Fingerprint Anti-spoofing Analysis: from Minutiae to Transformers

Tushar Goyal, Harshit Shakya, Aditya Khandelwal, Himanshu Patidar, and Tushar Sandhan

Perception and Intelligence Lab, Dept. of EE, IIT Kanpur, India
tushg@iitk.ac.in, harshit21@iitk.ac.in, adityak21@iitk.ac.in, patidar21@iitk.ac.in, sandhan@iitk.ac.in

Abstract. In the contemporary digital and interconnected world, the need for robust security measures is paramount. From personal devices to national security, biometric recognition has enhanced security, streamlined authentication processes, and improved overall efficiency. Fingerprint liveness detection plays a vital role in ensuring the security and reliability of biometric systems by differentiating between genuine fingerprints and presentation attacks. This work presents a detailed analysis on various computational algorithms for fingerprint liveness detection along with cross-sensor data evaluation. The study investigates the effectiveness of different feature extraction modules, including local binary pattern (LBP), histogram of oriented gradients (HOG), scale-invariant feature transform (SIFT), minutiae points, and deep feature extractors, including various convolutional neural architectures and transformers. Additionally, deep learning patch-based classification approaches with multiple weight initialization strategies and transfer learning from large-scale datasets are explored for liveness detection. The overall framework is achieving an outstanding average classification error (ACE) value of less than 0.5%. Moreover, the comparison of different architectures, evaluation on various large datasets, model complexity versus anti-spoofing robustness analysis and cross-sensor evaluations offer useful insights about fingerprint liveness detection.

Keywords: Fingerprint spoof detection · convolutional neural network · liveness detection · presentation attack detection.

1 Introduction

Automatic access to services has become increasingly vital in this information age, leading to the development of biometric authentication [15]. Due to its uniqueness, cost-effectiveness, convenience, and a relatively high degree of accuracy, fingerprint recognition has been widely adopted in prevalent biometric technologies in various applications. And its presence is expected to continue growing with the recent push by technology companies.

However, the vulnerability of fingerprint-based systems to presentation attacks or spoofing and the widespread availability and ease of manufacturing

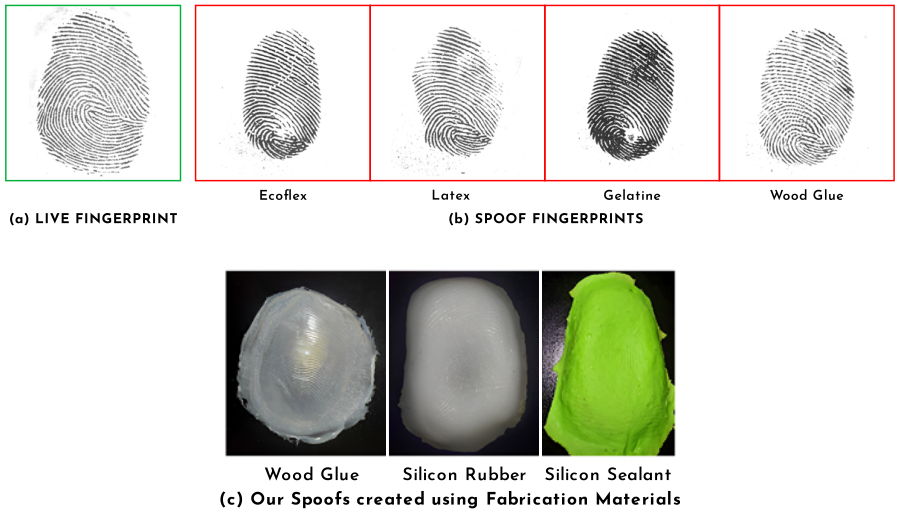

Fig. 1: (a) Live fingerprint vs (b) Spoof fingerprints made from Ecoflex, Latex, Gelatine and Wood Glue from LiveDet 2015 GreenBit sensor along with (c) Our spoofs created using various fabrication materials like wood glue, silicon rubber and silicon sealant.

presentation attack instruments (PAI) has emerged as a pressing security concern [25]. A presentation attack (PA), as defined by the ISO standard IEC 30107-1:2016(E), is a "presentation to the biometric data capture subsystem to interfere with the operation of the biometric system" [35]. Artificially created fingerprint patterns, either by replication via mould and cast operations or by direct manufacturing via methods like 3D printing and surface etching, are called spoofed fingerprints. In the case of fingerprint authentication systems, these attacks typically involve presenting a spoofed fingerprint to the biometric device, mimicking the fingerprint of an authorized user. Deceiving a fingerprint biometric device requires minimal skill and resources. Common materials such as wood glue, gelatin, clay, and playdough (as shown in Fig. 1) can be easily used for making spoof fingerprints, with the required process readily available on the internet and accessible at everyone's fingertips.

The goal of presentation attack detection (PAD) is to identify and detect fraudulent attempts to deceive biometric systems through the presentation of spoofed fingerprints. Liveness detection is a subset of PAD focusing on determining the liveness of the biometric attribute captured, i.e., whether the biometric attribute presented is genuine and not a spoof or false representation.

