A Unified Model for Face Matching and Presentation Attack Detection using an Ensemble of Vision Transformer Features

Rouqaiah Al-Refai and Karthik Nandakumar
Mohamed Bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE
rouqaiah.al-refai@mbzuai.ac.ae, karthik.nandakumar@mbzuai.ac.ae

Abstract

A typical automated face recognition system is composed of three main component tasks: face detection and alignment (FDA), face presentation attack detection (FPAD), and face representation and matching (FRM). These tasks are often treated as standalone problems and deep neural network (DNN)-based solutions have been proposed to address them individually. However, in resource-constrained scenarios it would be ideal to have a unified DNN model that can perform all the three tasks together. As a first step towards realizing this goal, this work attempts to perform joint FRM and FPAD based on a single Vision Transformer (ViT) backbone. Recent work demonstrating the ability of ViT to extract a diverse set of feature representations gives rise to the tantalising possibility of building an end-to-end face recognition system using a single ViT model. The standard approach for designing multi-task DNNs is to implement different classification heads (e.g., for FRM and FPAD) based on a common stem/base and learn these heads either individually or jointly. A key contribution of this work is to demonstrate that this naive multi-head approach results in sub-optimal performance for either FRM or FPAD, because the features required by these tasks are very different. While good FPAD performance depends on accurately characterizing the micro textures, face matching demands attention towards more global characteristics. Hence, we propose a novel feature ensemble approach, where an ensemble of local features extracted from the intermediate blocks of a ViT are utilized for FPAD, while face matching is performed based on the ViT class token. Experiments demonstrate that the proposed ViT feature ensemble approach is able to achieve good performance for both face matching and FPAD compared to the multi-head approach.

1. Introduction

Face recognition is one of the challenging applications of computer vision and face recognition systems are widely used for authentication and access control purposes in personal devices as well as border security and surveillance applications [20]. A face recognition system often consists of multiple modules stacked together to accomplish an end-to-end system. Typically, a face recognition system starts with a *face detector* that is capable of detecting the existence of faces in images or videos and identifying key points (landmarks) in the detected faces [54]. These landmarks are used to canonically align faces before they are matched. The *face matcher* extracts features from the aligned faces and computes a similarity metric that signifies whether two faces belong to the same identity or not [47] [36]. Given the predominant use of face recognition for authentication and access control tasks, there is a strong incentive for malicious users to fool the face recognition system by presenting a pre-captured photo/video or by wearing a mask/accessory. Such physical attacks on the camera sensor are known as presentation or spoof attacks [28]. To counter the vulnerability of face recognition systems to presentation attacks, a *face presentation attack detector* is usually applied before feeding the faces to the face matching module [47].

All the three key tasks in a face recognition system, namely, face detection and alignment (FDA), face representation and matching (FRM), and face presentation attack detection (FPAD) are usually considered as standalone problems. Over the last five decades, methods for solving these tasks have evolved from using traditional hand-crafted feature extraction approaches [5, 23] to the use of deep convolutional neural networks (CNN) [38, 33]. The use of multiple deep neural network (DNN) models within the face recognition pipeline creates an implementation challenge in resource-constrained settings such as edge devices. Given that all the three tasks are somewhat correlated, it is desirable to have a unified model that jointly performs these tasks. To the best our knowledge, the challenge of solving all the three tasks simultaneously and designing an end-to-end face recognition system using a single machine learning model has never been addressed in the literature. Towards achieving this goal, this work aims to design a unified DNN model for joint FRM and FPAD. Similar attempts have been

made in the biometrics literature, both in the case of face [53] and fingerprint [35] modalities, where multi-task CNN architectures have been proposed.

Transformer networks [43] use the concept of self-attention to robustly learn global relationships between individual elements of the input space. Exploiting this idea, Vision Transformer (ViT) [14] models have been successfully developed for various computer vision tasks including image classification [14], segmentation [39] and object detection [9]. Since ViTs have shown the ability to extract diverse features from images and jointly address multiple tasks [6, 31], we argue that ViT is an ideal choice to be the core backbone of a face recognition system. A simple approach to design a multi-task ViT model is by implementing individual classification heads for each task on top of a shared ViT-based feature extraction backbone. However, this approach is likely to fail when the tasks are not entirely complementary. For example, the FRM task typically relies on the extraction of global semantic features from the face image for achieving good face matching accuracy. Such features are usually extracted by the deeper layers/blocks of a ViT model. On the other hand, the clues required for FPAD often lie in the micro texture features, which are extracted by the initial ViT blocks. Thus, the main contributions of this work are two-fold: (i) Develop a *multi-task ViT model for joint FRM and FPAD*, and (ii) Propose a novel *feature ensemble* approach that uses local representations extracted by the intermediate ViT blocks to perform FPAD and the global features to perform face matching.

2. Related work

2.1. Face Representation and Matching

Face recognition systems have evolved over the years from the use of holistic methods [5, 18], handcrafted features [23, 2], and local descriptors [8] to the use of deep learning-based models. While earlier methods suffered from the inability to handle large intra-user variations, CNN-based methods [41, 40, 33, 38] have proven to be more successful in handling this problem. Apart from the network architecture, the importance of loss functions used to train the CNN models has also been investigated in detail leading to formulations such as SphereFace [24], CosFace [45], and ArcFace [12]. Finally, Zhong and Deng [55] have employed Vision Transformer (ViT) models for face matching and showed that it achieves state-of-the-art (SOTA) performance on large-scale datasets.

2.2. Face Presentation Attack Detection

Several techniques have been proposed for FPAD over the years and most of these methods exploit the fact that presentation attack instruments (PAI) can cause image quality degradation during recapture, which can be leveraged to distinguish attacks from bonafide samples [15]. Handcrafted texture-based features [29, 10, 22, 7, 42] have poor generalizability across different PAIs, camera devices, and illumination conditions. While dynamic features extracted from videos have been proposed for FPAD [32, 3], this approach requires user cooperation and degrades the usability and throughput of face recognition systems [1].

