice. Its deeper architecture likely enabled it to capture a wider range of features effectively, leading to superior performance on both validation and test sets. The dataset's complexity benefited from DenseNet-201's increased depth, allowing it to extract intricate patterns and achieve higher detection rates. Conversely, in datasets with different characteristics, such as ATVS, the deeper architecture might not yield the same benefits, potentially leading to overfitting. Thus, DenseNet-201's selection underscores the importance of considering dataset-specific factors and architecture when choosing a model.

Although DenseNet-201 demonstrated good results on the LivDet 2015 dataset, it misclassified some of the enrolled fingerprint examples. As shown in Figure 9(a), DenseNet-201 incorrectly classified the fingerprint as a spoof due to the low contrast and discontinuity of the image. Conversely, Figure 9(b) shows a misclassification as genuine, which can be attributed to the high contrast in an image or to the binarization process adopted.

(a) (b)

FIGURE 9: DenseNet-201 misclassification examples on LivDet2015 dataset, (a) is a Genuine sample predicted as Spoof (b) is a Spoof sample predicted as Genuine.

4.4 Experiment 3: Fingerprint spoof attacks detection using Inception v3

In the third experiment, the objective of this experiment is to evaluate the Inception v3 model's ability to classify fingerprint images as "Genuine" or "Spoof." This experiment aims to identify the optimal hyperparameter settings for Inception v3 when used to detect fingerprint spoofing attacks. The training and evaluation were conducted on the same datasets as in Experiment 3, LivDet 2015 dataset (Mura et al., 2015) and ATVS dataset (Galbally et al., 2010). Moreover, we adopted three configuration scenarios for hyperparameter tuning. These configurations varied in learning rates (0.01, 0.001, and 0.0001), used the SGD optimizer with a momentum of 0.9, and tested different training durations (epochs). A batch size of 32 was consistently maintained across all configurations to ensure a fair comparison. Table 12 displays the hyperparameter settings used for Inception v3 on the ATVS dataset and the LivDet 2015 dataset.

Dataset	Configuration (C#)	Learning Rate	Optimizer	Momentum	Number of Batches	Number of Epochs
ATVS	C1	0.01	SGD	0.9	32	15
	C2	0.001	SGD	0.9	32	21
	C3	0.0001	SGD	0.9	32	33
LivDet 2015	C1	0.01	SGD	0.9	32	33
	C2	0.001	SGD	0.9	32	20
	C3	0.0001	SGD	0.9	32	28

TABLE 12: Hyperparameter configurations considered for training Inception-v3.

Considering the configuration that allowed achieving the best result for each model, the results of the ATVS validation and test sets for the Inception v3 model that achieved the optimal results on the validation set are presented in Table 13. Similarly, the results related to the LivDet 2015 dataset are reported in Table 14.

	TDR at FDR 5%	F1-Score	ACE
Validation set	0.633	0.597	0.461
Test set	0.661	0.595	0.464

TABLE 13: Performance results on ATVS validation and test sets for the best Inception-v3 architecture with respect to the best configuration 3.

	TDR at FDR 5%	F1-Score	ACE
Validation set	0.779	0.612	0.512
Test set	0.755	0.601	0.510

TABLE 14: Performance results on LivDet2015 validation and test sets for the best Inception-v3 architecture with respect to the best configuration 1.

Figure 10 presents instances of misclassifications observed in Inception v3. In Figure 10 (a), both Configuration 1 and Configuration 3 yielded the misclassification of a genuine fingerprint image as a spoof. This misclassification occurs due to unclear edges within the fingerprint. In Figure 10 (b), Configuration 3 and Configuration 2 incorrectly identify a fake image as genuine, possibly due to motion blur present in the image. Lastly, in Figure 10 (c), Configuration 2 allowed misclassifying a genuine image as a spoof, likely caused by variations in the pressure applied during fingerprint scanning, resulting in inconsistencies in the captured image.

(a) (b) (c)

FIGURE 10: Inception v3 misclassifications examples predicted (a) as spoof when using Configuration 1 and Configuration 3, (b) as genuine when using Configuration 3 and 2, and (c) as spoof when using Configuration 2.