Subsequently, the structure of the paper will be as follows: section 2 presents a literature review, discussing previous studies and state-of-the-art methods related to fingerprint liveness detection; section 3 describes the methods and techniques employed for the detailed problem investigation; and section 4 presents the experimental results and our findings. Finally, section 5 concludes the paper by summarizing the important insights.

2 Literature Review

Methods for detecting the fake fingerprints can be divided into hardware and software-based categories. Hardware-based approaches require the integration of additional sensors to assess characteristics like pulse oximetry [36], blood pressure [17], skin distortion [8], or other biometric traits [40] associated with fingerprints, such as odour [10]. Drahansky et al. [17] employed blood pressure to differentiate genuine and synthetic fingerprints. Blood pressure values outside the normal ranges indicated a spoof. However, their limitation arose from counterfeit prints with normal blood pressure, allowing potential exploitation by hypertensive individuals. Baldisserra et al. [10] investigated odour as a discriminator between real skin and synthetic materials (e.g., silicone, latex). Voltage shifts were observed by sensors, with actual skin and gelatin exhibiting a reduction and synthetic materials producing an increase in the sensor voltage. However, gelatin-based artificial fingers that imitated real finger sensor responses constrained the biometric system's accuracy.

Gomez-Barrero et al. [21] utilized laser speckle contrast imaging (LSCI) and short wave infrared (SWIR) spectrum images to analyze finger surface and interior structure. By combining features and classifiers, they achieved a notable BPCER (rate of misclassification of live fingerprints) of 0.1% and APCER (rate of misclassification of spoof fingerprints) of 3% on a dataset including unknown PA. This multi-modal approach shows potential for boosting the security of fingerprint recognition. Biometric systems can categorize fingerprints based on a variety of other physical characteristics. Hardware techniques are effective yet difficult, expensive, and vulnerable to presentation attacks. Hence to tackle these challenges effectively, software-based approaches are used.

Software-based PAD methods extract features from fingerprint samples captured by primary sensors, leveraging the inherent uniqueness of live fingerprints to detect presentation attacks. They are classified as dynamic or static methods. Dynamic methods rely on traits like skin deformation [8], perspiration [5] or sweat pores [31]. Due to temporal variations, dynamic methods require multiple images or video frames. Johnson and Schuckers et al. [28] conducted a study investigating the use of pores for liveness detection. Their research examined the feasibility of differentiating between real and counterfeit fingerprints by analyzing variations in skin pore properties. Antonelli et al. [8] contributed significantly in identifying synthetic fingerprints by analyzing skin distortions. Through careful dataset analysis, they achieved an accuracy of 98.5% and an equal error rate (EER) of 11.24%, using human skin's elasticity and deformation characteristics.

Static methods directly extract PAD features from fingerprint samples. They do not rely on dynamic behaviour but rather focus on the inherent properties of the fingerprint image. These properties include the ridge pattern [44], texture [19], quality [18], and other inherent features of the fingerprint image. Tan et al. [44], proposed a PAD method evaluating noise in fingerprint ridge-valley patterns. The technique achieved a high classification percentage (90.9% to 100%) on diverse genuine and artificial fingerprint datasets. Ghiani et al. (2013) [19]

used binarized statistical image features (BISFs) to utilize texture features from fingerprint images.

The progress in deep learning techniques, specifically in computer vision, has led to the emergence of robust architectures for detecting presentation attacks in fingerprints [32,38]. These approaches have demonstrated significant advancements compared to earlier techniques. Nogueira et al. [34] conducted a comprehensive evaluation of three established convolutional neural networks (CNN) for fingerprint presentation attack detection. Their proposed CNN model achieved the highest accuracy in the LivDet 2015 [33] competition, achieving an overall accuracy of 95.5%. However, a notable limitation of these methods lies in their reliance on learning specific features from the entire fingerprint image, assuming a fixed size. This approach faces challenges even when dealing with similar datasets like that of LivDet, but the region of interest (ROI) covers only a tiny portion of the entire image. As a result, the effectiveness of presentation attack detection is compromised.

DeFraudNet, developed by Anusha et al. [9], is a highly effective network for fingerprint spoof detection. It leverages LBP and Gabor filters for image preprocessing and extracting textural features. The network employs two DenseNets [27] for the simultaneous whole image and patch-based feature extraction. By combining patch features with the entire image and utilizing attention mechanism, they achieved accuracy as high as 99.77% when trained and tested on Crossmatch 2013 [20] dataset surpassing [34] method, which is also based on CNN.

Tarang et al. [13] introduced a formalized approach comprising offline training and an online testing stage for fingerprint spoof detection. In the offline training stage, minutiae detection was performed on the sensed fingerprint image, followed by the extraction of localized patches centred and aligned based on the minutiae location and orientation. A MobileNet model [26] was subsequently trained on these aligned patches to obtain a refined representation of the fingerprint image. The evaluation employed a spoofness score, wherein the individual scores corresponding to the local patches were averaged to derive a global score. On the LivDet dataset [33], the strategy produced great results that were superior to those of the best practices.

