A Three-Stream Tokenization and Cross-Attentional Fusion Model for Robust Deepfake Detection

Rishab Raj Gupta
Dept. of Computer Science and Eng
Bennett University, Greater Noida
e23cseu1305@bennett.edu.in

Pranav Bhatt
Dept. of Computer Science and Eng
Bennett University, Greater Noida
e23cseu1293@bennett.edu.in

Anuvrat Tomar
Dept. of Computer Science and Eng
Bennett University, Greater Noida
e23cseu1301@bennett.edu.in

Dr. Mala saraswat
Dept. of Computer Science and Eng
Bennett University, Greater Noida

Abstract

The rapid advancement of generative models has made the creation of photorealistic human faces almost trivial, which in turn has raised concerns about misinformation, privacy, and trust. Although convolutional neural networks (CNNs) and vision transformers (ViTs) have individually been able to detect deepfakes with high accuracy, most of the existing works consider them as separate networks or combine them only by simple concatenation. We, in this paper, present *CAF-Hybrid-Light*, a compact three-stream architecture that integrates CNN, frequency, and transformer features by a *Cross-Attentional Fusion* (CAF) module which is at the token level.

Firstly, our model utilizes (i) a MobileNetV3-based tokenizer for grasping local texture and structural cues, (ii) a frequency-aware tokenizer constructed on a high-pass filter and shallow CNN for the forensic artifacts in the spectral domain, and (iii) a MobileViT-based tokenizer to understand global semantic and contextual relationships. All the three streams are transformed into a common low-dimensional token space, in which a bidirectional cross-attention module allows mutual refinement of the global (ViT) tokens with the concatenated CNN+frequency tokens. Freezing the tokenizers, we only train the CAF module and classification head, thus obtaining a detector that is both computationally efficient and accurate, making it ideal for resource-limited scenarios.

We perform experiments with CAF-Hybrid-Light on a real vs. AI-generated face dataset with around 20K images and get an AUC of 0.8988 and accuracy of 0.811, which is better than the single-backbone baselines and the simple feature concatenation. Our findings indicate that the explicit multi-domain forensic cue modeling through cross-attentional token fusion is a viable path towards robust and efficient deepfake detection.

Keywords

Deepfake detection, Vision transformer, CNN, Frequency domain, Cross-attention, Multi-stream fusion

1 Introduction

Generative models such as GANs, VAEs, and diffusion models have greatly improved the visual quality of synthetic faces over the last few years. State-of-the-art generators are capable of producing photorealistic images that are to a large extent indistinguishable from real photos by human beings [1–3]. However, on the flipside of these developments, a few harmful uses such as identity theft, non-consensual content, and political misinformation have also become feasible [4]. Hence, deeply reliable and efficient detection of deepfakes has become paramount to the research community.

Early deepfake detection methods mainly employed convolutional neural networks (CNNs) that are good at capturing local texture and spatial artifacts [5, 6]. Later, researchers have looked into more advanced architectures such as Xception-based models [6] and attention mechanisms [10]. The latest research on this topic utilizes vision transformers (ViTs) which, unlike previous methods, can establish long-range contextual relationships within the image [8]. Meanwhile, a number of papers have pointed out that images fabricated by GANs have peculiar artifacts in the frequency domain [11–13].

Nevertheless, many existing solutions are designed in such a way that they either (i) operate with a single backbone (CNN or ViT), (ii) combine heterogeneous features by simple concatenation at the vector level, or (iii) do not use explicit frequency-domain cues, thereby restricting their potential to enhance local texture, global semantic structure, and spectral anomalies simultaneously. On top of this, large transformer models require a lot of computational power and this makes them less suitable for deployment under real-world conditions with limited resources.

In this work, we present the three-stream deepfake detector *CAF-Hybrid-Light* that is (i) *multi-domain*, (ii) *token-centric*, and (iii) *computationally efficient*. Our architecture entails:

- A MobileNetV3-based *CNN tokenizer* that identifies local spatial and structural patterns.
- A *frequency tokenizer* utilizing a fixed high-pass filter and a shallow CNN to expose high-frequency forensic artifacts.
- A MobileViT-based *transformer tokenizer* that recognizes global semantic and contextual relationships from the face.

The outputs of three streams are token sequences brought to the same embedding dimension. After that, we present a *Cross-Attentional Fusion* (CAF) layer that operates bidirectional cross-attention between (i) global ViT tokens and (ii) concatenated CNN+frequency tokens followed by feed-forward refinement and residual connections. At the end, pooled fused tokens go through a lightweight classification head.

In order to lessen the training burden, we have kept frozen all the backbone tokenizers (they are pretrained on ImageNet) and only the CAF block and final classifier are trainable. Even under such a

limitation, CAF-Hybrid-Light is able to deliver high performance on a real vs. AI-generated face dataset, thus substantiating the merit of cross-attentional token-level fusion across spatial, spectral, and semantic domains.