4.5 Experiment 4: Fingerprint spoof attacks detection using ViT

This experiment aims to evaluate the effectiveness of the vision transformer models ViT16 and ViT32 in categorizing fingerprint images as "Genuine" or "Spoof". Similar to the previous experiments, this experiment aims to determine the most effective hyperparameter configurations for the ViT models in detecting fingerprint spoofing attacks. Training and evaluation are performed on the LivDet 2015 and ATVS datasets, maintaining the same split of 60% training, 20% validation, and 20% test data. Consistency is ensured by using a batch size of 32 and a training duration of 1000 epochs with early stopping to avoid overfitting. Across three configurations, we varied the learning rates (0.01, 0.001, and 0.0001) while employing the SGD optimizer with a momentum of 0.9. Performance is evaluated using identical metrics, including ACE, TDR @ FDR 5 %, and F1-Score to ensure direct comparison with previous experiments. Table 15 and Table 16 report the different hyperparameter settings used to train ViT-16 and ViT-32 using the ATVS dataset and the LivDet2015 dataset.

Dataset	Configuration (C#)	Learning Rate	Optimizer	Momentum	Number of Batches	Number of Epochs
ATVS	C1	0.01	SGD	0.9	32	86
	C2	0.001	SGD	0.9	32	25
	C3	0.0001	SGD	0.9	32	23
LivDet 2015	C1	0.01	SGD	0.9	32	34
	C2	0.001	SGD	0.9	32	41
	C3	0.0001	SGD	0.9	32	40

TABLE 15: Hyperparameter configurations considered for training ViT-16.

Dataset	Configuration (C#)	Learning Rate	Optimizer	Momentum	Number of Batches	Number of Epochs
ATVS	C1	0.01	SGD	0.9	32	20
	C2	0.001	SGD	0.9	32	11
	C3	0.0001	SGD	0.9	32	17
	C1	0.01	SGD	0.9	32	42

LivDet 2015	C2	0.001	SGD	0.9	32	14
	C3	0.0001	SGD	0.9	32	27

TABLE 16: Hyperparameter configurations considered for training ViT-32.

Considering the configuration that achieved the best result for each model, Figure11 shows the performance results on the ATVS validation set with respect to ViT-16 and ViT-32. Similarly, Figure12 shows the performance results for the test set with respect to ViT-16 and ViT-32.

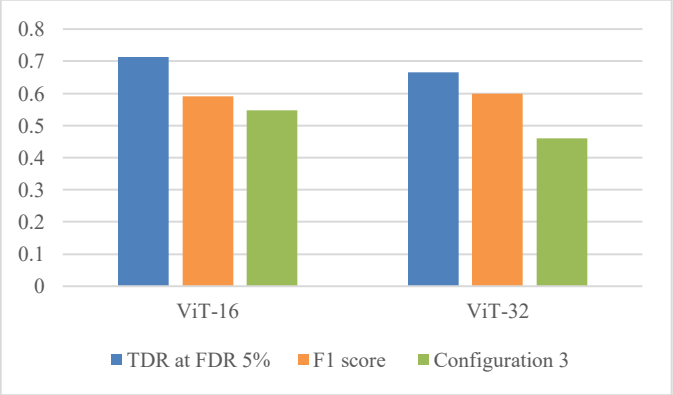

FIGURE 11: Performance results on the ATVS validation set with respect to ViT-16 and ViT-32.

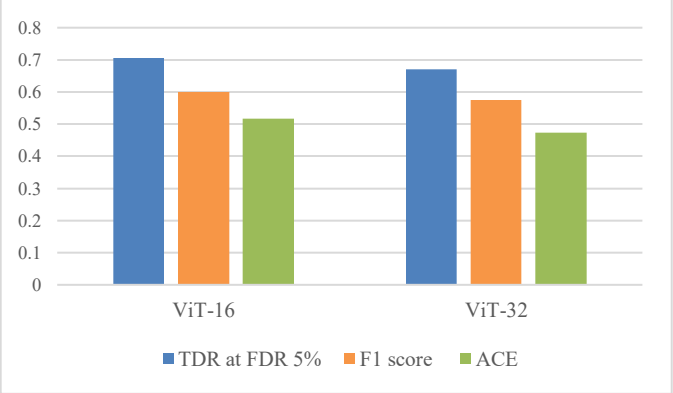

FIGURE 12: Performance results on the ATVS test set with respect to ViT-16 and ViT-32.

