International Journal of Cognitive Computing in Engineering

Face PAD using Vision Transformers through transformed input spaces

--Manuscript Draft--

Manuscript Number:	IJCCE-D-23-00105
Full Title:	Face PAD using Vision Transformers through transformed input spaces
Short Title:	
Article Type:	Full Length Article
Section/Category:	Artificial Intelligence
Keywords:	Vision Transformer; Spoofing; Deep Learning; CNN; Security; Face; Biometric Systems; PAD; DeiT; RGB; LBP; HSV; YCrCb; Grayscale; LA*B*
Corresponding Author:	Abhinav Dholi Vellore Institute of Technology INDIA
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	Vellore Institute of Technology
Corresponding Author's Secondary Institution:	
First Author:	Abhinav Dholi
First Author Secondary Information:	
Order of Authors:	Abhinav Dholi
	G C Charan
	Bhavya Harchandani
	M P Gopinath
	P R Subramaniam
	Aarthy S L
Order of Authors Secondary Information:	
Abstract:	Facial Recognition is used to verify an identity in biometric systems since they are one of the most distinguishing characteristics. Due to the advancements in social media and networks, photos and videos of a person are easily accessible and the same can be used to cheat a facial recognition system to gain unauthorized access. These attacks are commonly called face spoofing attacks and the most common types are face photo attacks, replay attacks, and 3D mask attacks. We propose the use of vision transformers, a recently developed type of deep learning model for detecting presentation attacks (PADs) that are to deceive the system by presenting fake or manipulated facial images, such as through print or display attacks. In the proposed methodology, we extract LBP texture filters of the facial data for different color spaces and train a model with the DeiT Vision Transformer (ViT), to classify PADs. We provide a detailed analysis of how the proposed model architecture performs when trained with the LBP texture filters of the different color spaces and find the best-performing combination for it. The experiments were performed on the OuluNPU Dataset, and the results were compared to several state-of-the-art baseline methodologies. Our proposed method went on to provide results comparable to these baseline methods. Our method outperformed all the baseline methods for the fourth protocol of the OuluNPU dataset.
Suggested Reviewers:	
Opposed Reviewers:	
Additional Information:	

Question	Response
Free Preprint Service Do you want to share your research early as a preprint? Preprints allow for open access to and citations of your research prior to publication. International Journal of Cognitive Computing in Engineering offers a free service to post your paper in a journal-branded First Look space on SSRN, an open access research repository, when your paper enters peer review. Once on SSRN, your paper will benefit from early registration with a DOI and early dissemination that facilitates collaboration and early citations. It will be available free to read regardless of the publication decision made by the journal. This will have no effect on the editorial process or outcome with the journal. Please consult the SSRN Terms of Use and FAQs.	YES, I want to share my research early and openly as a preprint.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☒The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

No interests to declare

Face PAD using Vision Transformers through transformed input spaces

Abhinav Dholi[a,*], G C Charan[a,*], Bhavya Harchandani[a,*], M P Gopinath[a], Prabhakar Rontala Subramaniam[b] and Aarthy S L[a]

[a] *Vellore Institute of Technology, Vellore, India*
[b] *University of Kwazulu-Natal, Westville, Durban, South Africa*

ARTICLE INFO

Keywords:
Vision Transformer, Spoofing, Deep Learning, CNN, Security, Face, Biometric Systems, PAD, DeiT, RGB, LBP, HSV, YCrCb, Grayscale, LA*B*

ABSTRACT

Facial Recognition is used to verify an identity in biometric systems since they are one of the most distinguishing characteristics. Due to the advancements in social media and networks, photos and videos of a person are easily accessible and the same can be used to cheat a facial recognition system to gain unauthorized access. These attacks are commonly called face spoofing attacks and the most common types are face photo attacks, replay attacks, and 3D mask attacks. We propose the use of vision transformers, a recently developed type of deep learning model for detecting presentation attacks (PADs) that are to deceive the system by presenting fake or manipulated facial images, such as through print or display attacks. In the proposed methodology, we extract LBP texture filters of the facial data for different color spaces and train a model with the DeiT Vision Transformer (ViT), to classify PADs. We provide a detailed analysis of how the proposed model architecture performs when trained with the LBP texture filters of the different color spaces and find the best-performing combination for it. The experiments were performed on the OuluNPU Dataset, and the results were compared to several state-of-the-art baseline methodologies. Our proposed method went on to provide results comparable to these baseline methods. Our method outperformed all the baseline methods for the fourth protocol of the OuluNPU dataset.

1. Introduction

Faces are one of the most distinguishing characteristics to identify a person, in addition to fingerprints and irises. In this digitally advancing world, intelligent identity authentication mechanisms such as face and fingerprint recognition systems are being used instead of traditional passwords as they are more convenient and secure to use in numerous fields. Large-scale face biometrics-based authentication systems are becoming more common; nowadays, even a low-budget mobile phone has a good-quality face unlock system as an alternative to usual passcodes Li, Correia and Hadid (2017). Face recognition is also used in education (for tracking attendance), immigration (for border control, particularly when it involves criminals and persons of interest who attempt to cross the border), access control, automobile security, and many such commercial areas. As face recognition is becoming more popular, security threats are also becoming stronger. Due to the advancements in social media and networks, photos and videos of a person are easily accessible and the same can be used to cheat a facial recognition system to gain unauthorized access Li et al. (2017).

An unprotected face recognition (FR) system could be fooled by simply placing artifacts in front of the camera, such as a photograph or video. A presentation attack instrument (PAI) is the artifact used in such an attack. Anjos, Günther, Pereira, Korshunov, Mohammadi and Marcel (2017)

Presentation attack detection (PAD) systems, as the name implies, are designed to safeguard FR systems from such malicious attempts. Though a wide range of presentation attacks are possible, the majority of research efforts have concentrated on the detection of 2D attacks such as prints and replays, owing to the ease with which such attack instruments can be produced. The majority of PAD research focuses on detecting these attacks using only the RGB spectrum, either using feature-based methods or Convolutional Neural Network (CNN)-based approaches. Several feature-based methods for performing PAD have been proposed over the years, using color, texture, motion, liveliness cues, histogram features Boulkenafet, Komulainen and Hadid (2015), local binary pattern Komulainen, Hadid and Pietikäinen (2011), Chingovska, Anjos and Marcel (2012), and motion patterns Anjos and Marcel (2011).

The development of deep learning in the field of computer vision and image classification provided more effective solutions for face spoof detection systems. Convolutional Neural Networks (CNNs) have been widely used in the field of face spoof detection. CNNs that include auxiliary information in the form of binary or depth supervision, in particular, have shown to significantly increase performance Atoum, Liu, Jourabloo and Liu (2017) George and Marcel (2019). Nevertheless, the maximum of these approaches perform effectively only against 2D attacks, and their efficacy decreases when tested against advanced 3D and partial attacks Liu, Stehouwer, Jourabloo and Liu (2019). Even when dealing with 2D attacks, these models frequently fail to generalize to previously unseen attacks and environments.