Deep learning (especially CNN-based) methods have shown good detection performance on various FPAD benchmarks [51, 4]. Liu et al. [26] used a CNN and recurrent neural network (RNN) combination model for FPAD, which employs auxiliary supervision utilizing depth and temporal features to help alleviate the problem of poor generalization. George et al. [15] introduced a frame-level CNN-based framework trained with binary and pixel-wise binary supervision instead of using synthesized depth values. Several multi-channel (e.g., color, depth, infrared, etc.) approaches have been introduced as a solution for handling different types of attacks [44, 17]. While the multi-channel approach makes it possible to achieve good FPAD performance, the increased hardware cost for the additional channels limits their adoption in real-world systems. One of the main issues with deep learning-based FPAD methods is the limited availability of training data, because it is expensive to collect a diverse set of presentation attacks. To mitigate this issue, Liu et al. [27] proposed a Deep Tree Network architecture trained in an unsupervised manner to partition the spoof samples into semantically meaningful subgroups. George and Marcel [16] introduced a simple pre-trained ViT-based FPAD framework and investigated its effectiveness for the problem of zero-shot FPAD.

2.3. Multi-task Vision Transformers

ViTs have been employed for jointly performing multiple tasks. The naive approach is to use ViT backbone as a feature extractor and add individual heads/classifiers for each task. The limitation of the naive approach is the need for a large-scale dataset that can be used for jointly training the network for multiple tasks and achieve good performance for all of them. In [6], a single end-to-end transformer framework was proposed to jointly learn multiple vision tasks using a shared attention mechanism to model dependencies between tasks. An alternative approach is to learn multiple class tokens for different tasks. For instance, Naseer et al. [31] introduced a shape token that learns to focus on shape representations in images in addition to the original class token that learns texture related features.

It is also well known that features extracted by different blocks of a ViT encode complementary information [49, 30]. Specifically, the deep ViT features mainly focus on global information, while the local information gets encapsulated in the low-level or mid-level tokens generated by the initial transformer blocks. This was leveraged in [46]

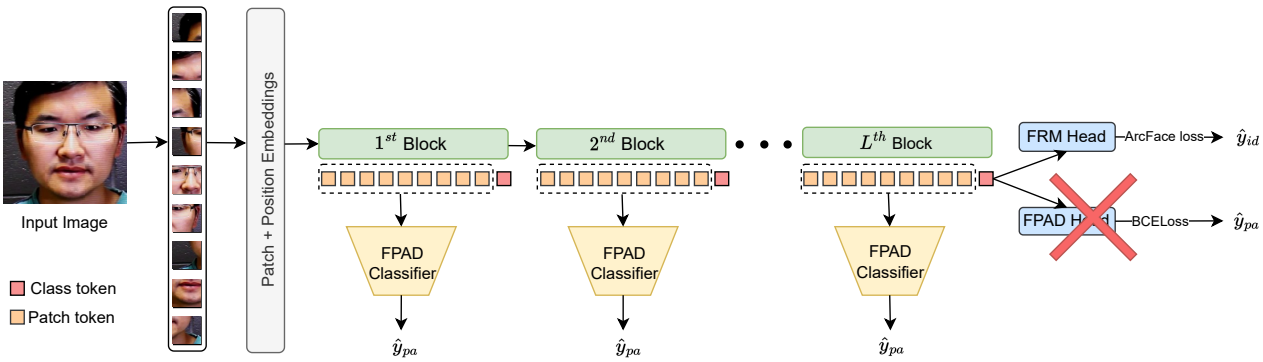


Figure 1. Illustration of proposed ViT feature ensemble framework for joint face recognition (FRM) and presentation attack detection (FPAD). Unlike the traditional approach of adding multiple heads after a shared feature extractor, the proposed method allows the ViT class token to focus on the global face information required for FRM and utilizes the micro texture features encapsulated by the patch tokens of intermediate ViT blocks for FPAD. The representative face image is from the SiW dataset [26].

to design a cross-layer relation-aware attention mechanism that achieves good generalization performance on the FPAD task using ViT. Despite these advancements, the problem of designing a deep learning model (such as CNN or ViT) that can jointly perform matching and PAD has been seldom addressed in the biometrics literature. The only known attempts are [53] for face and [35] for fingerprint, both of which follow the naive approach of using a shared CNN backbone for feature extraction and dual heads for matching and PAD. In the absence of large-scale training sets, this approach is unlikely to work well in practice.

3. Proposed Method

The overall objective of this work is to design a multi-task ViT model that achieves high accuracy for both FRM and FPAD. To achieve this goal, we start with a vanilla ViT model [14] pre-trained on the ImageNet dataset [11] as the backbone, and evaluate two different architectures: (i) a dual head approach that learns different classification heads for FPAD and FRM based on the shared ViT features, and (ii) a feature ensemble approach that utilizes different features within the ViT pipeline for FPAD and FRM.

3.1. Vanilla ViT Backbone

The vanilla ViT model is composed of L blocks, where each block in turn consists of multi-head self attention (MHSA) and multi-layer perceptron (MLP) layers together with normalization layers and residual connections [48]. The vanilla ViT model partitions an input image $\boldsymbol{x} \in \mathbb{R}^{h\times w\times t}$ (where h, w, and t represent the height, width, and number of channels in the image, respectively) into n non-overlapping patches, flattens each patch into a vector using a linear projection layer, and adds an one dimensional position embedding to each patch representation to maintain the spatial information of the patches. A learnable [*class*] token [13] is also concatenated to the sequence of patches to distill the knowledge learned. Thus, the input to the first ViT block can be denoted as $\boldsymbol{Z}_0 = [\boldsymbol{z}_{0,1}, \boldsymbol{z}_{0,2}, \cdots, \boldsymbol{z}_{0,n}, \boldsymbol{z}_{0,c}] \in \mathbb{R}^{(n+1)\times d}$, where the first n components represent the *patch tokens* and the last component $\boldsymbol{z}_{0,c}$ denotes the *class token*.