Thus it is evident from state-of-the-art approaches that various feature based as well as CNN based learned feature representations are capable of preventing presentation attacks. In the following subsequent sections, we give a detailed analysis incorporating feature engineering, CNN as well as transformer based feature representations for fingerprint anti-spoofing.

3 Methods

3.1 Feature Engineering

We explore the potential of various generalized feature extraction modules within a comprehensive classification pipeline, separating feature extraction from the fi-

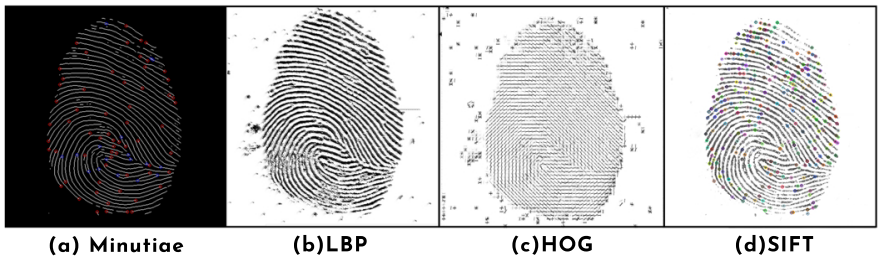

Fig. 2: Outputs of different texture analysis techniques used in feature engineering: a) Minutiae points [41] b) Local Binary Pattern [39] c) Histogram of Gradients [11] d) Scale-Invariant Feature Transform [46]

nal classification task and employing various generalized classifiers. The different image preprocessing techniques used for feature extraction are:

1. *Minutiae:* Minutiae points are the precise ridge characteristics, such as terminations and bifurcations, retrieved from pre-processed fingerprint pictures using binarization, thinning, and crossing number analysis methods [41]. We obtained a compact and representative image representation by quantizing these features into visual words using the bag-of-words (BOW) approach [47]. This enabled efficient comparison of visual word histograms for measuring image similarity, making the technique suitable for fingerprint anti-spoofing.

2. *Local Binary Pattern:* The local binary patterns (LBP) [39] are texture descriptors that compare the central pixel with its neighbours, generating a binary code that can represent different combinations. Considering two-bit transitions and reducing the feature vector length, we used the uniform or rotation-invariant LBP descriptor [6] in our study. We specifically take into account a radius of 24 pixels and eight neighbours around the core pixel. This configuration allowed us to capture and effectively represent local texture patterns in our analysis.

3. *Histogram of Gradients:* The histogram of oriented gradients (HOG) [11] is a feature descriptor that effectively characterizes visual material by quantifying gradient orientations within a detection frame. The gradient orientations used in our analysis were quantified using nine bins within a detection window. We divided the image into 8x8 pixel cells and then arranged them into blocks of 2x2 pixel cells. Using HOG with these settings successfully characterized the visual content of fingerprint images.

4. *Scale-Invariant Feature Transform (SIFT):* We used the bag-of-words method [47] with SIFT [46] for liveness detection due to its effectiveness and versatility. The SIFT methodology allowed us to extract robust and distinctive local features from the images, facilitating accurate description and identification by creating visual word histograms from the extracted key points.