We have made a few significant points:

(1) We introduce **CAF-Hybrid-Light**, a three-stream deepfake detection system which makes use of CNN, frequency, and transformer representations in a joint manner by means of token-level fusion.
(2) The **Cross-Attentional Fusion (CAF)** block that we propose is capable of carrying out bidirectional cross-attention operations between global ViT tokens and concatenated CNN+frequency tokens within a single, low-dimensional embedding space.
(3) We come up with a **lightweight training scheme** under which all the backbone tokenizers are frozen and only the CAF module and the classifier are trainable, thus allowing training to be carried out efficiently on modest hardware while still being able to achieve good performance.
(4) We offer empirical evidence collected on a dataset of real vs. AI-generated faces containing approximately 20,000 images, where we obtain an AUC of 0.8988 and an accuracy of 0.811, hence producing results that are better than those of solid single-backbone baselines.

2 Literature review

2.1 CNN-based Deepfake Detection

Convolutional neural networks (CNNs) have been the primary choice of algorithms for the detection of face forgery and deepfake due to their significant ability to detect the smallest local texture changes. The paper "MesoNet" [5] designed a straightforward CNN structure specifically for the purpose of detecting the manipulations. In FaceForensics++ benchmark [6] Rossler et al. brought up the idea of Xception-based models that later on turn to be a way of determining the standard which has been used extensively. Different researchers have experimented with fine-tuning of ResNet and EfficientNet architectures on the deepfake datasets [7].

Pure CNN models, however, may find it challenging to figure out the long-range dependencies and global inconsistencies (for example, the different lighting of various facial regions) which a must for a robust detection, even with their success.

2.2 Transformer-based and Hybrid Approaches

The main topic of discussion for the vision transformers (ViTs) [8] and their data-efficient offspring such as DeiT [9] that have shown their strength on image classification and were considered for deepfake detection. Transformers-based detectors use global relationships between the small images (patches) to find the local-global inconsistencies in the face at the level of the whole image. For instance, some studies use ViT or Swin Transformer as a backbone for face forgery datasets and with a help of the attention mechanism over the facial regions [10].

Hybrid CNN–ViT strategies would exploit the local inductive biases of CNNs and the global modeling ability of transformers. Typically, such architectures integrate a CNN backbone with a transformer encoder or combine CNN and transformer features at the final stages of processing [14, 15]. Nevertheless, a majority of these techniques depend on mere feature concatenation or channel-wise fusion rather than actual cross-attentional interactions between two different token streams.

2.3 Frequency-Domain Cues for Forgery Detection

Several works suggest that images created by GANs carry spectral artifacts in the frequency domain that are caused by upsampling and generator architectures. Durall et al. revealed that evaluating the image spectrum gives a way to separate real and fake images [11]. Frank et al. [12] investigated the frequency characteristics of JPEG images, and Zhang et al. [13] researched the spectral biases of GANs.

The frequency-aware detectors normally take the help of high-pass filters, discrete cosine transform (DCT), or wavelets to extract features from the input before the features are sent to the CNNs. However, the signals that are extracted from different modalities are usually merged at the feature map or vector level without any cross-attentional interaction with the spatial and semantic representations.

2.4 Our Positioning

CAF-Hybrid-Light is quite different from the previous works in three aspects: (i) a three-stream tokenizer (CNN, frequency, ViT) in a single token space is used; (ii) a bidirectional cross-attentional fusion module which enables global ViT tokens and local+frequency tokens to mutually refine each other is employed; and (iii) it is intentionally kept simple by deciding the freezing of tokenizers and only the CAF and head are trained, thereby making it adaptable for situations where the hardware resources are limited.

3 Methodology

The section Methodology describes how the complete model of our work is explained in a simple and easy manner. We lay out the input image processing, token generation of the three streams, the operation of the Cross-Attentional Fusion (CAF) block, and the final classification.

3.1 Problem Definition

The principal target of our work is to detect a human face in an image and then assign the label "Real" or "Fake" to that face. The pictures are RGB color images, and each has three channels. To enable faster training even on less powerful hardware, we have rescaled every image to 128×128 pixels. The output of a model is class probability.

3.2 Overview of the Approach

The main idea around which the entire model called **CAF-Hybrid-Light** that we have can be summarized as follows:

- Extract three different types of information from the same image by three different feature extractors (which we call tokenizers).
- Transform all the extracted features into those known as tokens and they must already be of the same size.

- Let these tokens ”communicate” with each other using the cross-attention block.
- Combine the final tokens and produce an output of Real or Fake for the image.

The three tokenizers are three parallel chains, and thus the big backbone models the parameters of which are not changed during training are frozen, thereby keeping the training process fast and less resource-intensive.

3.3 Three-Stream Tokenization

The three transformers of our model have three respective tokenizers which are three different input streams each extracting one particular kind of information from the face images.