The sequence of tokens is processed through the self-attention mechanism $\mathcal{A}$ to learn the relationship between the patches. For an input $\boldsymbol{Z} \in \mathbb{R}^{(n+1)\times d}$, the MHSA layer outputs a concatenation of k parallel self-attention heads, i.e., $MHSA(\boldsymbol{Z}) = [\mathcal{A}_1(\boldsymbol{Z}), \mathcal{A}_2(\boldsymbol{Z}), \ldots, \mathcal{A}_k(\boldsymbol{Z})]$, where $\mathcal{A}_j(\boldsymbol{Z}) = softmax(\boldsymbol{Q}_j\boldsymbol{K}_j^T/\sqrt{d})\boldsymbol{V}_j$, $\boldsymbol{Q}_j/\boldsymbol{K}_j/\boldsymbol{V}_j = \boldsymbol{Z}\boldsymbol{W}^j_{Q/K/V}$, $\boldsymbol{W}^j_{Q/K/V} \in \mathbb{R}^{d\times\frac{d}{k}}$, and $j = 1, 2, \cdots, k$. Let $\boldsymbol{Z}_\ell = [\boldsymbol{z}_{\ell,1}, \boldsymbol{z}_{\ell,2}, \cdots, \boldsymbol{z}_{\ell,n}, \boldsymbol{z}_{\ell,c}] \in \mathbb{R}^{(n+1)\times d}$ denote the output of the ℓ-th ViT block, $\ell = 1, 2, \cdots, L$. Typically, the class token output by the last block $\boldsymbol{z}_{L,c}$ is considered as the final embedding of the given input image. Thus, the ViT backbone can be considered as an encoder (E_ω) that maps an input image $\boldsymbol{x}$ to its latent representation (class token) $\boldsymbol{z}_{L,c} \in \mathbb{R}^d$, i.e., $\boldsymbol{z}_{L,c} = E_\omega(\boldsymbol{x})$.

3.2. Dual Head Architecture for FRM and FPAD

In this approach, there are no changes to the ViT backbone, which acts as a shared feature extractor. On top of this backbone, two task-specific classifier heads, denoted as H_η and G_ϕ, are added for the FPAD and FRM tasks, respectively. The class token embedding $\boldsymbol{z}_{L,c}$ obtained from the last ViT block is passed on to FPAD classification head H_η to predict whether the presented input image is bonafide or attack, i.e., $\hat{y}_{pa} = H_\eta(\boldsymbol{z}_{L,c}) \in \{0, 1\}$. Here, the labels 0 and 1 represent the attack and bonafide presentations, respectively. Given a set of N_{pa} training samples $\{\boldsymbol{x}_i, y^i_{pa}\}_{i=1}^{N_{pa}}$, where $y^i_{pa} \in \{0, 1\}$ denotes the ground truth attack label for the i-th training sample, the FPAD head can be trained by minimizing the following binary cross-entropy loss ($\mathcal{L}_{FPAD}$).

$$\mathcal{L}_{FPAD} = -\frac{1}{N_{pa}} \sum_{i=1}^{N_{pa}} \left(y_{pa}^i \log(\hat{y}_{pa}^i) + (1 - y_{pa}^i) \log(1 - \hat{y}_{pa}^i) \right). \tag{1}$$

Similarly, the class token is passed to the FRM head G_ϕ, which predicts the identity of the person $\hat{y}_{id} = G_\phi(\boldsymbol{z}_{L,c}) \in \{1, 2, \cdots, C\}$. Here, C denotes the number of unique identities in the training dataset. Given a set of N_{id} training samples $\{\boldsymbol{x}_i, y_{id}^i\}_{i=1}^{N_{id}}$, where $y_{id}^i \in \{1, 2, \cdots, C\}$ denotes the ground truth identity label for the i-th training sample, the following ArcFace [12] loss function is used for supervising the training of the FRM head.

$$\mathcal{L}_{FRM} = -\frac{1}{N_{id}} \sum_{i=1}^{N_{id}} \log \frac{e^{s \cos \tilde{\theta}_{y_{id}^i}}}{e^{s \cos \tilde{\theta}_{y_{id}^i}} + \sum_{j=1, j \neq y_{id}^i}^{C} e^{s \cos \theta_j}}, \tag{2}$$

where $\tilde{\theta}_r = (\theta_r + m)$, $r \in \{1, 2, \cdots, C\}$, s represents the scale parameter, and m is the angular margin parameter in the ArcFace loss. Furthermore, $\theta_r = \arccos(\|\boldsymbol{W}_r\| \|\boldsymbol{v}\|)$, where $\|\cdot\|$ represents L2 norm, $\boldsymbol{v} \in \mathbb{R}^{\tilde{d}}$ is the face embedding produced by the penultimate layer of the network G_ϕ, and $\boldsymbol{W}_r$ denotes the r-th column of the weight matrix $\boldsymbol{W} \in \mathbb{R}^{\tilde{d}} \times C$ corresponding to the softmax [25] layer. During face verification, the cosine similarity between a pair of face embeddings ($\boldsymbol{v}$) is used as the similarity score to determine whether the pair of images belong to the same identity.