*These authors contributed equally
abhinav.dholi@gmail.com (A. Dholi); charangajjala7@gmail.com (G.C. Charan); bhavya.harchandani.2001@gmail.com (B. Harchandani); mpgopinath@vit.ac.in (M.P. Gopinath); prabhakarr@ukzn.ac.za (P.R. Subramaniam); aarthy.sl@vit.ac.in (A.S. L)
ORCID(s): 0009-0009-5583-2326 (A. Dholi); 0009-0005-4697-1042 (G.C. Charan); 0009-0008-1585-0901 (B. Harchandani); 0000-0001-7511-2910 (M.P. Gopinath); 0000-0003-2719-3503 (P.R. Subramaniam); 0000-0002-3412-1818 (A.S. L)

To address the limitations of PAD systems, many multi-channel techniques have recently been proposed George, Mostaani, Geissenbuhler, Nikisins, Anjos and Marcel (2019) Nikisins, George and Marcel (2019) George and Marcel (2021a). Though such approaches outperform RGB solutions in terms of performance, the expense of additional hardware required limits their use to safeguard legacy RGB-based FR systems. As a result, it is preferable to have a robust RGB-based PAD approach that is resistant to a wide range of 2D, 3D, and partial attacks. A PAD system should ideally be able to generalize well to unseen attacks and environments. Most of the time, the amount of data available to train a PAD model is very limited. This restricts the ability to train George and Marcel (2021b) deep architectures from the scratch. According to previous research, transfer learning from a pre-trained network is an effective method for dealing with the limited data problem Zhuang, Qi, Duan, Xi, Zhu, Zhu, Xiong and He (2021). Furthermore, because the model has seen a wide variety of images in diverse environments, a pre-trained network could aid in addressing the changes in environments.

Vision transformers are a type of neural network architecture. They differ from traditional convolutional neural networks (CNNs), commonly used for image classification as they do not rely on convolutional layers. Instead, they use self-attention mechanisms to process input data and make predictions. This makes them more efficient and scalable, as they can process input data of any size without the need for fixed-size filters Dosovitskiy, Beyer, Kolesnikov, Weissenborn, Zhai, Unterthiner, Dehghani, Minderer, Heigold, Gelly, Uszkoreit and Houlsby (2021). In this work, we investigate the effectiveness of the DeiT (data-efficient image transformers) vision transformer model for the presentation attack detection problem through transformed input spaces. DeiT are more efficiently trained transformers for image classification, requiring far less data and far less computing resources compared to the original ViT models Touvron, Cord, Douze, Massa, Sablayrolles and Jégou (2021). We investigate the performance of this method in challenging unseen attack and cross-database scenarios.

To the best knowledge of the authors, this is the first work using DeiT vision transformers for the presentation attack detection task. The main contributions of this work are listed below:

- Extract LBP texture filters of the facial dataset of the different color spaces
- Build a robust Vision Transformer based PAD framework
- Evaluate the robustness of the model by error evaluation dedicated to biometrics over several protocols provided with the dataset.

The rest of the paper is organized as follows:

The second section comprises of the related work to the study we have conducted. The third section presents the methodology of our proposed framework. The experimental setup is mentioned in the fourth section. The fifth section has the results obtained. The sixth section concludes our paper and finally, the seventh section provides future work.

2. Related Work

A thorough literature review was conducted in the domain of face detection. According to the authors George and Marcel (2019), biometric systems can be attacked both directly and indirectly. The attackers can use photographs, gelatin fingers, contact lenses, etc., to generate synthetic samples of biometric traits to perform direct attacks (spoofing attacks) through which access to authentication systems can be acquired. In the case of indirect attacks, they obtain information about the internal functionality of the system. They modify the algorithms used to protect biometric templates even further. Because of its low acquisition cost and universality, the face is a promising biometric authentication trait. However, various methods can be used to fool face recognition systems George et al. (2019).

2.1. Conventional Methods

A lot of work is being done in the field of spoofing detection in faces employing fixed features. These features could be based on motion, texture, reflectance properties, etc. They use handcrafted features to differentiate between real and spoofed faces. Before training the data with any algorithm, the features are calculated. Techniques like Haar-like features and Linear Discriminant Analysis (LDA) can be used to scan faces in photos and identify spoofing attacks Erdogmus and Marcel (2013). The co-occurrence of adjacent local binary pattern (CoALBP) technique was proposed in Gragnaniello, Poggi, Sansone and Verdoliva (2015) to extract facial texture features to detect spoofed faces. The authors Li, Tan and Jain (2004) investigated textural differences to detect spoofing by examining the Fourier spectra of 2D and 3D photos. They discovered that the surface reflection characteristic caused a change in the frequency distributions of these images. To distinguish between real and spoofed faces, the authors Mahitha (2018) retrieved hidden facial texture features using Difference-of-Gaussian (DoG) filters. The authors Tan, Li, Liu and Jiang (2010) propose an approach for extracting time-spectral descriptors from a video, capturing both spatial and temporal information from a biometric sample. These descriptors have been demonstrated to be effective in detecting various types of attacks in a wide range of environments. Researchers in the field of spoofing detection have presented various motion-based approaches. This involves detecting motions such as eye movements Siddiqui, Bharadwaj, Dhamecha, Agarwal, Vatsa, Singh and Ratha (2016). Other approaches involve evaluating image quality and the property of reflection to distinguish between real and fake faces. To identify spoofing, authors use reflection, color diversity, and blurriness between real and fake faces.

2.2. Deep learning-based techniques

These days, the most widely used approach to deal with spoofing is the use of Convolutional Neural Networks (CNN). CNNs have been used for face presentation attack detection (PAD) in recent years. The use of CNNs for PAD has been shown to be effective in detecting various types of presentation attacks, including printed photos, replay attacks, and 3D masks Almeida, Andaló, Padilha, Bertocco, Dias, Torres, Wainer and Rocha (2020). One of the most common methodologies for CNN-based PAD is to use a binary classification approach, where the input image is classified as either real or fake Benlamoudi, Bekhouche, Maarouf, Bensid, Ouahabi, Hadid and taleb ahmed (2022). Another methodology is to use a multi-class classification approach, where the input image is classified into one of several classes, such as real, printed photo, or replay attack Benlamoudi et al. (2022). In general, CNN-based PAD systems consist of two main components: feature extraction and classification Benlamoudi et al. (2022). The feature extraction component extracts features from the input image that are relevant for PAD. The classification component then uses these features to classify the input image as either real or fake. Some of the most commonly used CNN architectures for PAD include VGGNet Zhang, Zhao, Yan, Gao and Chen (2020), ResNet Zhang et al. (2020), and MobileNet Zhang et al. (2020). These architectures have been shown to be effective in detecting presentation attacks in various scenarios. However, the research is in progress, and little literature is available that studies the detection of face spoofing using deep learning. One of the approaches discussed in the previous literature relied on dimensionality reduction and feature extraction of input frames using pre-trained weights of convolutional autoencoders, followed by classification using the SoftMax classifier Arora, Bhatia and Mittal (2021). Authors have also used combinations of texture and illumination features with CNNs, specifically by generating the differences of Gaussian (DoG) and low pass filters which were used to extract texture features and specular reflection features respectively then used to train a CNN with those features and original images by stacking them as multi-channel input Gu, Xia, Fei, Yuan and Zhang (2020). Another approach uses a Deep channel that contains multiple layers of CNNs followed by a pooling layer and a fully connected layer. A Shallow feature unit extracts color texture features, such as LBP, CoALBP, and LBQ, from gray-scale, HSV, and YCbCr color spaces. SiW, ROSE-Youtu, and NUAA Imposter datasets were used by the authors Hashemifard and Akbari (2021). In another literature, the authors suggested dividing the method into two streams: patch-based CNN and depth-based CNN. They trained a deep neural network end-to-end to learn rich appearance attributes for the patch-based CNN stream, and a fully convolutional network (FCN) to estimate the depth of a face picture for the depth-based CNN stream, then merged both the streams Atoum et al. (2017). Another approach employs a pre-trained deep residual network to acquire highly discriminative features, which it then combines with Long Short-Term Memory (LSTM) units to uncover long-range temporal associations of video frames for classification on Replay attack and CASIAFASD datasets Tu and Fang (2017). Another deep learning-based method for face PAD is Generative Adversarial Networks (GANs). GANs have been used to generate synthetic face images that can be used to train a classifier for detecting presentation attacks Brownlee (2019). There are many other deep learning-based methods for face PAD such as Capsule Networks, Recurrent Neural Networks (RNNs), and Autoencoders.