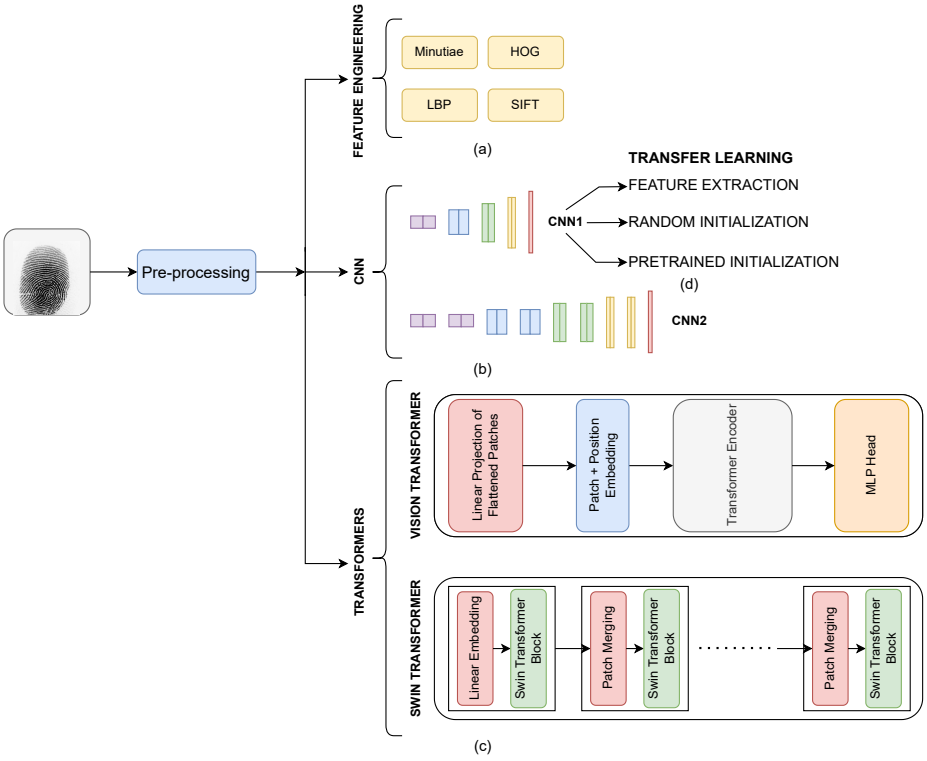

Fig. 3: Overview of the proposed comparative framework: (a) Handcrafted feature engineering methods used with different classifiers, (b) CNN1 (small architectures) and CNN2 (large architectures) are compared using same and cross-sensor evaluation, (c) Transformers (ViT and Swin Transformer), (d) Random initialization compared and contrasted with transfer learning.

The effectiveness of each of these important features is analyzed independently on 2 datasets in Table 3.

3.2 CNN and Transformer based analysis

Transfer learning efficiently utilises pre-trained models from source tasks to address data scarcity [42]. Rather than beginning from zero, transfer learning allows models to benefit from prior information and generalisations learnt from a different but related activity. We tested fine-tuning and feature extraction strategies, adapting models trained on the ImageNet dataset [14] for liveness detection.

1. *Deep Feature Extractor:* We employed DenseNet-121 [27] and VGG-19 [43], well-known pre-trained CNN models, as feature extractors. Densely connected convolutional blocks in densenet enable feature reuse and propagation, while the deep network structure of VGG is renowned for its simple

Table 1: Our classification of architectures with trainable parameters and floating point operations (FLOPs).

Model	**Parameters**	**FLOPs**	**Our Classification**
Resnet-18	12 Million	2 Billion	Small
MobileNet v3 Large	5 Million	0.22 Billion	Small
EfficientNet B0	5.3 Million	0.39 Billion	Small
EfficientNet B1	7.8 Million	0.70 Billion	Small
EfficientNet B7	66 Million	37 Billion	Large
Densenet 161	29 Million	8 Billion	Large
ViT-base 16 patches	86 Million	55.4 Billion	Transformer
Swin Transformer Tiny	29 Million	4.5 Billion	Transformer

structure and generalizability. We extracted high-level abstract representations from the input data by utilizing the whole model without the final classifier head and tested different classification methods.

2. *Small CNN:* Considering the number of parameters as well as generalizability about feature representation, we have chosen ResNet-18 [22], MobileNet v3 [26], EfficientNet B0, EfficientNet B1 [24] for liveness detection using fine-tuning by re-training the models initialized with imagenet weights. For evaluation, we employed tenCrop as given in [34], which divides the input image into ten overlapping patches, evaluates each patch separately, and finally combines the results. TenCrop improves the reliability of our classification task by collecting various spatial information from distinct parts of the fingerprint.

3. *Large CNN:* Large CNN (EfficientNet B7, DenseNet-161) have a better learning capacity as compared to small CNN and are better able to capture intricate and complex patterns in data. We tested Large CNN for liveness detection similarly to small CNN, by using the same training and evaluation scheme to test for the effectiveness of different architectures keeping other variables constant.

4. *Transformer:* The transformer design [45], which is built on the concept of self-attention mechanisms allows the network to balance the significance of various locations in the input sequence when processing each element. Vision Transformer (ViT) [16] applies the transformer architecture developed for language tasks to visual tasks such as image classification, object detection, and image segmentation by replacing the sequence of words with a grid

of image patches. The Swin Transformer [29] modifies the original Transformer's [45] self-attention mechanism and provides a hierarchical structure that enables more efficient processing of large-scale pictures. This mechanism allows the model to capture local dependencies within each window while also considering global dependencies across different windows. We evaluate the effectiveness of ViT-base 16 patches version and Swin transformer tiny version for liveness detection.

4 Experiments

4.1 Dataset

The LivDet datasets [30,20] are well known as a significant benchmark for evaluating the effectiveness of fingerprint liveness detection methods. The dataset used in this study is LivDet 2015 [33], a well-known and widely available source for research on fingerprint liveness detection. The dataset contains images from four optical scanners: Green Bit [4], Biometrika [1], Digital Persona [3], and Crossmatch [2]; over 4000 images on each device, including live images taken from numerous finger acquisitions in various conditions, including normal mode, wet and dry fingers, and high and low pressure. The spoof fingerprints present in an equal distribution in the dataset are gathered using cooperative techniques. A training set for algorithm configuration and a testing set for performance assessment are present in the dataset. We have followed the same settings for wide applicability of our study. We did not utilize the Biometrika dataset in this study due to its resolution of 1000 dpi, which exceeds the typical 500 dpi resolution used in real-world scenarios. We also rejected the Digital Persona dataset due to its small image size, which does not yield satisfactory results [33]; subsequently, choosing CrossMatch and GreenBit datasets for evaluating the proposed comparative framework.