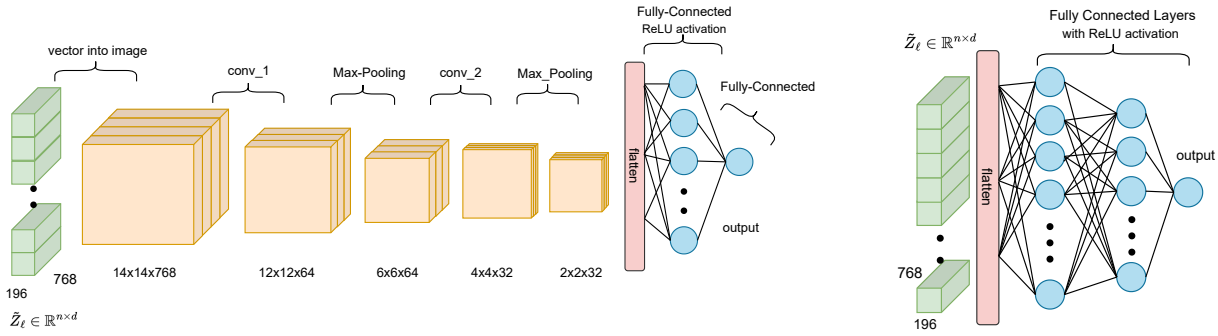
Training Strategies: The dual head architecture can be trained in a number of ways. Firstly, the weights of the ImageNet pre-trained ViT backbone can be frozen (fixed ω) and only the FPAD and FRM heads can be learned individually using equations 1 and 2 to obtain η and ϕ, respectively. We refer to this scenario as "Zero Shot" approach. Secondly, the whole architecture can be trained for one of the tasks (while the task-specific head is initialized randomly and learned from scratch, the ViT backbone starts with the pre-trained weights and gets finetuned), the backbone can be frozen, and only the head for the other task can be learned subsequently. In this case, the ViT backbone is optimized for the first task and could be potentially sub-optimal for the second task. This scenario is referred to as "Task Finetuning", which can be further categorized into "FRM Finetuning" and "FPAD Finetuning" depending on whether FRM or FPAD task is learned first. In the above training scenarios, the datasets used for learning the FRM and FPAD tasks can be completely independent. When a sufficiently large dataset containing both multiple bonafide samples for each identity as well as attack samples for those identities is available, it is possible to learn the FRM and FPAD heads in parallel (referred to as "Joint Finetuning - Heads") or add up the $\mathcal{L}_{FPAD}$ and $\mathcal{L}_{FRM}$ losses to jointly finetune the whole architecture (referred to as "Joint Finetuning - Whole").

3.3. Feature Ensemble Architecture

Typically, ViTs rely only on the class token at the last block $\boldsymbol{z}_{L,c}$ for image classification, while ignoring the rest of the learned features (intermediate class and patch tokens). Our hypothesis is that learning global face information is critical for the FRM task, while micro-texture features are important for the FPAD task. Attempting to distill both these types of information into the same class token is likely to lead to sub-optimal performance for one of the tasks. A potential solution is to add another class token and distill different types of information into the two class tokens [31]. However, this approach requires modifications to the ViT backbone and training the modified backbone. Instead, we propose a simpler alternative based on the insight that micro features required for the FPAD task are captured by the patch tokens of initial ViT blocks. Hence, features extracted from these patch tokens can be used for the FPAD task, while the ViT class token focuses on the FRM task.

The proposed feature ensemble architecture appends a classification branch (denoted as F_ψ) to an intermediate block of the ViT. This classifier takes the patch tokens generated by the block $\tilde{\boldsymbol{Z}}_\ell = [\boldsymbol{z}_{\ell,1}, \boldsymbol{z}_{\ell,2}, \cdots, \boldsymbol{z}_{\ell,n}] \in \mathbb{R}^{n \times d}$ (note that the intermediate class token $\boldsymbol{z}_{\ell,c}$ is ignored) as input and outputs a binary classification decision $\hat{y}_{pa} \in \{0, 1\}$. Two network architectures have been considered for the classifier F_ψ (see Figure 2). While the first architecture is a multi-layer perceptron (MLP), the second one is styled based on a CNN. The MLP version of F_ψ involves three fully connected (FC) layers, with ReLU activation function after the first two layers for faster and reliable convergence, and sigmoid activation at the third (output) layer. The CNN version of F_ψ is composed of two convolutional layers interspersed with ReLU activation and max-pooling operations and two FC layers for final prediction. This classifier reshapes the input $\tilde{\boldsymbol{Z}}_\ell$ into a $(n_1 \times n_2 \times d)$ tensor where $(n_1 \times n_2) = n$, uses convolutional kernels of size (3×3) with stride of 2, and performs max-pooling over windows of size (2×2). The output of the second convolutional layer is flattened before being passed through the FC layers. Both the MLP and CNN versions of F_ψ are evaluated in our experiments to determine the better design.

The classification branch in the feature ensemble method is trained as follows. The ViT backbone E_ω with the FRM head $G\phi$ is first finetuned using only the $\mathcal{L}_{FRM}$ loss (similar to "FRM Finetuning" in the dual head scenario). Next, the backbone is frozen, $\tilde{\boldsymbol{Z}}_\ell$ is extracted from an intermediate layer, and only the classifier F_ψ is trained using the $\mathcal{L}_{FPAD}$ loss. Thus, independent datasets can be used for learning the two tasks. Moreover, the FPAD classifier is a byproduct of the FRM model and FPAD training has no impact on the FRM performance. It is also possible to append FPAD classifiers to more than one intermediate block in the ViT and fuse their outputs to obtain the final prediction $\hat{y}_{pa}$.



(a) Convolution (b) MLP

Figure 2. Two classifier designs, (a) CNN and (b) MLP, have been considered for the FPAD task in the feature ensemble approach, which utilizes the patch tokens produced by the intermediate ViT blocks $\tilde{\boldsymbol{Z}}_\ell$ for FPAD.