2.3. Research Gaps and Motivation

According to a Joshi, Verma, Saxena and Paraye (2019), one of the drawbacks of CNNs is that they require large amounts of data to learn everything from scratch. CNNs perform better in low data regimes due to their hard inductive bias. Another source states that CNNs are significantly slower due to operations such as maxpool. If the CNN has several layers, the training process takes a lot of time if the computer lacks a good GPU. Other research articles have also stated that ViT has great success with NLP and is now applied to images Dosovitskiy et al. (2021). Recent work has shown that ViT can achieve comparable or even superior performance on image classification tasks compared to CNNs Raghu, Unterthiner, Kornblith, Zhang and Dosovitskiy (2021).

Local Binary Pattern (LBP) is a texture descriptor that has been used in presentation attack detection (PAD) systems Peng, Meng and Long (2022). Surveys on face PAD mechanisms have found that LBP has been frequently used as image descriptors in several detection mechanisms as input to a classifier for face PADs Sharma and Selwal (2023). We are using the pre-trained DeiT (Data-Efficient Image Transformers) Vision Transformer model, proposed by Touvron et al. (2021), which is a more efficiently trained transformer model which requires a lesser amount of data as well as lesser computing resources than the original ViT model, to perform our analysis. We are motivated to provide a complete analysis of the performance of the DeiT ViT model in detecting PADs when trained on different color spaces of LBP.

3. Methodology

3.1. Local Binary Patterns (LBP)

Local binary patterns (LBP) is a type of visual descriptor used for classification in computer vision. LBP is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number. LBP is initially formed in an eight-pixel neighborhood, with the value of the central pixel serving as a threshold. Any neighbors with values greater than or equal to the center pixel are assigned a value of 1, otherwise, they are set to 0. The values after thresholding (either 0 or 1) would multiply with the corresponding pixel weight, yielding the LBP value as an additive result, which is overwritten to the pixel on which the operation is performed. Song, YAN, CHEN and Zhang (2013) **??**

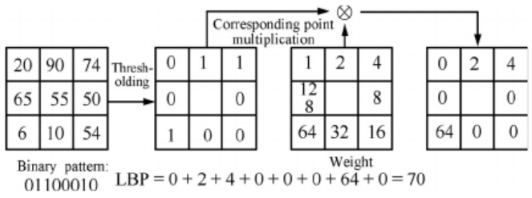

Fig. 1: LBP Algorithm Song et al. (2013)

In our methodology, we've used LBP on different color spaces, by applying the LBP algorithm separately to each color channel of an image. This results in separate LBP feature maps equivalent to the number of color channels in the provided color space, which can then be combined to create a final feature vector for texture analysis. In [Fig. 2] LBP algorithm is being performed on RGB color space where, first the R, G, and B channels are separated from the input image, then LBP is applied to those color channels individually then, the transformed color channels are merged into a single LBP transformed RGB image. Further, a similar approach is applied to other color spaces such as HSV, LA*B*, Grayscale, and YCrCb.

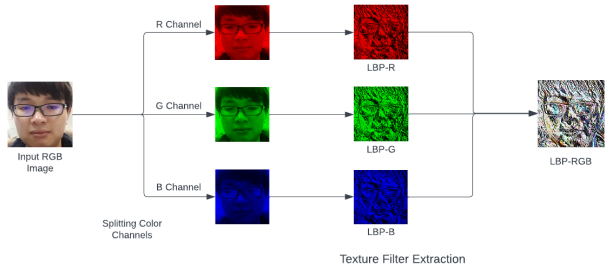

Fig. 2: RGB-LBP Process

3.2. Vision Transformer

The Vision Transformer, or ViT, is an image classification model that employs a Transformer-like architecture over patches of an image. The image is divided into fixed-size patches, that are then linearly embedded. Position embeddings are then added, and the resulting vector sequence is fed into a standard Transformer encoder. The standard approach of adding an extra learnable "classification token" to the sequence is used to perform classification Dosovitskiy et al. (2021).

ViT can be a revolutionary model in the field of image classification, where previous research has shown that it outperforms, CNN in the terms of accuracy and computational efficiency when trained over a large dataset. Unlike CNNs, ViT uses an attention mechanism to compute the relationship between all pixels in an image. This allows the model to learn global dependencies between different parts of the image. It adds position embeddings to the input image, which encodes the spatial location of each pixel. This allows the model to learn positional relationships between different regions of the image, which is particularly useful for tasks such as image classification Dosovitskiy et al. (2021) Paul and Chen (2021).

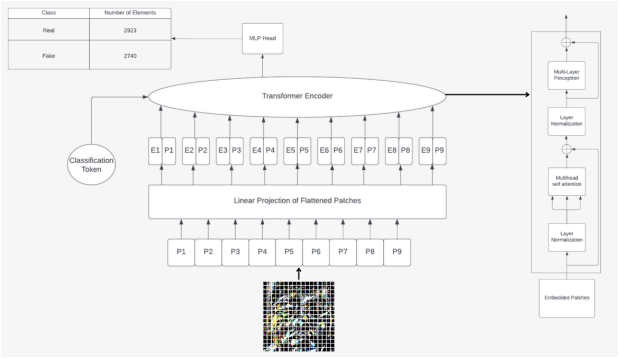

Fig. 3: Vision Transformer Architecture

An overview of the model is depicted in [Fig. 3]. A 1D sequence of token embeddings is fed into the standard Transformer. To handle 2D images, we reshape the image $R^{H\times W\times C}$ into a sequence of flattened 2D patches $x_p \in \mathbb{R}^{N\times(P^2\cdot C)}$, where (H, W) is the original image's resolution, C is the number of channels, (P, P) is the resolution of each image patch, and $N = \frac{HW}{P^2}$ is the resulting number of patches, which also serves as the Transformer's effective input sequence length. Because the Transformer employs a constant latent vector size D across all of its layers, we flatten the patches and map to D dimensions using a trainable linear projection. (Eq. 7).

The last step is adding positional encoding to get the final vector Z. Position embeddings are added to patch embeddings, conventional learnable 1D position embeddings are used Dosovitskiy et al. (2021). Till this step, "positional encodings" are added to the input embeddings at the bottom of the encoder and decoder stacks. The positional encodings and embeddings have the same dimension d_{model}, therefore the two can be added. There are various learned and fixed positional encodings to choose from George et al. (2019). In this work, sine and cosine functions of varying frequencies are used:

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}}) \quad (1)$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{model}}) \quad (2)$$

Where pos is the position and i is the dimension. That is, each positional encoding dimension corresponds to a sinusoid. The wavelengths are arranged in a geometric sequence from 2π to $10000 \cdot 2\pi$. We selected this function because we expected that it would allow the model to easily learn to attend by relative positions because PE_{pos+k} can be expressed as a linear function of PE_{pos} for any fixed offset k Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser and Polosukhin (2017).

The Transformer Encoder architecture is similar to that described in. It is comprised of several stacks of identical

blocks. Each block begins with a Multi-Head Attention layer and ends with a Feed-Forward layer Vaswani et al. (2017).