The results of the ViT model in the ATVS validation and test sets are shown in Table 17 (ViT-16), which achieved the best results in the validation set.

	TDR at FDR 5%	F1-Score	ACE
Validation set	0.779	0.612	0.512
Test set	0.755	0.601	0.510

TABLE 17: Performance results on ATVS validation and test sets for the best ViT-16 architecture with respect to the best configuration.

Among the different ViT variants evaluated for the ATVS dataset, ViT-16 proved to be the preferred choice due to its balanced mix of complexity and performance, outperforming ViT-32. While ViT-32 offers a larger architecture with more parameters and computational requirements, it also increases the risk of overfitting on smaller datasets. In contrast, ViT-16, with its smaller

size, is more efficient and less prone to overfitting. During testing, ViT-16 demonstrated superior performance, supporting its position as the optimal choice over ViT-32.

Considering the configurations that lead to the best results for each model, Figure13 illustrates the performance results on the LivDet 2015 validation set with respect to ViT-16 and ViT-32. Similarly, Figure14 shows the performance results on the LivDet 2015 test set with respect to ViT-16 and ViT-32. Notably, ViT-16 emerges as the top performer and achieves the optimal result.

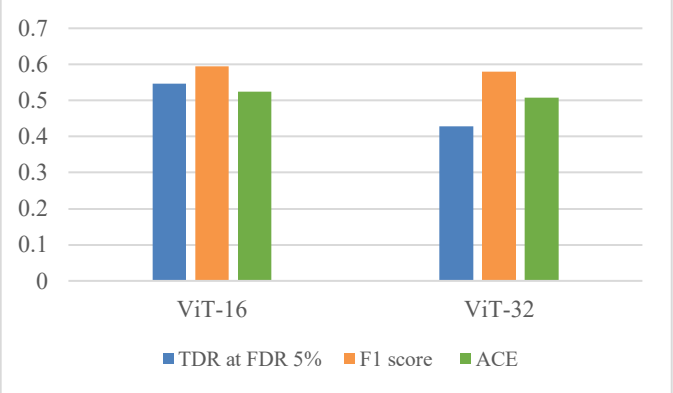

FIGURE 13: Performance results on the LivDet 2015 validation set with respect to ViT-16 and ViT-32.

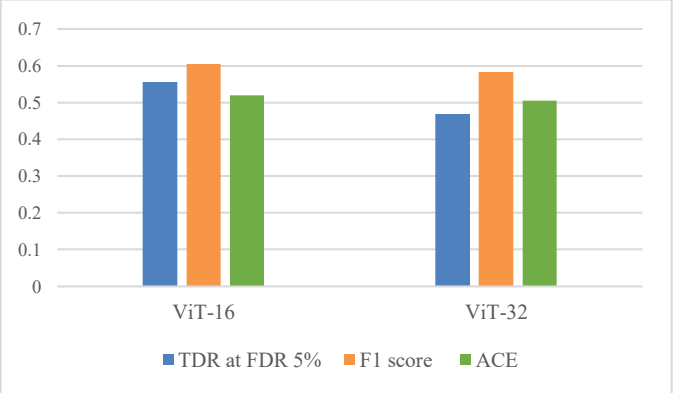

FIGURE 14: Performance results on the LivDet 2015 test set with respect to ViT-16 and ViT-32.

Table18showstheperformanceresults of the ViT-16 model in the validation and test sets. The ViT-16 model outperformed ViT-32 in the LivDet2015 validation set. The ViT-16 model shows consistent performance between the validation and test sets. Despite a slight improvement in the test set, this indicates a robust generalization of the model beyond the training data.