4. Experimental Results

4.1. Datasets

For most experiments, we use the CASIA-WebFace [52] dataset for learning the FRM task, the LFW [19] dataset for benchmarking the FRM accuracy, and the training and test partitions of SiW [26] dataset for learning the FPAD task and evaluating the FPAD performance. Only in the case of joint training of FRM and FPAD tasks, the training partition of SiW dataset is used for both FRM and FPAD learning.

CASIA-WebFace [52] consists of 10,575 real identities, with each subject having around 46 images of size 256×256. Since it is collected from the Internet, it covers a wide range of pose, illumination, and expression. The data is pre-processed by detecting faces using MTCNN [54], cropping the face region, and resizing the images to 224×224.

LFW Labeled Faces in the Wild [19] is a database that consists of 13,233 face images corresponding to 5,794 different identities collected from the Internet. It is a widely used public benchmark for face verification. Following the LFW standard evaluation protocol [12], 6,000 image pairs are randomly generated from the dataset and used for reporting face recognition testing accuracy. Data pre-processing on LFW is similar to that of the CASIA dataset.

SiW Spoof in the Wild dataset [26] consists of bonafide and presentation attack videos collected from 165 subjects. There are 4,478 videos in total, with each subject having 8 live (bonafide) videos and up to 20 spoof (presentation attack) videos. The live videos are captured with different variations of pose, expression, illumination, and distance. The spoof videos include presentation attack instruments such as replay and printed paper. The videos in this dataset have a 1080p HD resolution. These videos were processed using the face bounding box files that come with the dataset, which contain the face coordinates for each corresponding video. These coordinates were used to crop faces from selected frames and the cropped faces are resized to a resolution of 224×224 to be compatible with the input size accepted by the ViT model.

Three specific evaluation protocols have been defined for the SiW dataset to benchmark the FPAD generalization performance under different scenarios. In line with these defined evaluation protocols, the subjects are partitioned into train and test sets, with 90 and 75 participants, respectively. However, the focus of this work is not to evaluate the generalization of FPAD methods, but to study the interplay between FPAD performance and face matching accuracy when the two tasks are carried out using a unified model. Since this dataset is also used for joint training of FRM and FPAD tasks, the frame selection process needs to be compatible with both FRM and FPAD requirements. Hence, we do not follow any of the three prescribed protocols for frame selection. Instead, a collection of 10 frames per video are chosen randomly to form the training and test sets. Since no validation set was provided with the SiW dataset, the train set was split identity-wise into train and validation sets using an 80%-20% ratio. Thus, the train set has 72 identities and the validation set has 18 subjects with no overlap of identities between the two sets.

4.2. Experimental Setup

ViT Backbone We use the base variation of ViT from the open-source implementation provided in [50], which has $L = 12$ blocks. Since the patch size is 16×16, the number of patch tokens is $n = 196 = (14 \times 14)$ and dimensionality of each patch is $d = 768$. The ViT backbone is initialized with ImageNet 21k classes pre-trained weights. For FRM finetuning using the CASIA-WebFace dataset, the whole model is finetuned with the ArcFace [12] loss ($s = 32$ and $m = 0.5$) and updated using the stochastic gradient descent

(SGD) [37] algorithm with a fixed learning rate of 10^{-3}. We use a batch size of 256 and train the network for 40 epochs. Then, the best model is selected based on the best LFW testing accuracy obtained after each training epoch. On the the other hand, for FPAD finetuning using the SiW dataset, the whole model is trained using the binary cross-entropy loss and updated using the Adam [21] optimizer with a learning rate of 10^{-4}. The model is trained for 20 epochs using 16 samples per batch and data augmentation (random horizontal flipping with probability 0.5) is employed. The models are implemented using PyTorch [34] and all experiments are carried out on a single Nvidia RTX A6000 GPU with 40 GB memory.

FRM and FPAD Classifiers The FPAD classification head H_η has one FC (linear) layer with sigmoid activation. The FRM classification head G_ϕ has one hidden FC layer that converts the d-dimensional class token into a $\tilde{d} = 512$ dimensional face embedding before the ArcFace loss is computed. These two task-specific classification heads are updated using the Adam optimizer with a learning rate of 10^{-3}. The FPAD classification branch F_ψ in the feature ensemble architecture has two possible designs as discussed in Section 3.3. We append a separate FPAD classification branch after each ViT block, resulting in $L = 12$ separate classifiers. All these classifiers are trained with the same hyperparameter settings to allow for easy comparison. Irrespective of whether CNNs or MLPs are used, the FPAD classifiers are supervised with binary cross-entropy loss and their parameters are updated using Adam optimizer with a learning rate of 10^{-5}. These networks are trained with batch size of 32 for 30 epochs and data augmentation (random horizontal flip with probability 0.5) is used.

Evaluation Metrics The FPAD task is evaluated using the accuracy metric as well as standard ISO/IEC 30107-3 metrics such as Attack Presentation Classification Error Rate (APCER), Bonafide Presentation Classification Error Rate (BPCER), and Average Classification Error Rate (ACER). A score threshold is calculated based on the validation set and used for reporting results on the test set. The FRM task is evaluated based on the face matching accuracy, which is computed by performing face verification on the LFW image pairs.

5. Results and Discussion

Single-task ViTs The first set of experiments is used to evaluate the ability of ViT model to perform FPAD and FRM tasks individually. Table 1 compares the face matching accuracy of the ViT model for the zero shot FRM scenario (with ImageNet pre-trained ViT as feature extractor) and for the case when the whole model is finetuned using the large-scale CASIA dataset. Finetuning the ViT on a large-scale face dataset is clearly beneficial, leading to more than 10% improvement in the matching accuracy compared to the zero shot case. Furthermore, the better performance of the finetuned ViT compared to the Face Transformer model proposed in [55] shows that starting with the ImageNet initialized weights and finetuning the model is a more prudent strategy than training the ViT from scratch as done in [55]. The FPAD performance of ViT is also enhanced by finetuning on the FPAD task. For example, when trained on the SiW dataset, the finetuned ViT model achieves an order of magnitude lower ACER compared to the zero-shot scenario as shown in Table 2.