In the multi-head attention layer, standard qkv self-attention Vaswani et al. (2017) is a key component for the neural architecture. We compute a weighted sum over all values v in an input sequence $z \in \mathbb{R}^{N \times D}$ for each element. The attention weights Aij are deduced by the pairwise similarity of two elements of the sequence and their respective query q i and key k j representations.

$$[\mathbf{q},\mathbf{k},\mathbf{v}] = \mathbf{z}\mathbf{U}_{\mathbf{qkv}} \quad \mathbf{U}_{qkv} \in \mathbb{R}^{D \times 3D_h}, \tag{3}$$

$$A = \mathrm{softmax}\left(\mathbf{q}\mathbf{k}^{\top}/\sqrt{D_h}\right) \quad A \in \mathbb{R}^{N \times N}, \tag{4}$$

$$\mathrm{SA}(\mathbf{z}) = A\mathbf{v} \tag{5}$$

Multihead self-attention (MSA) is an extension of SA in which k self-attention operations, referred to as "heads," are run in parallel and their concatenated outputs are projected. When changing k, D_h (Eq. 6) is typically set to D/k to keep the compute and the number of parameters constant.

$$\mathrm{MSA}(\mathbf{z}) = \left[\mathrm{SA}_1(z); \mathrm{SA}_2(z); \cdots; \mathrm{SA}_k(z)\right] \mathbf{U}_{msa}, \quad \mathbf{U}_{msa} \in \mathbb{R}^{k} \cdot D_h \times D \tag{6}$$

To obtain the final vector of embedded dimension D, these attention heads are concatenated and passed through a dense layer Dosovitskiy et al. (2021).

Each of the two sub-layers has a residual connection, which is followed by layer normalization. The model's sub-layers and embedding layers all generate an output of embedded dimension D. The preceding step's Z vector is passed through the transformer encoder architecture to produce the context vector C.

Multi-Layer Perceptrons(MLP) contains two-layer with Gaussian Error Linear Unit(GELU). A classification head is implemented for image classification using MLP with one hidden layer at pre-training time and a single linear layer for fine-tuning. ViT's higher layers learn global features, while the lower layers learn both global and local features, which allows ViT to learn more generic patterns. Dosovitskiy et al. (2021)

$$z_0 = \left[x_{class}; x_p^1 E; x_p^2 E; \cdots; x_p^N E\right] + E_{pos}, \quad E \in \mathbb{R}^{P^2 \cdot C} \times D, \quad E_{pos} \in \mathbb{R}^{N+1} \times D \tag{7}$$

$$z'_l = MSA(LN(z_{l-1})) + z_{l-1}, \quad l = 1 \cdots L \tag{8}$$

$$z_l = MLP(LN(z'_l)) + z'_l, \quad l = 1 \cdots L \tag{9}$$

$$y = LN(z_L^0) \tag{10}$$

Once, we have our context vector C, we are only interested in the context token for classification purposes. This context token is passed through an MLP head to give us the final probability vector to help predict the class.

3.3. DeiT (Data Efficient Image Transformers)

While Vision Transformer, ViT, requires hundreds of millions of images to be pre-trained using external data, it does not generalize well when trained on insufficient amounts of data. Dosovitskiy et al. (2021). Touvron et al. (2021) proposes the Data-Efficient Image Transformer, DeiT, as a solution to this problem. While the architecture of DeiT is similar to that of ViT, it is trained on ImageNet in less than 3 days using a single computer and no external data. A teacher-student strategy is introduced with a distillation token Touvron et al. (2021).

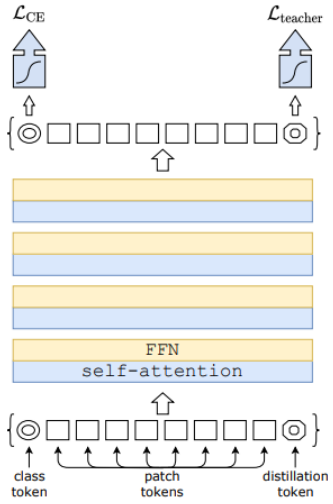

Fig. 4: Distillation Procedure Touvron et al. (2021)

In [Fig. 4], A new token, the distillation token, is added to the initial embeddings (patches and class token). The distillation token is used similarly as the class token: it interacts with other embeddings through self-attention, and is output by the network after the last layer. Its target objective is given by the distillation component of the loss. The distillation embedding allows the model to learn from the output of the teacher, as in a regular distillation, while remaining complementary to the class embedding. Two kinds of distillation equations have been proposed, soft distillation (Eq. 11) and hard label distillation (Eq. 12)

$$\mathcal{L}_{\mathrm{global}} = (1-\lambda)\mathcal{L}_{\mathrm{CE}}\left(\psi\left(Z_{\mathrm{s}}\right), y\right) + \lambda\tau^2 \mathrm{KL}\left(\psi\left(Z_{\mathrm{s}}/\tau\right), \psi\left(Z_{\mathrm{t}}/\tau\right)\right) \tag{11}$$

$$\mathcal{L}_{\mathrm{global}}^{\mathrm{hardDistill}} = \frac{1}{2}\mathcal{L}_{\mathrm{CE}}\left(\psi\left(Z_s\right), y\right) + \frac{1}{2}\mathcal{L}_{\mathrm{CE}}\left(\psi\left(Z_s\right), y_{\mathrm{t}}\right) \tag{12}$$

3.4. Proposed Framework

The process of classification consists of the following major steps, starting from the videos of the dataset, we extract the face frames uniformly using the LBPH face detection algorithm and then save these frames with proper

labeling conventions. Further, we create the LBP texture filter from these frames in multiple color spaces such as RGB-LBP, HSV-LBP, YCbCr-LBP, La*b*-LBP, and LBP-grayscale. Next, we feed these individual LBP color space images in the form of feature vectors to the Vision Transformer model (DeiT) for training and testing. Finally, error metrics such as Accuracy, F1 Score, EER, FAR, FRR, HTER are computed on the test data to evaluate the robustness of the created model.

Fig. 5: System Flow Diagram

4. Experimental Setup

4.1. Datasets Used

We have evaluated our model architecture on OULU-NPU dataset. The Oulu-NPU face presentation attack detection database contains 4950 real access and attack videos. The videos were shot in three sessions (Session 1, Session 2 and Session 3) with varying lighting and background environments using the front cameras of six mobile devices (Samsung Galaxy S6 edge, HTC Desire EYE, MEIZU X5, ASUS Zenfone Selfie, Sony XPERIA C5 Ultra Dual, and OPPO N3). Print and video-replay are the presentation attack types assessed in the OULU-NPU database. The attacks were carried out with the use of two printers (Printer 1 and Printer 2) and two display devices. (Display 1 and Display 2). [Fig. 5] depicts some images of real accesses and attacks captured using the Samsung Galaxy S6 edge phone. The 55 subjects' videos were separated into three subject-disjoint subsets for training, development, and testing. The Table 1 below provides a full summary of this database's partitioning.

For the evaluation of the generalization capability of the face PAD methods, four protocols are used.

4.1.1. Protocol 1

The first protocol has been designed to evaluate the generalization of face PAD approaches under previously unseen circumstances, specifically illumination and background environment.

4.1.2. Protocol 2

The second protocol has been designed to evaluate the impact of attacks performed with various printers or displays on the performance of face PAD approaches, which may suffer from new types of artifacts. By introducing a previously unseen print and video-replay attack into the test set, the effect of attack variety is evaluated.