TABLE 18: Performance results on LivDet2015 validation and test sets for the best ViT-16 architecture with respect to the best configuration.

	TDR at FDR 5%	F1-Score	ACE
Validation set	0.546	0.594	0.524
Test set	0.555	0.604	0.520

Although ViT-16 showed strong performance in recognizing fingerprints, a subset of the fingerprints captured were misclassified. Figure 15 illustrates examples of misclassifications by ViT-16. In Figure15 (a), a spoof fingerprint image is incorrectly identified as genuine. This misclassification could be due to the low resolution and discontinuity of the image. Conversely, Figure15 (b) shows a genuine image that has been misclassified as a spoof. The reason for this

misclassification could be the blurring of details in dark color, resulting in poor image quality.

(a) (b)

FIGURE 15: ViT misclassification examples predicted (a) as genuine, ground truth is spoof, (b) as spoof, ground truth is genuine.

4.6 Performance Comparison

The best models were evaluated using TDR @ FDR = 0.05, F1 score, and ACE measures, with computational efficiency assessed by the number of FLOPs. Figure16 and Figure17 illustrate that ResNet34 and DenseNet169 outperformed the other models across both datasets. Conversely, InceptionV3 demonstrated poor results on the ATVS dataset but achieved performance comparable to ResNet34 on LivDet2015. In contrast, ViT underperformed on both datasets. A plausible explanation is that ViT's transformer architecture, while powerful for capturing long-range dependencies, is less efficient at extracting fine-grained image details. This makes ResNet34 and DenseNet169 more suitable for tasks like fingerprint spoof detection, where detailed feature extraction is crucial.

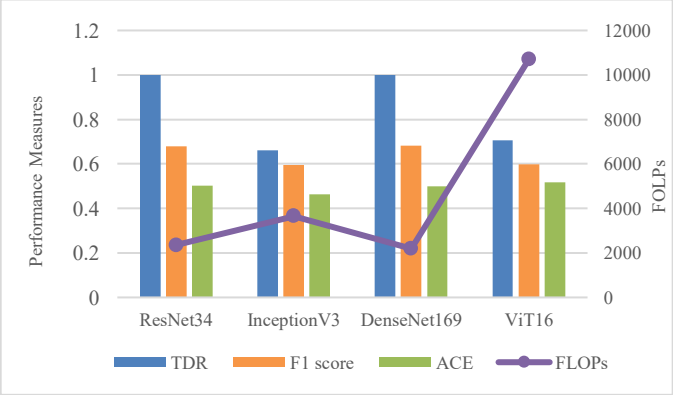

FIGURE 16: Performance comparison of InceptionV3, ResNet34, DenseNet169, and ViT in terms of TDR@FDR 0.05, F1-score, ACE, and FLOPs on the ATVS dataset.

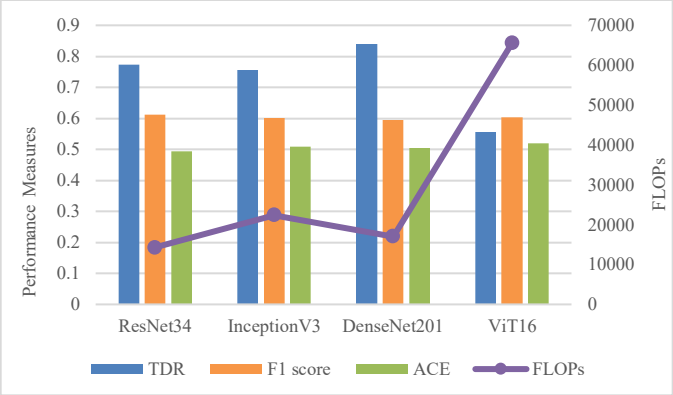

FIGURE 17: Performance comparison of InceptionV3, ResNet34, DenseNet201, and ViT in terms of TDR@FDR 0.05, F1-score, ACE, and FLOPs on the LivDet 2015 dataset.