Table 2: Scanner specifications and materials used for fingerprint acquisition in training LivDet 2015 dataset [33]

Scanner	Model	DPI	Image Size	Samples	Materials Used
Digital Persona	U.are.U 5160	500	252x324	4500	Ecoflex, Gelatin, Latex, Wood Glue
Cross-Match	L Scan Guardian	500	640x480	5948	Body Double, EcoFlex, PlayDoh
GreenBit	Dacty-Scan 26	500	500x500	4500	Ecoflex, Gelatin, Latex, Wood Glue
Biometrika	HiScan-PRO	1000	1000x1000	4250	Gelatin, Latex, Wood Glue

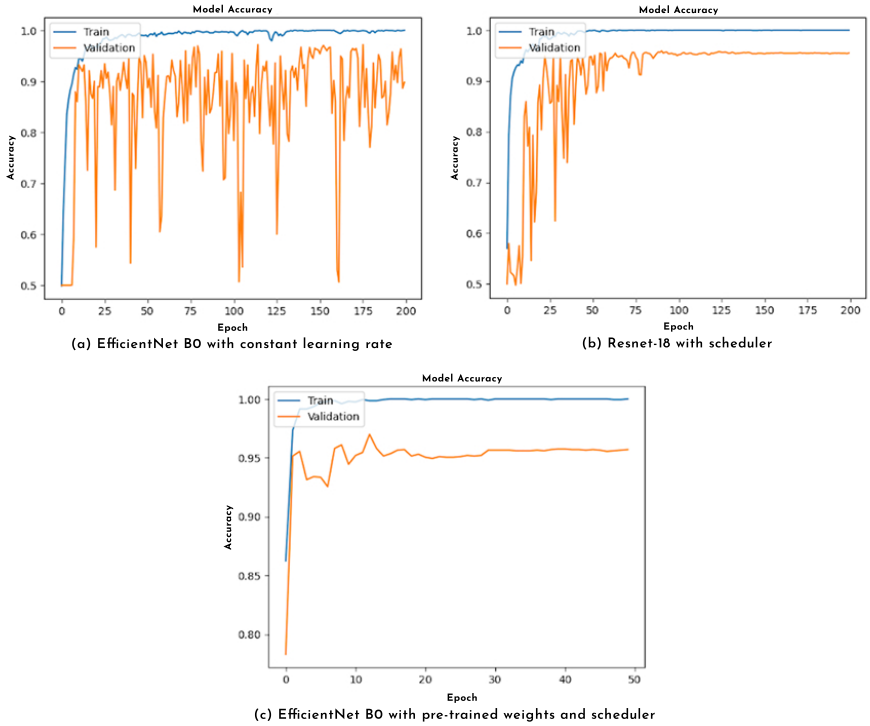

Fig. 4: Model accuracy at different epochs for different training schemes.

4.2 Performance Metrics

Standard evaluation metrics for liveness detection are accuracy, attack presentation error classification rate (APCER), bona fide presentation classification error rate (BPCER) and average classification error (ACE).

Let, N_{live} = Total number of live fingerprint images,
N_{spoof} = Total number of spoof fingerprint images
$N_{spoof|live}$ = Number of live fingerprints misclassified as spoof,
$N_{live|spoof}$ = Number of spoof fingerprints misclassified as live
Then,

$$APCER = \frac{N_{live|spoof}}{N_{spoof}} \quad ; \quad BPCER = \frac{N_{spoof|live}}{N_{live}} \tag{1}$$

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

Table 3: Average classification error (ACE) scores were obtained by evaluating different feature extractors (Minutiae, LBP, HOG, SIFT, VGG-19, Densenet-121) and classifiers (SVM-linear, RBF, polynomial; ANN; Random Forest; XG Boost; linear regression) on the Crossmatch and GreenBit datasets. The ACE scores provide insights into the effectiveness and performance of these combinations.

Dataset	Model	Minutiae	LBP	HOG	SIFT	VGG	Densenet
CrossMatch	SVM(linear)	43.01	18.70	35.70	10.36	19.37	14.70
	SVM(rbf)	34.77	**16.14**	33.71	10.68	15.72	14.59
	SVM(poly)	33.52	22.91	**33.56**	10.96	21.83	18.20
	ANN	37.23	16.92	34.67	**9.01**	15.64	**12.84**
	Random Forest	35.93	17.48	43.53	10.18	22.55	23.96
	XG Boost	**33.42**	16.99	36.47	9.63	19.37	18.68
	Logistic Regression	44.77	35.20	44.16	10.57	**15.38**	24.78
GreenBit	SVM(linear)	38.15	10.42	17.24	7.05	13.67	10.50
	SVM(rbf)	31.35	13.61	13.90	6.95	12.97	**7.94**
	SVM(poly)	31.9	12.92	17.75	7.9	13.55	11.42
	ANN	29.6	**9.44**	15.27	**6.75**	**12.06**	9.04
	Random Forest	**27.2**	10.62	13.86	11.65	13.70	12.08
	XG Boost	30.53	10.79	16.14	10.25	13.07	10.67
	Logistic Regression	37.8	19.76	**13.75**	11.7	15.65	9.94