4.1.3. Protocol 3

A Leave One Camera Out (LOCO) procedure is used to investigate the impact of input camera variation. The real and attack videos captured with five smartphones are used to train and tune the algorithms in each iteration, and the generalization of the models is examined using videos recorded with the remaining smartphones.

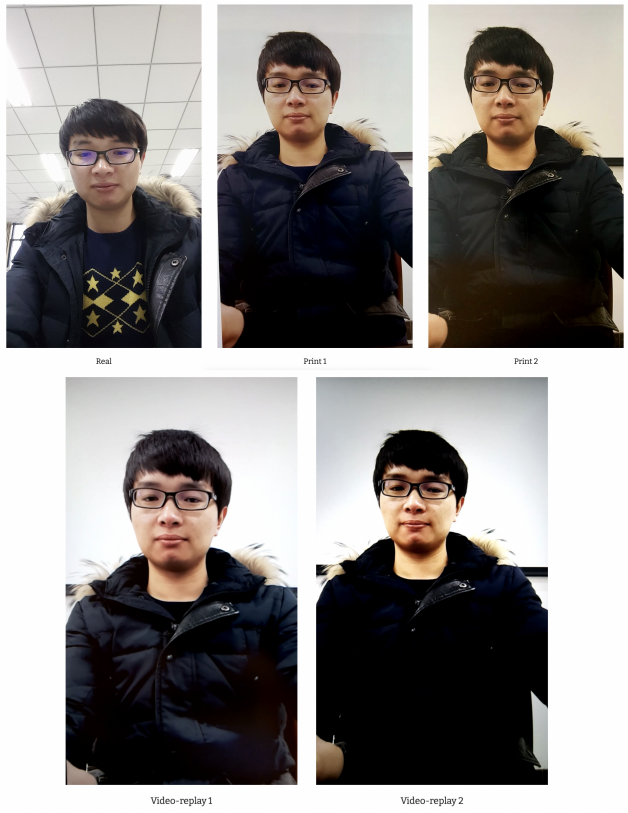

Fig. 6: Sample Images

Table 1
OuluNPU Dataset Structure

	Users	Real Access	Print Attack	Video Attack	Total
Training	20	360	720	720	1800
Development	15	270	540	540	1350
Test	20	360	720	720	1800

4.1.4. Protocol 4

In the final and most difficult protocol, all three factors are considered simultaneously, and the generalization of face PAD methods is evaluated across previously unseen environmental conditions, attacks, and input sensors.

4.2. Preprocessing

In order to prepare the data to feed into the Transformer, firstly the frames with detected faces are extracted from the video clip, and the extracted frames are labeled accordingly. Further, the frames are separated into separate files (train, dev, test), in accordance with the protocols. Next, the training data is balanced according to the classes, by undersampling the number of frames of individual videos, not compromising the variety of the dataset and the protocol. Then, LBP texture filters of all the color spaces mentioned are extracted

Table 2
OuluNPU Protocols

Protocol	Subset	Session	Phones	Users	Attacks Created Using	#real videos	#attack videos	#all videos
Protocol I	Train	Session 1, 2	6 Phones	1-20	Printer 1, 2 ; Display 1, 2	240	960	1200
	Dev	Session 1, 2	6 Phones	21-35	Printer 1, 2 ; Display 1, 2	180	720	900
	Test	Session 3	6 Phones	36-55	Printer 1, 2 ; Display 1, 2	120	480	600
Protocol II	Train	Session 1, 2, 3	6 Phones	1-20	Printer 1 ; Display 1	360	720	1080
	Dev	Session 1, 2, 3	6 Phones	21-35	Printer 1 ; Display 1	270	540	810
	Test	Session 1, 2, 3	6 Phones	36-55	Printer 2 ; Display 2	360	720	1080
Protocol III	Train	Session 1, 2, 3	5 Phones	1-20	Printer 1, 2 ; Display 1, 2	300	1200	1500
	Dev	Session 1, 2, 3	5 Phones	21-35	Printer 1, 2 ; Display 1, 2	225	900	1125
	Test	Session 1, 2, 3	1 Phone	36-55	Printer 1, 2 ; Display 1, 2	60	240	300
Protocol IV	Train	Session 1, 2	5 Phones	1-20	Printer 1 ; Display 1	200	400	600
	Dev	Session 1, 2	5 Phones	21-35	Printer 1 ; Display 1	150	300	450
	Test	Session 3	1 Phone	36-55	Printer 2 ; Display 2	20	40	60

in order to feed to the transformer for training. Finally, we evaluate the model on different error metrics.

4.2.1. Frame Extraction

We have used the LBPH Algorithm Wang and Siddique (2020) to detect the faces from the videos, every video is running at 25 fps. Cropped face frames of dimension 128x128 pixels are extracted from the video. Around 10-15 frames are extracted from each video. The labeling convention used for this is as follows '{+1/-1},{folder_number}_{video_name}_{frame_number}_{color_space}'. Here, '+1' denotes the real face, and '-1' denotes the spoofed face. Finally, these frames are separated into the train, dev, and test folders according to the protocols provided.

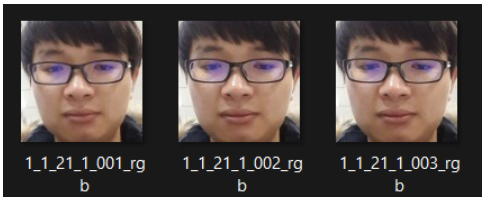

Fig. 7: Labeled extracted faces for the OuluNPU Dataset

4.2.2. Undersampling

In most of the cases, we found that the amount of training data for the spoofed image scenario was in higher proportion to the real image scenario, hence making the dataset imbalanced for training it on the transformer. The dataset is undersampled according to the classes, where we undersample the number of frames considered for every spoof scenario video, hence not compromising the protocol provided. Finally, the undersampled data is divided protocol-wise into folders according to the classes.

4.2.3. LBP Texture Extractions

Now, the LBP texture filters of the extracted frames are extracted for different color spaces namely: RGB-LBP, HSV-LBP, LA*B*-LBP, Grayscale-LBP, and YCrCb-LBP. The setup considers the radius of 1 pixel and 8 neighbors in order to extract the filters Song et al. (2013). Now the dataset is ready to be fed into the transformer for training.

4.2.4. Data Augmentations

The data is augmented with the following parameters before training: These include random horizontal and vertical flips of images with a probability of 0.5 each for increased data variability during training. Then the input image is resized to have a height and width of 256 pixels. Further, the image is center cropped with a dimension of 224 x 224 and converted to a tensor. The pixel values of the tensor image are then, normalized using the ImageNet dataset mean and standard deviation values (mean = (0.485, 0.456, 0.406), std = (0.229, 0.224, 0.225)) in order to provide better model performance.

4.3. Training

We are using the "deit_base_patch16_224" model which refers to a specific variant of the DeiT architecture, where "patch16" indicates the size of the input image patches that the model processes. In the DeiT architecture, input images are divided into non-overlapping patches of size 16x16 pixels, and each patch is then linearly embedded into a 1D vector before being processed by the Transformer layers, and "224" refers to the input image resolution, here, the model is designed to take input images of size 224x224 pixels Touvron et al. (2021). While making the observations, the model was trained on the DeiT Transformer for 50 epochs with a learning rate of 0.03. Stochastic gradient descent (SGD), was used as an optimizer for the complete training process, at the momentum of 0.5.