Moreover, the transformer architecture incorporates image processing patches and attention mechanisms, which contribute to a higher number of parameters. Additionally, its multiple self-attention mechanisms and feed-forward layers involve extensive computations compared to convolutional neural networks (CNNs). As a result, the time complexity and parameter count of ViT are significantly higher than those of DenseNet and ResNet, as illustrated in Figure16 and Figure17. Consequently, when evaluating both performance and computational efficiency, ResNet and DenseNet emerge as the optimal choices for detecting fingerprint spoof attacks in the ATVS and LivDet 2015 datasets.

In contrast to DenseNet and ResNet, InceptionV3 demonstrated poorer performance due to its complex architecture. Its reliance on factorized convolutions and aggressive dimensionality reduction can result in the loss of critical features necessary for tasks such as fingerprint spoof detection. Additionally, InceptionV3 struggles with the vanishing gradient problem in its deeper layers, as it lacks the residual connections present in ResNet and DenseNet, which prevent vanishing and exploding gradients. Residual connections ensure stable and effective learning, especially in deep networks, leading to faster convergence and more effective parameter updates. DenseNet's dense connections further enhance performance and training efficiency by promoting feature reuse and ensuring robust gradient flow throughout the network, thereby enhancing the model's ability to learn complex patterns.

5 CONCLUSION AND FUTURE WORKS

Recent advancements in artificial intelligence and deep learning boosted the performance of computer vision systems to solve real-world problems such as spoof fingerprint detection. As such, this paper proposed to investigate recent deep learning paradigms and figure out the most suitable one for presentation attack detection. For this purpose, different models and their variants are trained and then tested. These experimental results underscore that ResNet34 and DenseNet169 consistently outperformed other models on both the ATVS and LivDet2015 datasets, achieving superior metrics in TDR @ FDR 0.05, F1 score, and ACE while maintaining lower time complexity. Conversely, InceptionV3 displayed subpar performance on the ATVS dataset but performed comparably to ResNet34 on LivDet2015. ViT 16 and ViT 32, despite their sophisticated transformer architecture, exhibited the highest time complexity and underperformed across both datasets. These findings highlight the efficiency and effectiveness of ResNet and DenseNet, making them the most suitable choices for fingerprint spoof detection due to their robust performance and computational efficiency. Notably, no advanced preprocessing or data augmentation techniques were applied to the training data, allowing for an unbiased evaluation of each model's inherent feature extraction capability in real-world conditions. This study addresses a notable gap in the literature by offering a direct and systematic comparison between CNN-based models and Vision Transformers for fingerprint presentation attack detection — an area where limited research has been conducted. Our work not only evaluates ViT's feasibility for this

task but also highlights the importance of considering architectural suitability based on biometric-specific characteristics, such as ridge structure and texture continuity.

Future work will focus on further optimizing the computational efficiency and performance of deep learning models for fingerprint spoof detection. This includes exploring hybrid architectures that combine the strengths of CNNs and transformers to potentially improve both accuracy and speed. Additionally, we suggest investigating advanced techniques for data augmentation and preprocessing to enhance model robustness against diverse spoofing methods. Another area of interest is the development of lightweight models that can be deployed on resource-constrained devices without significant loss of accuracy. Furthermore, exploring the vulnerability of these models to adversarial attacks and proposing mitigation strategies will be essential to ensure secure deployment in real-world scenarios. Finally, a comprehensive analysis of the model's performance on a wider range of datasets can be conducted to ensure their generalizability and effectiveness in real-world applications.

6 REFERENCES

Ali, M. M., Mahale, V. H., Yannawar, P., & Gaikwad, A. T. (2016, February). Fingerprint recognition for person identification and verification based on minutiae matching. *IEEE 6th International Conference on Advanced Computing (IACC)*, (pp. 332–339).

Bárta, B. M. (2016). Generation of skin disease into the synthetic fingerprints. *International Journal of Image Processing (IJIP)*, *CSC Journals*, *10*(3), 229–248.

Brereton, R. G. (2010). Support vector machines for classification and regression. *Analyst*, *135*(2), 230–267.