Table 1. Face matching accuracy on the LFW [19] benchmark dataset for various ViT models (with vanilla architecture) trained for the FRM task using the CASIA-WebFace dataset [52].

Training Scenario	Accuracy (%)
Zero Shot	88.70
FRM Finetuning	**98.96**
Face Transformer [55]	97.42

Dual Head ViTs The second set of experiments is focused on benchmarking the multi-task ViT models with dual head architecture trained using the various training strategies outlined in Section 3.2. It can be observed from Table 3 that none of the dual head models achieve good accuracy on both the tasks. While FRM finetuning results in an optimal model for the FRM task, it degrades the FPAD accuracy by more than 4%. In contrast, FPAD finetuning severely degrades the face matching accuracy (by more than 30%), while achieving the best FPAD accuracy. This shows that while the FRM and FPAD tasks are related, the features required for these tasks are very different. The representations learned for the FRM task are not discriminative enough for FPAD and vice versa. This confirms our hypothesis that FRM relies on global features and FPAD depends on local texture features.

Both the joint finetuning methods also fail to achieve good accuracy for both the tasks. While joint finetuning of the whole ViT model results only in a marginal drop in FPAD accuracy, the large drop in face matching accu-

Table 2. FPAD performance on the SiW [26] test set for various ViT models (with vanilla architecture) trained for the FPAD task using the SiW training set.

Training Scenario	Accuracy (%)	ACER (%)
Zero Shot	97.65	2.39
FPAD Finetuning	**99.79**	**0.22**

racy is unacceptable. This is primarily due to small size of the dataset used for joint finetuning. The SiW training set used for joint finetuning contains only 72 identities, which is more than two orders of magnitude fewer than the CASIA WebFace dataset. Collecting a large-scale dataset containing both identity and attack labels is prohibitively expensive. Therefore, joint training of dual head architectures to achieve good performance on both tasks becomes practically infeasible.

Table 3. Performance of multi-task ViT models when trained using different strategies. Here, Acc denotes accuracy expressed in %.

Training Scenario	FRM Acc	FPAD Acc
Zero Shot	88.70	98.74
FRM Finetuning	**98.96**	95.60
FPAD Finetuning	66.18	**99.79**
Joint Fine-tuning (Heads)	79.72	96.19
Joint Fine-tuning (Whole)	78.78	98.89
Feature Ensemble	**98.96**	*99.12*

Best Intermediate Features for FPAD Experiments on the dual head architecture clearly show that it is difficult to distill both the global and local features into a single ViT class token to achieve good performance on both FRM and FPAD task. This justifies the proposed feature ensemble architecture, which attempts to exploit intermediate patch tokens for the FPAD task without impacting the class token used for the FRM task. However, a key question is the optimal location (within the ViT backbone) for appending the FPAD classification branch. To answer this question, we insert a FPAD classifier after each ViT block and measure the FPAD performance. These results are summarized in Table 4. From this table, we observe that the FPAD performance of the initial and mid-ViT blocks are better in comparison to the FPAD head (which is based on class token embedding). This demonstrates that the micro texture features required for the FPAD task are well-represented in the initial and mid-ViT blocks (between 3 and 6). In particular, blocks 4 and 5 provide the best accuracy irrespective of the classifier type and training strategy. The FPAD accuracy of the subsequent ViT blocks suffers a slow degradation trend, which could be potentially due to the nature of the deeper blocks to focus more on global features instead of local features.

We can also observe that the zero shot scenario (with ImageNet pre-trained weights) is capable of producing more discriminative representations to detect presentation attacks, resulting in higher performance compared to the FRM finetuning scenario. This also confirms that FRM finetuning biases the ViT to focus more on global features required for the FRM task, which leads to some loss of information for the FPAD task. Nevertheless, the representations extracted from the mid-ViT blocks are still good enough for FPAD. Moreover, in terms of network architecture choice, there is little to choose between the MLP and CNN designs. While the CNN architecture is marginally better in the zero shot scenario, the MLP architecture marginally outperforms the CNN in the FRM finetuning scenario. Since our goal is to maximize performance on both tasks, we choose the MLP architecture for the FPAD classification branch. Finally, block 5 gives the best performance for the MLP case and block 4 performs marginally better in the case of CNN, but these differences are negligible.

Table 4. FPAD accuracy (%) and ACER (%) results for the 12 FPAD classification branches and the FPAD classification head on SiW dataset for both zero shot and FRM finetuning scenarios.

	Zero Shot				FRM Finetuning			
	MLP		CNN		MLP		CNN	
Block	ACC	ACER	ACC	ACER	ACC	ACER	ACC	ACER
1	97.96	2.21	98.49	1.66	96.70	3.59	97.74	2.43
2	99.06	0.99	99.03	1.03	98.22	1.88	98.26	1.84
3	99.27	0.82	99.52	0.51	98.67	1.35	98.71	1.34
4	99.54	0.49	**99.82**	**0.18**	98.82	1.20	**98.99**	**1.06**
5	**99.67**	**0.36**	99.77	0.25	**99.19**	**0.88**	98.96	1.10
6	99.36	0.73	99.75	0.27	98.67	1.48	98.90	1.15
7	99.31	0.76	99.50	0.55	98.07	1.95	98.42	1.69
8	99.42	0.61	99.42	0.60	98.00	1.98	98.35	1.68
9	99.05	0.99	99.45	0.57	97.98	1.99	97.72	2.35
10	98.92	1.11	99.33	0.72	96.92	3.24	97.33	2.69
11	98.78	1.25	98.98	1.09	96.58	3.62	96.91	3.12
12	97.81	2.21	98.64	1.43	95.94	4.24	95.71	4.46
class	98.74	1.31	98.74	1.31	95.13	5.32	95.13	5.32

Table 5. FPAD accuracy (%) and ACER (%) results on SiW dataset for the ensemble of FPAD classification branches appended to ViT blocks 4 and 5 under both zero shot and FRM finetuning scenarios.