The augmented image is then passed through the three main components of the architecture shown in the Figure 8 in the following sequence:

4.3.1. Pretrained Transformer Layers:

This component is in control of the architecture's primary duty, picture classification. It is composed of a series of transformer layers that have been pretrained on a huge dataset of images. These layers extract features from the input image and encode them into a set of high-level representations that capture the semantic meaning of the image.

4.3.2. Custom Head:

This component is added on top of the pretrained transformer layers and is accountable for transforming the output of the transformer layers into a format that can be used for classification. The head is made up of numerous layers described in a Sequential container that are applied to the input features in a sequential order. The following is a breakdown of the several layers employed in the head:

- BatchNorm1d(768): Batch normalization Li and Lima (2021) layer that normalizes the input along the batch dimension. It takes an input size of 768, which is the number of input features coming from the backbone.
- Linear(n_inputs, 512): Fully connected layer with n_inputs (input features) as input size and 512 as output size. It performs a linear transformation on the input features.
- Mish(): Activation function called Mish, which is a variant of the popular activation function called Mish (Mish: A Self Regularized Non-Monotonic Neural Activation Function). It introduces a non-linearity to the model to capture complex patterns in the data.
- Dropout(0.2, inplace=True): Dropout layer that randomly sets a fraction of input elements to 0 during training, with a probability of 0.2. It helps in regularizing the model and preventing overfitting.
- Linear(512, 256), Linear(256, 128), Linear(128, 64): Additional fully connected layers with decreasing output sizes of 256, 128, and 64 respectively. These layers continue to apply linear transformations to the features with decreasing complexity.
- Mish(): Mish activation function is used after each of the fully connected layers.
- Linear(64, len(classes)): Final fully connected layer with an output size equal to the number of classes in the classification task. It produces the logits, which are the raw predictions for each class.

The last fully connected layer produces a set of class predictions that are used to calculate the class label for the input image. The Custom Head is trained to adapt the high-level features extracted by the Transformer Encoder to the specific task at hand. The Custom Head in this case serves to generate class predictions for a specific collection of classes.

4.3.3. Transformer Classifier:

This component is responsible for completing the final classification task. It is composed up of a single linear layer that takes the custom head's output as input and provides the final classification scores. This layer's purpose is to map the extracted features to the classes in the output space.

In summary, the proposed DeiT architecture consists of a pretrained transformer network, a custom head, and a transformer classifier that, when combined, enable high-performance image classification on a variety of datasets. The transformer layers extract features, the custom head transforms features, and the transformer classifier classifies them.

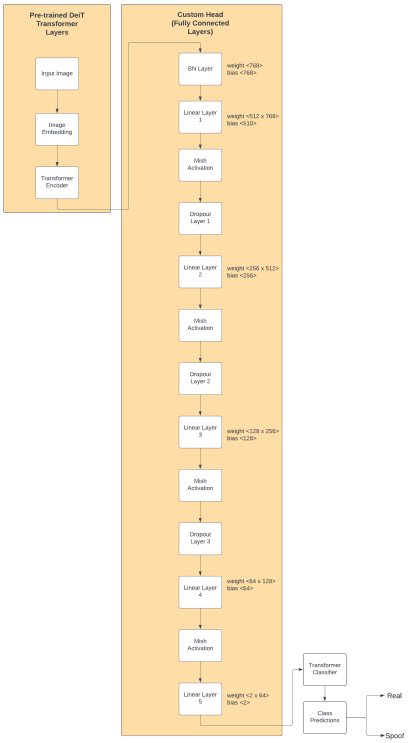

Fig. 8: Model Architecture

5. Results

The evaluation of the experiment is done on the basis of the testing accuracy on the OuluNPU Dataset, a detailed display of the testing accuracy and F1 score results for the five color spaces namely, RGB-LBP, HSV-LBP, Grayscale-LBP, YCrCb-LBP, and LA*B*-LBP is provided in the Table 3.

5.1. Metrics

For the evaluations in the Oulu-NPU dataset, we have used the ISO/IEC 30107-3 metrics Technical Committee ISO/TC JTC1 (2016), Attack Presentation Classification Error Rate (APCER), and Bonafide Presentation Classification Error Rate (BPCER) along with the Average Classification

Table 3
Test Accuracies for OuluNPU

Protocol Name	Color space	Real Detection Accuracy	Spoof Detection Accuracy	Testing Accuracies	F1 Score
Protocol 1	RGB-LBP	0.84	0.84	0.84	0.69
	HSV-LBP	0.64	0.77	0.75	0.51
	Grayscale-LBP	0.68	0.69	0.69	0.48
	YCrCb-LBP	0.83	0.72	0.74	0.57
	LA*B*-LBP	0.82	0.79	0.79	0.63
Protocol 2	RGB-LBP	0.88	0.85	0.86	0.83
	HSV-LBP	0.52	0.81	0.7	0.56
	Grayscale-LBP	0.69	0.83	0.78	0.7
	YCrCb-LBP	0.8	0.8	0.8	0.75
	LA*B*-LBP	0.77	0.77	0.77	0.72
Protocol 3	RGB-LBP	0.78 ± 0.11	0.925± 0.025	0.915± 0.025	0.775 ± 0.055
	HSV-LBP	0.46± 0.11	0.76± 0.15	0.74± 0.07	0.45 ± 0.15
	Grayscale-LBP	0.58± 0.12	0.775± 0.155	0.77± 0.07	0.595 ± 0.125
	YCrCb-LBP	0.59± 0.26	0.91± 0.06	0.86± 0.04	0.615 ± 0.135
	LA*B*-LBP	0.575± 0.165	0.845± 0.115	0.845±0.095	0.6085 ± 0.069
Protocol 4	RGB-LBP	0.84± 0.09	0.74± 0.19	0.725± 0.105	0.685 ± 0.095
	HSV-LBP	0.69± 0.18	0.735± 0.065	0.72± 0.09	0.625 ± 0.135
	Grayscale-LBP	0.665± 0.145	0.58± 0.25	0.635± 0.135	0.575 ± 0.085
	YCrCb-LBP	0.655± 0.285	0.725± 0.185	0.725± 0.065	0.615 ± 0.135
	LA*B*-LBP	0.85± 0.07	0.565± 0.135	0.645± 0.095	0.615 ± 0.065

Error Rate (ACER) in the test set. We have displayed the best values received amongst all the color spaces in the Table 4. We also obtain the ROC Curve of the model for each of the mentioned configurations. The best-produced ROC Curves for protocol 1 and 2 Boulkenafet, Komulainen, Li, Feng and Hadid (2017) are displayed below in [Fig. 9].

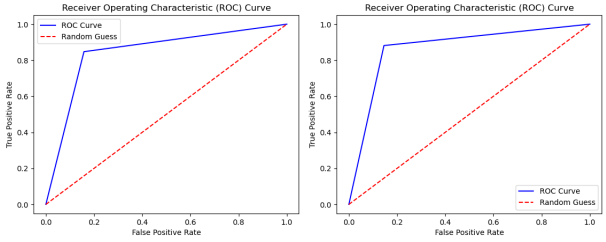

Fig. 9: ROC Curves Protocol 1[L], Protocol 2[R]

5.2. Baseline Methods

5.2.1. *LBP-SVM Ming, Visani, Luqman and Burie (2020)*

This is the most widely used algorithm in face analysis-related problems, such as face recognition, face detection, and facial expression recognition. It has several advantages, including a certain robustness toward illumination variations. This strategy is depicted in [Fig. 9]. On a normalized 64x64 image, three different LBPs were applied: $LBP^{u2}_{8,2}$, a uniform circular LBP extracted from an 8-pixel neighborhood with a 2-pixel radius, $LBP^{u2}_{16,2}$, a uniform circular LBP extracted from a 16-pixel neighborhood with a 2-pixel radius, and $LBP^{u2}_{8,1}$, a uniform circular LBP extracted from an 8-pixel neighborhood with a 1-pixel radius. Finally, a concatenation of all generated histograms forms an 833-bin/dimension histogram. This histogram is then used as a global micro-texture feature and input into a non-linear (RBF) SVM classifier for PAD.