Carta, S., Casula, R., Orrù, G., Micheletto, M., &Marcialis, G. L. (2025). Interpretability of fingerprint presentation attack detection systems: a look at the “representativeness” of samples against never-seen-before attacks. *Machine Vision and Applications*, *36*(2), 44.

Chen, X., Hsieh, C.-J., & Gong, B. (2021). When vision transformers outperform ResNets without pre-training or strong data augmentations. *arXiv preprint arXiv:2106.01548*.

Chugh, T., Kai, C., & Jain, A. K. (2017). Fingerprint spoof detection using minutiae-based local patches. *IEEE International Joint Conference on Biometrics (IJCB)*,(pp. 581–589).

CN, D., Sankar, S. P., & George, N. (2013). Multimodal identification system in monozygotic twins.*International Journal of Image Processing (IJIP)*, *7*(1), 72–84.

Djara, T., Assogba, M. K., Naït-Ali, A., &Vianou, A. (2015). Fingerprint registration using Zernike moments: An approach for a supervised contactless biometric system. *International Journal of Image Processing (IJIP)*, *9*(5), 254–264.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... &Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Galbally, J., Fierrez, J., Alonso-Fernandez, F., & Martinez-Diaz, M. (2010). Evaluation of direct attacks to fingerprint verification systems. *Telecommunication Systems*, *47*(3–4), 243–254.

Ghiani, L., Yambay, D., Mura, V., Tocco, S., Marcialis, G. L., Roli, F., & Schuckers, S.(2013). LivDet 2013—Fingerprint liveness detection competition 2013. *6th IAPRIEEE International Conference on Biometrics*, (pp. 1–6).

Ghojogh, B., & Ghodsi, A. (2023). Recurrent neural networks and long short-term memory networks: Tutorial andsurvey. *arXiv preprint arXiv:2304.11461*.

Grosz, S. A., Wijewardena, K. P., & Jain, A. K. (2023, September). ViT unified: Joint fingerprint recognition and presentation attack detection. *IEEE International Joint Conference on Biometrics (IJCB),* (pp. 1-9). IEEE.

Guennouni, S., Mansouri, A., &Ahaitouf, A. (2020). Biometric systems and their applications. Visual Impairment and Blindness - What We Know and What We Have to Know. *IntechOpen*.

He, K., Zhang, X., Ren, S., & Sun, J.(2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition,* (pp. 770-778).

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q.(2017). Densely connected convolutional networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 2261–2269).

Jian, W., Zhou, Y., & Liu, H. (2021). Densely connected convolutional network optimized by genetic algorithm for fingerprint liveness detection. *IEEE Access*, *9*, 2229–2243.

Kothadiya, D., Shah, D., Patel, M., & Pujara, V. (2023). Enhancing fingerprint liveness detection accuracy using deep learning: A comprehensive study and novel approach. *Journal of Imaging*, *9*(8), 158.

Lee, Y. K., Jeong, J., & Kang, D. (2022). An effective orchestration for fingerprint presentation attack detection. *Electronics*, *11*(16), 2515.

Liu, Y., Huang, A., Yang, Y., Xiao, B., Yuan, L., Zhang, L.,.& Chen, D. (2022). A survey of visual transformers. *IEEE Transactions on Neural Networks and Learning Systems*.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., ... & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, (pp. 10012–10022).

Marasco, E., Wild, P., &Cukic, B. (2016). Robust and interoperable fingerprint spoof detection via convolutional neural networks. *IEEE Symposium on Technologies for Homeland Security (HST)*, 1–6.

Marcialis, G. L., Lewicke, A., Tan, B., Coll, P., Roll, F., Grimberg, D., ... & Schuckers, S. (2009). First international fingerprint liveness detection competition (LivDet 2009). *Proceedings of ICIAP*.

Maurício, J., Domingues, I., & Bernardino, J. (2023). Comparing vision transformers and convolutional neural networks for image classification: A literature review. *Applied Sciences*, *13*(9), 5521.

Mura, V. (2018). LIVDET 2017 fingerprint liveness detection competition 2017. *arXiv*.