Network		MLP	CNN
Zero Shot	ACC	99.64	**99.80**
	ACER	0.40	**0.21**
FRM Finetuning	ACC	**99.12**	98.93
	ACER	**0.91**	1.16

Feature Ensemble ViT Based on the insights from Table 4, we average the outputs produced by the 4-th and 5-th ViT blocks to build an ensemble classifier for the FPAD task. Table 5 summarizes the performance of this feature ensemble ViT, which achieves a high FPAD accuracy of 99.12%

Table 6. FPAD performance (ACER (%)) comparison on the HQWMCA dataset between the proposed method and ViTranZFAS [16].

Method	ViTranZFAS	Zero Shot		FRM Finetuning	
		MLP	CNN	MLP	CNN
Flexiblemask	2.60	0.93	**0.54**	**2.27**	7.57
Glasses	15.90	47.39	25.93	49.70	31.67
Makeup	25.80	29.18	28.94	38.42	38.27
Mannequin	2.70	0.28	**0.26**	4.03	3.08
Papermask	2.30	**0.15**	0.27	**0.75**	2.85
Rigidmask	9.50	3.64	**1.19**	18.16	11.20
Tattoo	2.40	0.53	**0.49**	**0.82**	2.60
Replay	12.40	12.02	**9.45**	24.81	**9.76**
Mean $\pm Std$	9.20 $\pm$7.99	11.76 $\pm$16.38	8.38 $\pm$11.39	17.37 $\pm$17.67	13.37 $\pm$12.93

and ACER of 0.91% based on the MLP architecture and FRM finetuning strategy. Note that while these results are marginally lower than best performing block (i.e., block 5 for the MLP case), the differences are small and the ensembling produces more stable predictions. Hence, we recommend using the feature ensemble ViT architecture based on blocks 4 and 5, if overall network size is not a constraint. The performance of this architecture is also shown in the last row of Table 3, which directly compares this approach against the dual head methods. It is obvious that the proposed feature ensemble ViT achieves the same face matching accuracy as the individual ViT finetuned for the FRM task, while its performance on the FPAD is only marginally lower than that of the individual ViT finetuned for the FPAD task. In summary, we have demonstrated that ViTs can be utilized effectively for multiple tasks (FPAD and FRM) in a face recognition system, while achieving high performance for both the tasks using the feature ensemble approach.

5.1. HQWMCA Evaluation

We also evaluate the FPAD performance of the feature ensemble ViT under the unseen attack setting in the HQWMCA dataset. This dataset consists of eight different attacks evaluated using the leave-one-out protocol for each attack. The performance results summarized in Table 6 show that the proposed approach with FRM finetuning outperforms the SOTA results in [16] for some attacks (e.g., flexible mask, paper mask and tattoo), while it fails on other attacks such as glasses and makeup. While we are still investigating the reasons for this failure, our preliminary analysis points towards potential overfitting issues caused due to smaller sample sizes. However, it must be emphasized that our proposed method is finetuned for the FRM task, whereas the work in [16] focuses only on the FPAD task. Without the FRM finetuning, the proposed approach achieves better results for most attacks except glasses.

6. Conclusion and Future Work

In this paper, we propose using a feature ensemble approach to jointly learn the face presentation attack detection and face matching tasks. We exploit the architecture of Vision Transformer (ViT) models to make use of the local features extracted from the intermediate ViT blocks to perform FPAD, while using the global features learned by the class token to perform face matching. Experiments conducted in various settings prove that the ViT feature ensemble method can achieve good performance for both FPAD and face matching tasks in comparison to a basic multi-head approach. Vision transformers are powerful deep learning models that can be utilized effectively for multi-task problems, but this effort is stymied by the lack of large datasets containing both identity and attack labels. Going forward, we aim to achieve a single ViT model that can implement a complete face recognition pipeline.

References

[1] Faseela Abdullakutty, Eyad Elyan, and Pamela Johnston. A review of state-of-the-art in face presentation attack detection: From early development to advanced deep learning and multi-modal fusion methods. *Information fusion*, 75:55–69, 2021.

[2] Timo Ahonen, Abdenour Hadid, and Matti Pietikainen. Face description with local binary patterns: Application to face recognition. *IEEE transactions on pattern analysis and machine intelligence*, 28(12):2037–2041, 2006.

[3] Ajat Arora and Manminder Singh. A novel face liveness detection algorithm with multiple liveness indicators. *Wireless Personal Communications*, 2018, 06 2018.

[4] Yousef Atoum, Yaojie Liu, Amin Jourabloo, and Xiaoming Liu. Face anti-spoofing using patch and depth-based cnns. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pages 319–328, 2017.

[5] Peter N. Belhumeur, Joao P Hespanha, and David J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class

specific linear projection. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7):711–720, 1997.

[6] Deblina Bhattacharjee, Tong Zhang, Sabine Süsstrunk, and Mathieu Salzmann. Mult: An end-to-end multitask learning transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12031–12041, 2022.

[7] Zinelabidine Boulkenafet, Jukka Komulainen, and Abdenour Hadid. Face antispoofing using speeded-up robust features and fisher vector encoding. *IEEE Signal Processing Letters*, 24(2):141–145, 2017.

[8] Zhimin Cao, Qi Yin, Xiaoou Tang, and Jian Sun. Face recognition with learning-based descriptor. In *2010 IEEE Computer society conference on computer vision and pattern recognition*, pages 2707–2714. IEEE, 2010.