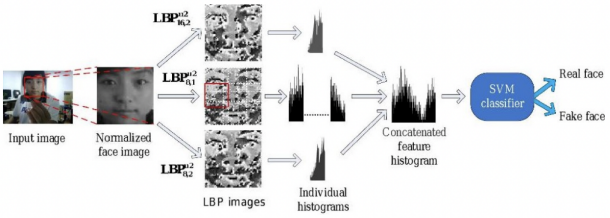

Fig. 10: LBP-SVM for Face PAD Ming et al. (2020)

5.2.2. *DeepPixBis George and Marcel (2019)*

A frame level CNN based framework which does not require temporal features for identifying presentation attacks is proposed in this section. The proposed framework uses a densely connected neural network trained using both binary and pixelwise binary supervision (DeepPixBiS). The framework uses deep pixel-wise supervision, which means that it learns to classify each pixel of the input image as real or fake, based on ground truth masks. The framework uses only frame level information, which means that it does not need any temporal or spatial information from video sequences, making it suitable for deployment in smart devices with minimal computational and time overhead.

5.2.3. *IQM-SVM Korshunov and Marcel (2018)*

The IQM-SVM method is a presentation attack detection method that uses image quality metrics (IQMs) and support vector machine (SVM). It is used to detect Deepfake videos by treating them as digital presentation attacks. The method is based on the idea that Deepfake videos have different

Table 4
Error Metrics for OuluNPU

Protocol Name	Error Metric	Values	Best Performing Color Space
Protocol 1	**APCER(%)**	15.81	RGB-LBP
	BPCER(%)	15.26	
	ACER(%)	15.54	
Protocol 2	**APCER(%)**	14.57	RGB-LBP
	BPCER(%)	11.87	
	ACER(%)	13.22	
Protocol 3	**APCER(%)**	10.2 $\pm$ 5.61	RGB-LBP
	BPCER(%)	21.945 $\pm$ 11.145	
	ACER(%)	14.56 $\pm$ 4.64	
Protocol 4	**APCER(%)**	30.535 $\pm$ 14.015	RGB-LBP
	BPCER(%)	15.145 $\pm$ 8.955	
	ACER(%)	24.935 $\pm$ 9.395	

Table 5
Comparison of the proposed method (in RGB-LBP color space) with baseline methods

Protocol Name	Method	APCER(%)	BPCER(%)	ACER(%)
Protocol 1	LBP-SVM	12.92	51.67	32.29
	IQM-SVM	19.17	30.83	25
	DeepPixBis	0.83	0	0.42
	Proposed Method	15.81	15.26	15.54
Protocol 2	LBP-SVM	30	20.28	25.14
	IQM-SVM	12.5	16.94	14.72
	DeepPixBis	11.39	0.56	5.97
	Proposed Method	14.57	11.87	13.22
Protocol 3	LBP-SVM	28.5 $\pm$ 33.05	23.33 $\pm$ 17.99	25.92 $\pm$ 11.25
	IQM-SVM	21.94 $\pm$ 9.99	21.95 $\pm$ 16.79	21.95 $\pm$ 8.09
	DeepPixBis	11.67 $\pm$ 19.57	10.56 $\pm$ 14.06	11.11 $\pm$ 9.4
	Proposed Method	10.2 $\pm$ 5.61	21.945 $\pm$ 11.145	14.56 $\pm$ 4.64
Protocol 4	LBP-SVM	41.67 $\pm$ 27.03	55.0 $\pm$ 21.21	48.33 $\pm$ 6.07
	IQM-SVM	34.17$\pm$25.89	39.17$\pm$23.35	36.67$\pm$12.13
	DeepPixBis	36.67$\pm$29.67	13.33$\pm$16.75	25.0$\pm$12.67
	Proposed Method	30.535 $\pm$ 14.015	15.145 $\pm$ 8.955	24.935 $\pm$ 9.395

image quality metrics than real videos, and it uses SVM to classify them as real or fake.

5.3. Comparison with Baseline Methods

On comparing our proposed methodology with baseline methods on the basis of the ISO/IEC 30107-3 metrics Technical Committee ISO/TC JTC1 (2016), we find that our method performs competitively for protocols 1, 2 and 3 to the baseline methods. Our method outperforms all the baseline methods for protocol 4 of the OuluNPU Dataset Boulkenafet et al. (2017) which can be seen in Table 5. The protocol 4 of the OuluNPU Dataset is the most difficult protocol which considers changes in illumination, changes in printers and display, as well as variations in camera to justify sensor interoperability together. Hence making our proposed method most feasible for different environments.

6. Conclusion

In this work, we have presented a novel methodology for Face Presentation Attack Detection (PAD). We created an architecture where we made use of the DeiT Vision Transformer Touvron et al. (2021) to train the LBP transformed spaces of the OuluNPU Dataset Boulkenafet et al. (2017)

with different color spaces. We analysed the working of the model created when different LBP color spaces were used. We were able to find that the model performs the best when trained on the RGB-LBP color space in comparison to the other color spaces mentioned above. We compared our proposed method to other baseline methodologies for Face PAD for the OuluNPU Dataset, where it was found that our method provides comparable results for the Protocols 1, 2 and 3 of the dataset and outperforms all the baseline methods for Protocol 4.

7. Future Work

Additional work can be done on the provided methodology in order to increase the accuracy of the model. One of the approaches we would like to explore with this model is by boosting it using XGBoost. Furthermore, the model can be tested on more datasets, and some cross-database testing can provide more information about the model's performance in unknown environments.

References

Almeida, W.R., Andaló, F.A., Padilha, R., Bertocco, G., Dias, W., Torres, R.d.S., Wainer, J., Rocha, A., 2020. Detecting face presentation attacks in mobile devices with a patch-based cnn and a sensor-aware loss function. PloS one 15, e0238058.

Anjos, A., Günther, M., Pereira, T., Korshunov, P., Mohammadi, A., Marcel, S., 2017. Continuously reproducing toolchains in pattern recognition and machine learning experiments.

Anjos, A., Marcel, S., 2011. Counter-measures to photo attacks in face recognition: a public database and a baseline. IAPR IEEE International Joint Conference on Biometrics doi:10.1109/IJCB.2011.6117503.

Arora, S., Bhatia, M., Mittal, V., 2021. A robust framework for spoofing detection in faces using deep learning. The Visual Computer , 1–12.

Atoum, Y., Liu, Y., Jourabloo, A., Liu, X., 2017. Face anti-spoofing using patch and depth-based cnns, in: 2017 IEEE International Joint Conference on Biometrics (IJCB), pp. 319–328. doi:10.1109/BTAS.2017.8272713.

Benlamoudi, A., Bekhouche, S.E., Maarouf, K., Bensid, K., Ouahabi, A., Hadid, A., taleb ahmed, A., 2022. Face presentation attack detection using deep background subtraction. Sensors 22. doi:10.3390/s22103760.

Boulkenafet, Z., Komulainen, J., Hadid, A., 2015. Face anti-spoofing based on color texture analysis, in: 2015 IEEE International Conference on Image Processing (ICIP), pp. 2636–2640. doi:10.1109/ICIP.2015.7351280.