Mura, V., Ghiani, L., Marcialis, G. L., Roli, F., Yambay, D. A., & Schuckers, S. A. (2015). LivDet 2015—Fingerprint liveness detection competition 2015. IEEE 7th International Conference on Biometrics Theory, *Applications and Systems (BTAS)*, (pp. 1–6).

Rai, A., Dey, S., Patidar, P., & Rai, P. (2023). EXPRESSNET: An explainable residual slim network for fingerprint presentation attack detection. *ResearchGate*.

Ramaneswaran, S., Srinivasan, K., Vincent, P., & Chang, C. (2021). Hybrid inception v3 XGBoost model for acute lymphoblastic leukemia classification. *Computational and Mathematical Methods in Medicine*, (pp. 1–10).

Reza, N., & Jung, H. Y. (2023). Enhancing Ensemble Learning Using Explainable CNN for Spoof Fingerprints. *Sensors (Basel, Switzerland)*, *24*(1), 187.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Fei-Fei, L. (2015). ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, *115*(3), 211–252.

Sandouka, A. S. B., Bazi, Y., & Rahhal, M. M. A. (2020). EfficientNet combined with generative adversarial networks for presentation attack detection. International Conference on Artificial Intelligence & Modern Assistive Technology (ICAIMAT), (pp. 1–5).

Sharma, K., Raghuwanshi, A., & Sharma, V. K. (2015). Biometric system—a review. *International Journal of Computer Science and Information Technologies (IJCSIT)*, 6(5), 4616–4619.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A.(2015). Going deeper with convolutions. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 1–9).

Tabassi, E., Tarang, C., Deb, D., & Jain, A. K. (2018). Altered fingerprints: Detection and localization. *IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, (pp. 1–9).

Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., & Jégou, H. (2020). Training data-efficient image transformers & distillation through attention. *Proceedings of the 37th International Conference on Machine Learning (ICML)*, (pp. 10347–10357).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., &Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems (NIPS)*, 30, 5998–6008.

Vurity, A., Marasco, E., Ramachandra, R., & Park, J. (2025). ColFigPhotoAttnNet: Reliable Finger Photo Presentation Attack Detection Leveraging Window-Attention on Color Spaces. *arXiv preprint arXiv:2503.05247*.

Wen, J., Yujie, Z., &Hongming, L. (2020). Densely connected convolutional network optimized by genetic algorithm for fingerprint liveness detection. *IEEE Access*, 9, 2229–2243.

Yambay, D., Luca, G., Paolo, D., Gian, M. L., Fabio, R., & Schuckers, S. (2012). LivDet 2011—Fingerprint liveness detection competition 2011. *5th IAPR International Conference on Biometrics (ICB)*, (pp. 208–215).

Yuan, C., Li, X., Jonathan, W., Li, J., & Sun, X. (2017). Fingerprint liveness detection from different fingerprint materials using convolutional neural network and principal component analysis. *Computers, Materials & Continua*, *53*(3), 357–371.

Yuan, C., Xu, Z., Li, X., Zhou, Z., Huang, J., & Guo, P.(2024). An Interpretable Siamese Attention ResCNN for Fingerprint Spoofing Detection. *IETBiometrics*, 1, 6630173.

Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z., ... & Yan, S. (2021). Tokens-to-token ViT: Training vision transformers from scratch on ImageNet. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, (pp. 558–567).

Zhang, K., Huang, S., & Zhao, H. (2023). LFLDNet: Lightweight fingerprint liveness detection based on ResNet and Transformer. *Sensors*, *23*(15), 6854.

Zhang, Y., Liu, Y., Zhan, X., Cao, D., & Li, Z. (2020). FLDNet: Light dense CNN for fingerprint liveness detection. *IEEE Access*, 8, 84141–84152.

Zhang, Y., Shi, D., Zhan, X., Cao, D., Zhu, K., & Li, Z. (2019). Slim-ResCNN: A deep residual convolutional neural network for fingerprint liveness detection. *IEEE Access*, 7, 45390–45400.

Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. *IEEE International Conference on Computer Vision (ICCV)*, (pp. 2242–2251).