4.3 Loss function

The model weights for small, large CNN and transformer are updated by minimizing the binary cross entropy loss function.

$$L = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i)) \tag{3}$$

4.4 Evaluation and Results

We evaluated different feature extraction modules with different classifiers. We used the SVM classifier [23] through grid search cross-validation to optimize its cost and gamma parameters. Next, we used artificial neural network (ANN) classifier, employing a sequential model with dense layers, and trained using the Adam optimizer. Subsequently, random forest classifier [7], an ensemble classifier, was employed by initializing it with a set of 1000 decision trees. After that, we used the logistic regression classifier [37] with maximum iteration set to 500. Finally, XG Boost [12], an optimized gradient boosting algorithm, was employed for classification. The number of trees in the ensemble and the learning rate were set to 100 and 0.3, respectively. The ACE value from each of the above-mentioned

Table 4: Our evaluation of various deep learning models on GreenBit and Crossmatch datasets. Models are re-trained for all layers while they are initialized with imagenet pretrained weights as a starting point. The models utilized include small CNNs, large CNNs and transformers as defined in Table 1. Training and testing times per epoch are reported for single GPU setup (Nvidia RTX 3060 12GB) for small CNNs and asterisk-marked (*) multi GPU setup (2xNvidia RTX 4080 16GB) for large CNNs and transformers. Testing time is greater than training time due to ten crop evaluation.

Model	**Train**	**Test**	**Accuracy**	**ACE**	**Training Time(s)**	**Testing Time(s)**
Resnet-18	CrossMatch	CrossMatch	99.11	0.98	6.19	16.67
		GreenBit	62	38		14.24
	GreenBit	CrossMatch	84.94	14.01	4.26	16.65
		GreenBit	97.25	2.75		14.47
Mobilenet v3 Large	CrossMatch	CrossMatch	99.32	0.81	6.04	15.33
		GreenBit	65.40	34.60		13.12
	GreenBit	CrossMatch	72.10	37.50	4.08	15.13
		GreenBit	97.15	2.85		12.91
EfficientNet B0	CrossMatch	CrossMatch	99.70	**0.36**	9.83	21.67
		GreenBit	63.10	36.90		18.54
	GreenBit	CrossMatch	87.15	6.58	6.73	22.08
		GreenBit	97.65	2.35		18.78
EfficientNet B1	CrossMatch	CrossMatch	99.06	1.29	15.70	54.24
		GreenBit	67.40	32.60		46.18
	GreenBit	CrossMatch	82.48	22.33	10.68	54.52
		GreenBit	97.45	2.55		46.64
EfficientNet B7	CrossMatch	CrossMatch	99.11	1.03	330.69*	643.07*
		GreenBit	55.25	44.75		546.26*
	GreenBit	CrossMatch	79.46	24.80	228.79*	646.80*
		GreenBit	98.05	1.95		548.71*
Densenet161	CrossMatch	CrossMatch	99.36	**0.75**	27.28*	95.20*
		GreenBit	61.35	38.65		80.68*
	GreenBit	CrossMatch	85.03	19.08	17.48*	94.02*
		GreenBit	96.10	3.90		78.89*
ViT-base 16 patches	CrossMatch	CrossMatch	96.00	4.55	29.09*	31.22*
		GreenBit	67.45	32.55		23.49*
	GreenBit	CrossMatch	55.65	34.89	18.88*	29.88*
		GreenBit	89.10	10.9		24.28*
Swin Transformer Tiny	CrossMatch	CrossMatch	96.85	3.81	23.50*	48.02*
		GreenBit	58.10	41.90		37.96*
	GreenBit	CrossMatch	48.26	40.62	13.39*	50.15*
		GreenBit	90.40	9.60		40.86*

Table 5: Comparison of pre-trained imagenet models [14] and models initialized with random weights. The pre-trained models were also used as feature extractors to assess the influence of imagenet weights on model accuracy. The models pre-trained on imagenet outperform those initialized with random weights even though the two domains are very different.

Model	Dataset	Feature Extractor	Random Weights	Imagenet Weights
Resnet-18	CrossMatch	96.26	98.68	99.10
	GreenBit	93.70	96.25	97.25
	Average Cross Sensor	71.17	66.50	73.46
Efficientnet B0	CrossMatch	95.32	98.85	99.70
	GreenBit	94.35	97.20	97.65
	Average Cross Sensor	68.34	66.96	75.12
Efficientnet B1	CrossMatch	96.68	98.97	99.06
	GreenBit	94.95	96.65	97.45
	Average Cross Sensor	71.9	67.23	74.94

classifiers along with each mentioned feature extractor are presented in Table 3.