[9] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[10] Ivana Chingovska, André Anjos, and Sébastien Marcel. On the effectiveness of local binary patterns in face anti-spoofing. In *2012 BIOSIG-proceedings of the international conference of biometrics special interest group (BIOSIG)*, pages 1–7. IEEE, 2012.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[12] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[15] Anjith George and Sébastien Marcel. Deep pixel-wise binary supervision for face presentation attack detection. In *2019 International Conference on Biometrics (ICB)*, pages 1–8. IEEE, 2019.

[16] Anjith George and Sébastien Marcel. On the effectiveness of vision transformers for zero-shot face anti-spoofing. In *2021 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–8, 2021.

[17] Anjith George, Zohreh Mostaani, David Geissenbuhler, Olegs Nikisins, André Anjos, and Sébastien Marcel. Biometric face presentation attack detection with multi-channel convolutional neural network. *IEEE Transactions on Information Forensics and Security*, 15:42–55, 2020.

[18] Xiaofei He, Shuicheng Yan, Yuxiao Hu, Partha Niyogi, and Hong-Jiang Zhang. Face recognition using laplacianfaces. *IEEE transactions on pattern analysis and machine intelligence*, 27(3):328–340, 2005.

[19] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[20] Anil K Jain and Stan Z Li. *Handbook of face recognition*, volume 1. Springer, 2011.

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[22] Jukka Komulainen, Abdenour Hadid, and Matti Pietikäinen. Context based face anti-spoofing. In *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pages 1–8, 2013.

[23] Chengjun Liu and Harry Wechsler. Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. *IEEE Transactions on Image processing*, 11(4):467–476, 2002.

[24] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017.

[25] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. *arXiv preprint arXiv:1612.02295*, 2016.

[26] Yaojie Liu, Amin Jourabloo, and Xiaoming Liu. Learning deep models for face anti-spoofing: Binary or auxiliary supervision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 389–398, 2018.

[27] Yaojie Liu, Joel Stehouwer, Amin Jourabloo, and Xiaoming Liu. Deep tree learning for zero-shot face anti-spoofing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4680–4689, 2019.

[28] Sébastien Marcel, Mark S Nixon, Julian Fierrez, and Nicholas Evans. *Handbook of biometric anti-spoofing: Presentation attack detection*, volume 2. Springer, 2019.

[29] Jukka Määttä, Abdenour Hadid, and Matti Pietikäinen. Face spoofing detection from single images using micro-texture analysis. In *2011 International Joint Conference on Biometrics (IJCB)*, pages 1–7, 2011.

[30] Muzammal Naseer, Kanchana Ranasinghe, Salman Khan, Fahad Shahbaz Khan, and Fatih Porikli. On improving adversarial transferability of vision transformers. *arXiv preprint arXiv:2106.04169*, 2021.

[31] Muhammad Muzammal Naseer, Kanchana Ranasinghe, Salman H Khan, Munawar Hayat, Fahad Shahbaz Khan, and

Ming-Hsuan Yang. Intriguing properties of vision transformers. *Advances in Neural Information Processing Systems*, 34:23296–23308, 2021.

[32] Gang Pan, Lin Sun, Zhaohui Wu, and Shihong Lao. Eyeblink-based anti-spoofing in face recognition from a generic webcamera. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007.

[33] Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. 2015.

[34] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[35] Additya Popli, Saraansh Tandon, Joshua J Engelsma, Naoyuki Onoe, Atsushi Okubo, and Anoop Namboodiri. A unified model for fingerprint authentication and presentation attack detection. In *Proc. of International Joint Conference on Biometrics (IJCB)*, pages 1–8, 2021.

[36] Rajeev Ranjan, Swami Sankaranarayanan, Ankan Bansal, Navaneeth Bodla, Jun-Cheng Chen, Vishal M Patel, Carlos D Castillo, and Rama Chellappa. Deep learning for understanding faces: Machines may be just as good, or better, than humans. *IEEE Signal Processing Magazine*, 35(1):66–83, 2018.

[37] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[38] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[39] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7262–7272, 2021.

[40] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation from predicting 10,000 classes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1891–1898, 2014.

[41] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, 2014.

[42] Xiaoyang Tan, Yi Li, Jun Liu, and Lin Jiang. Face liveness detection from a single image with sparse low rank bilinear discriminative model. In *Computer Vision – ECCV 2010*, pages 504–517. Springer Berlin Heidelberg, 2010.

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[44] Guoqing Wang, Chuanxin Lan, Hu Han, Shiguang Shan, and Xilin Chen. Multi-modal face presentation attack detection via spatial and channel attentions. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1584–1590, 2019.

[45] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5265–5274, 2018.

[46] Jun Wang, Xiaohan Yu, and Yongsheng Gao. Feature fusion vision transformer for fine-grained visual categorization. *arXiv preprint arXiv:2107.02341*, 2021.

[47] Mei Wang and Weihong Deng. Deep face recognition: A survey. *Neurocomputing*, 429:215–244, 2021.

[48] Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F Wong, and Lidia S Chao. Learning deep transformer models for machine translation. *arXiv preprint arXiv:1906.01787*, 2019.

[49] Zhuo Wang, Qiangchang Wang, Weihong Deng, and Guodong Guo. Face anti-spoofing using transformers with relation-aware mechanism. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 4(3):439–450, 2022.

[50] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019.

[51] Jianwei Yang, Zhen Lei, and Stan Z Li. Learn convolutional neural network for face anti-spoofing. *arXiv preprint arXiv:1408.5601*, 2014.

[52] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014.

[53] Xiaowen Ying, Xin Li, and Mooi Choo Chuah. Liveface: A multi-task cnn for fast face-authentication. In *Proc. of International Conference on Machine Learning and Applications (ICMLA)*, pages 955–960.

[54] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, Oct 2016.

[55] Yaoyao Zhong and Weihong Deng. Face transformer for recognition. *arXiv preprint arXiv:2103.14803*, 2021.