Boulkenafet, Z., Komulainen, J., Li, L., Feng, X., Hadid, A., 2017. Oulunpu: A mobile face presentation attack database with real-world variations, in: 2017 12th IEEE International Conference on Automatic Face Gesture Recognition (FG 2017), pp. 612–618. doi:10.1109/FG.2017.77.

Brownlee, J., 2019. A gentle introduction to deep learning for face recognition. Machine Learning Mastery .

Chingovska, I., Anjos, A., Marcel, S., 2012. On the effectiveness of local binary patterns in face anti-spoofing.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., 2021. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv:2010.11929.

Erdogmus, N., Marcel, S., 2013. Spoofing in 2d face recognition with 3d masks and anti-spoofing with kinect, in: 2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS), pp. 1–6. doi:10.1109/BTAS.2013.6712688.

George, A., Marcel, S., 2019. Deep pixel-wise binary supervision for face presentation attack detection. arXiv:1907.04047.

George, A., Marcel, S., 2021a. Cross modal focal loss for rgbd face anti-spoofing. arXiv:2103.00948.

George, A., Marcel, S., 2021b. On the effectiveness of vision transformers for zero-shot face anti-spoofing, in: 2021 IEEE International Joint Conference on Biometrics (IJCB), pp. 1–8. doi:10.1109/IJCB52358.2021.9484333.

George, A., Mostaani, Z., Geissenbuhler, D., Nikisins, O., Anjos, A., Marcel, S., 2019. Biometric face presentation attack detection with multi-channel convolutional neural network. IEEE Transactions on Information Forensics and Security PP, 1–1. doi:10.1109/TIFS.2019.2916652.

Gragnaniello, D., Poggi, G., Sansone, C., Verdoliva, L., 2015. An investigation of local descriptors for biometric spoofing detection. IEEE Transactions on Information Forensics and Security 10, 849–863. doi:10.1109/TIFS.2015.2404294.

Gu, F., Xia, Z., Fei, J., Yuan, C., Zhang, Q., 2020. Face spoof detection using feature map superposition and cnn. International Journal of Computational Science and Engineering 22, 355. doi:10.1504/IJCSE.2020.107356.

Hashemifard, K., Akbari, M., 2021. A compact deep learning model for face spoofing detection.

Joshi, S., Verma, D., Saxena, G., Paraye, A., 2019. Issues in Training a Convolutional Neural Network Model for Image Classification. pp. 282–293. doi:10.1007/978-981-13-9942-8_27.

Komulainen, J., Hadid, A., Pietikäinen, M., 2011. Face spoofing detection from single images using micro-texture analysis. doi:10.1109/IJCB.2011.6117510.

Korshunov, P., Marcel, S., 2018. Deepfakes: a new threat to face recognition? assessment and detection.

Li, B., Lima, D., 2021. Facial expression recognition via resnet-50. International Journal of Cognitive Computing in Engineering 2. doi:10.1016/j.ijcce.2021.02.002.

Li, J., Tan, T., Jain, A., 2004. Live face detection based on the analysis of fourier spectra. Proceedings of SPIE - The International Society for Optical Engineering 5404, 296–303. doi:10.1117/12.541955.

Li, L., Correia, P., Hadid, A., 2017. Face recognition under spoofing attacks: Countermeasures and research directions. IET Biometrics 7. doi:10.1049/iet-bmt.2017.0089.

Liu, Y., Stehouwer, J., Jourabloo, A., Liu, X., 2019. Deep tree learning for zero-shot face anti-spoofing, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

Mahitha, 2018. Face spoof detection using machine learning with colour features.

Ming, Z., Visani, M., Luqman, M.M., Burie, J.C., 2020. A survey on anti-spoofing methods for facial recognition with rgb cameras of generic consumer devices. Journal of Imaging 6, 139. URL: http://dx.doi.org/10.3390/jimaging6120139, doi:10.3390/jimaging6120139.

Nikisins, O., George, A., Marcel, S., 2019. Domain adaptation in multi-channel autoencoder based features for robust face anti-spoofing. arXiv:1907.04048.

Paul, S., Chen, P.Y., 2021. Vision transformers are robust learners.

Peng, F., Meng, S.h., Long, M., 2022. Presentation attack detection based on two-stream vision transformers with self-attention fusion. J. Vis. Comun. Image Represent. 85. URL: https://doi.org/10.1016/j.jvcir.2022.103518, doi:10.1016/j.jvcir.2022.103518.

Raghu, M., Unterthiner, T., Kornblith, S., Zhang, C., Dosovitskiy, A., 2021. Do vision transformers see like convolutional neural networks? CoRR abs/2108.08810. URL: https://arxiv.org/abs/2108.08810, arXiv:2108.08810.

Sharma, D., Selwal, A., 2023. A survey on face presentation attack detection mechanisms: hitherto and future perspectives. Multimedia Systems , 1–51doi:10.1007/s00530-023-01070-5.

Siddiqui, T., Bharadwaj, S., Dhamecha, T., Agarwal, A., Vatsa, M., Singh, R., Ratha, N., 2016. Face anti-spoofing with multifeature videolet aggregation, pp. 1035–1040. doi:10.1109/ICPR.2016.7899772.

Song, K.C., YAN, Y.H., CHEN, W.H., Zhang, X., 2013. Research and perspective on local binary pattern. Acta Automatica Sinica 39, 730–744. doi:10.1016/S1874-1029(13)60051-8.

Tan, X., Li, Y., Liu, J., Jiang, L., 2010. Face liveness detection from a single image with sparse low rank bilinear discriminative model. ECCV (6) 6316, 504–517.

Technical Committee ISO/TC JTC1, Information technology, S.S..B., 2016. iso/iec 30107-1:2016. URL: https://www.iso.org/obp/ui/#iso:std:iso-iec:30107:-1:ed-1:v1:en.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H., 2021. Training data-efficient image transformers distillation through attention. arXiv:2012.12877.

Tu, X., Fang, Y., 2017. Ultra-deep neural network for face anti-spoofing, in: Neural Information Processing: 24th International Conference, ICONIP 2017, Guangzhou, China, November 14-18, 2017, Proceedings, Part II 24, Springer. pp. 686–695.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need. URL: https://arxiv.org/pdf/1706.03762.pdf.

Wang, L., Siddique, A.A., 2020. Facial recognition system using lbph face recognizer for anti-theft and surveillance application based on drone technology. Measurement and Control 53, 1070–1077. URL: https://doi.org/10.1177/0020294020932344, doi:10.1177/0020294020932344, arXiv:https://doi.org/10.1177/0020294020932344.

Zhang, Y., Zhao, M., Yan, L., Gao, T., Chen, J., 2020. Cnn-based anomaly detection for face presentation attack detection with multi-channel images, in: 2020 IEEE International Conference on Visual Communications and Image Processing (VCIP), pp. 189–192. doi:10.1109/VCIP49819.2020.9301818.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., He, Q., 2021. A comprehensive survey on transfer learning. Proceedings of the IEEE 109, 43–76. doi:10.1109/JPROC.2020.3004555.

Highlights

- A methodology for detecting face spoofing attacks in facial recognition systems using the recently developed vision transformer model.
- Evaluated the LBP + ViT model's performance using biometric error evaluation methods on face presentation attack detection (PAD) data
- The provided methodology provides more robust results in comparison to other state of the art methods for face PAD.
- Provided framework is expected to perform better in variable conditions.