CNN and Transformer based evaluation: Standard network architectures like Resnet [22], Mobilenet [26], Efficientnet [24], ViT [16] were used for classification, with the classifier head replaced by a 256-node penultimate layer and 2-nodes at final classifier head. A dropout layer with dropout probability of 0.4 was used for regularization. The models were initialized with imagenet weights [14] and standard data augmentation techniques like random cropping and horizontal flipping of the images were used for increasing generalizability. The evaluation process involved cropping the original image into 10 patches (four from the corners, one from center and their horizontally flipped counterparts) and considering the average of the LogSoftmax of the model output for classification.

We performed cross-sensor testing to evaluate the model's capacity to adjust to new sensors. This involved assessing the models on a sensor other than the ones they were initially trained on. The results of different architectures, when trained and tested on CrossMatch and GreenBit 2015 dataset, are summarised in Table 4.

4.5 Effect of Transfer Learning

We tested the effects of transfer Learning on the model by using the model pre-trained on imagenet weights [14] as a feature extractor and training only the last 2 newly added layers. Initialization using random weights was also used to test the effect of those weights on the accuracy. As seen in Table 5, models pre-trained on imagenet perform better than random initialization on fingerprint

liveness classification even though the domain of both differ significantly. The pre-trained models also generalize better as seen by the average cross sensor accuracy.

5 Conclusion

This provides the useful insights that feature representation learnt via imagenet training offers a very good initial starting point for optimization algorithms to find the optimal neural network weights. We presented a detailed comprehensive framework for accurate fingerprint liveness detection, focusing on feature engineering and classification techniques. Feature extraction methods, including texture analysis methods and deep feature extractors, were extensively evaluated on cross domain and cross-sensor datasets. The effectiveness of these techniques in capturing distinctive patterns and improving the precision of liveness detection in fingerprints was observed. Additionally, the efficacy of patch-based CNN classification algorithms to generalize to other sensors was examined. The effects of transfer learning on models was evaluated by initializing models with imagenet weights, using the pre-trained models as feature extractor and comparing them to random initialization. This research significantly advances fingerprint liveness detection by exploring feature engineering techniques, classification algorithms, CNN and transformer based approaches, examining accuracy and robustness.

Acknowledgement: This work is supported by the project C3IHUB/EE/2023221 C3i (cybersecurity and cybersecurity for Cyber-Physical Systems) Innovation Hub IIT Kanpur.

References

1. Biometrika HiScan-Pro Scanner. https://www.neurotechnology.com/fingerprint-scanner-biometrika-hiscan-pro.html
2. Cross-Match L Scan Guardian Scanner. http://www.itworks.co.th/L_Scan_Guardian/L_SCAN_Guardian.pdf
3. Digital Persona U.are.U 5160 Scanner. https://www.neurotechnology.com/fingerprint-scanner-digitalpersona-u-are-u-5100-5160.html
4. GreenBit Dacty-Scan 26 Scanner. https://www.neurotechnology.com/fingerprint-scanner-green-bit-dactyscan-26.html
5. Abhyankar, A.S., Schuckers, S.C.: A wavelet-based approach to detecting liveness in fingerprint scanners. In: Biometric Technology for Human Identification. vol. 5404. International Society for Optics and Photonics, SPIE (2004)
6. Ahonen, T., Matas, J., He, C., Pietikäinen, M.: Rotation invariant image description with local binary pattern histogram fourier features. In: Image Analysis. Springer Berlin Heidelberg (2009)
7. Ali, J., Khan, R., Ahmad, N., Maqsood, I.: Random forests and decision trees. International Journal of Computer Science Issues(IJCSI) (2012)
8. Antonelli, A., Cappelli, R., Maio, D., Maltoni, D.: Fake finger detection by skin distortion analysis. IEEE Transactions on Information Forensics and Security (2006)

9. Anusha, B., Banerjee, S., Chaudhuri, S.: Defraudnet:end2end fingerprint spoof detection using patch level attention. In: IEEE Winter Conference on Applications of Computer Vision (WACV) (2020)
10. Baldisserra, D., Franco, A., Maio, D., Maltoni, D.: Fake fingerprint detection by odor analysis. In: Advances in Biometrics. Springer Berlin Heidelberg (2005)
11. Carcagnì, P., Del Coco, M., Leo, M., Distante, C.: Facial expression recognition and histograms of oriented gradients: a comprehensive study. SpringerPlus (2015)
12. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Association for Computing Machinery (2016)
13. Chugh, T., Cao, K., Jain, A.: Fingerprint spoof buster: Use of minutiae-centered patches. IEEE Transactions on Information Forensics and Security (2018)
14. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition
15. Dharavath, K., Talukdar, F.A., Laskar, R.H.: Study on biometric authentication systems, challenges and future trends: A review. In: 2013 IEEE International Conference on Computational Intelligence and Computing Research
16. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2021)
17. Drahansky, M., Notzel, R., Funk, W.: Liveness detection based on fine movements of the fingertip surface. In: IEEE Information Assurance Workshop (2006)
18. Galbally, J., Alonso-Fernandez, F., Fierrez, J., Ortega-Garcia, J.: Fingerprint liveness detection based on quality measures. In: First IEEE International Conference on Biometrics, Identity and Security (BIdS) (2009)
19. Ghiani, L., Hadid, A., Marcialis, G.L., Roli, F.: Fingerprint liveness detection using binarized statistical image features. In: 2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)
20. Ghiani, L., Yambay, D., Mura, V., Tocco, S., Marcialis, G.L., Roli, F., Schuckcrs, S.: Livdet 2013 fingerprint liveness detection competition. In: 2013 International Conference on Biometrics (ICB)
21. Gomez-Barrero, M., Kolberg, J., Busch, C.: Multi-modal fingerprint presentation attack detection: Analysing the surface and the inside. In: International Conference on Biometrics (ICB) (2019)
22. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
23. Hearst, M., Dumais, S., Osuna, E., Platt, J., Scholkopf, B.: Support vector machines. IEEE Intelligent Systems and their Applications (1998)
24. Hoang, V.T., Jo, K.H.: Practical analysis on architecture of efficientnet. In: 14th International Conference on Human System Interaction (HSI) (2021)
25. Hosseini, S.: Fingerprint vulnerability: A survey. In: 2018 4th International Conference on Web Research (ICWR)
26. Howard, A., Sandler, M., Chu, G., Chen, L.C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q.V., Adam, H.: Searching for mobilenetv3. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
27. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.: Densely connected convolutional networks (2017)

28. Johnson, P., Schuckers, S.: Fingerprint pore characteristics for liveness detection. In: 2014 International Conference of the Biometrics Special Interest Group (BIOSIG)
29. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
30. Marcialis, G.L., Lewicke, A., Tan, B., Coli, P., Grimberg, D., Congiu, A., Tidu, A., Roli, F., Schuckers, S.: First international fingerprint liveness detection competition—livdet 2009. In: Image Analysis and Processing – ICIAP 2009
31. Marcialis, G.L., Roli, F., Tidu, A.: Analysis of fingerprint pores for vitality detection. In: 2010 20th International Conference on Pattern Recognition
32. Menotti, D., Chiachia, G., Pinto, A., Schwartz, W.R., Pedrini, H., Falcão, A.X., Rocha, A.: Deep representations for iris, face, and fingerprint spoofing detection. IEEE Transactions on Information Forensics and Security (2015)
33. Mura, V., Ghiani, L., Marcialis, G.L., Roli, F., Yambay, D.A., Schuckers, S.A.: Livdet fingerprint liveness detection competition. In: IEEE 7th International Conference on Biometrics Theory, Applications and Systems (BTAS) (2015)
34. Nogueira, R.F., de Alencar Lotufo, R., Campos Machado, R.: Fingerprint liveness detection using convolutional neural networks. IEEE Transactions on Information Forensics and Security (2016)
35. Organization, I.S.: "iso/iec 30107-1:information technology - biometric presentation attack detection - part 1: Framework" (2016)
36. Osten, D.W., Carim, H.M., Blan, B.L., Arneson, M.R.: Biometric, personal authentication system (1995)
37. Peng, J., Lee, K., Ingersoll, G.: An introduction to logistic regression analysis and reporting. Journal of Educational Research - J EDUC RES (2002)
38. Pinto, A., Pedrini, H., Krumdick, M., Becker, B., Czajka, A., Bowyer, K., Rocha, A.: Counteracting Presentation Attacks in Face, Fingerprint, and Iris Recognition (2018)
39. Prakasa, E.: Texture feature extraction by using local binary pattern. Jurnal INKOM (2016)
40. Van der Putte, T., Keuning, J.: Biometrical Fingerprint Recognition: Don't get your Fingers Burned. Springer US (2000)
41. Ravi, J., K B, R., K R, V.: Fingerprint recognition using minutia score matching. CoRR (2010)
42. Ribani, R., Marengoni, M.: A survey of transfer learning for convolutional neural networks. In: 32nd SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T) (2019)
43. Tammina, S.: Transfer learning using vgg-16 with deep convolutional neural network for classifying images (2019)
44. Tan, B., Schuckers, S.C.: New approach for liveness detection in fingerprint scanners based on valley noise analysis. Journal of Electronic Imaging (2008)
45. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems (2017)
46. Wang, R., Jeon, J.W.: Design of real-time sift feature extraction. In: IECON - 43rd Annual Conference of the IEEE Industrial Electronics Society (2017)
47. Zhang, Y., Jin, R., Zhou, Z.H.: Understanding bag-of-words model: A statistical framework. International Journal of Machine Learning and Cybernetics (2010)