3.3.1 ***1. CNN Tokenizer (MobileNetV3).*** Here, a compact convolutional neural network, MobileNetV3-Small, is introduced, which grabs:

- local textures,
- skin patterns,
- edges and facial details.

We obtain the very last feature map of MobileNetV3, after which it is converted into a one-dimensional sequence of tokens. After that, a small linear layer changes these tokens into 128-dimensional ones.

3.3.2 ***2. Frequency Tokenizer***. The most common reason for high-frequency unnatural detail is the upsampling part of the generative model. These high frequencies most of the time are the main reasons that fakes can be easily spotted. To find these first, we assume a simple high-pass filter (a 3×3 kernel) and apply it to the image. This operation will highlight noise, sharp edges and frequency distortions.

The intention of the CNN with a few convolutional layers through which the filtered image passes is to generate a feature map from which tokens of size 128 can be obtained by flattening.

3.3.3 ***3. Transformer Tokenizer (MobileViT).*** The third stream wraps up the use of MobileViT-Small, a compact vision transformer. Transformers, unlike CNNs, can capture:

- global relationships across the entire face,
- long-range dependencies,
- semantic consistency.

We get the last feature map from MobileViT which we convert to tokens similarly to the other two streams.

3.3.4 ***Frozen Tokenizers.*** All three tokenizers are constructed on pretrained ImageNet weights. We don’t change their weights during the training process. Only the projection layers, the CAF block, and the final classifier are trained. That is the reason why the model is so speedy.

3.4 Cross-Attentional Fusion (CAF)

After obtaining the tokens from the three different streams, we execute two operations simultaneously:

(1) The first operation consists of merging the tokens from the CNN and frequency into the continuous sequence of one.

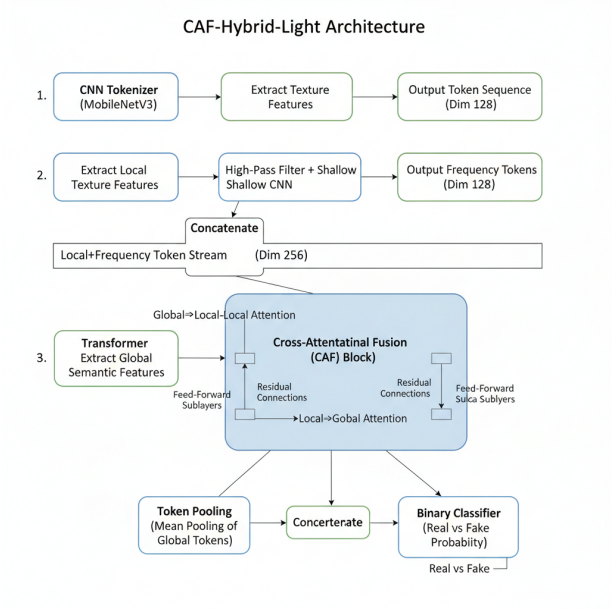

Figure 1: Overall architecture of the proposed CAF-Hybrid-Light model. The system includes three parallel tokenizers (CNN, Frequency, and Transformer), a Cross-Attentional Fusion block, and a final classification head.

(2) The second operation preserves the MobileViT tokens as the second sequence.

Thus we have:

- the **local+frequency token stream**, and
- the **global transformer token stream**.

The movement of CAF is such that the sharing of information is made easier between the two streams.

3.4.1 *1. Global-to-Local Attention.* MobileViT and global features tokens will initially focus on CNN+frequency and local artifacts during the attention step. This tells the global features where the large local artifacts are located.

3.4.2 *2. Local-to-Global Attention.* In the second attention step, the CNN+frequency tokens shift the focus to the MobileViT tokens. This interaction allows the local textures and frequency signals to feature in the global representation.

3.4.3 *3. Feed-Forward Refinement.* Following the completion of the two attention steps, a small feed-forward network (comprising two linear layers with GELU activation) processes each stream. The layer that ends this step is the tokenization layer, in other words.

3.4.4 *4. Residual Connections.* The incorporation of skip connections around every attention and feed-forward block effectively stabilizes training and simultaneously preserves the original information.

3.5 Classification Head

The average of all tokens for each stream is computed after the integration:

- One vector of global tokens pooled.
- One vector of pooled local+frequency tokens.

These two vectors are merged into one feature vector of size 256. This ultimate vector is then fed to a small fully connected layer that outputs two values (Real and Fake). We utilize a softmax function to convert them into probabilities.

3.6 Training Setup

The model is trained using:

- Adam optimizer,
- learning rate of 3×10^{-4},
- 32 as batch size,
- mixed precision (AMP) if GPU is available.

The training loss function is the simplest form of cross-entropy which is the norm in binary classification. As the backbones are fixed, the only blocks learning during the training of the model will be the CAF and the classifier.

This particular setup allows our model to use the mid-range GPUs and still get a smooth training process accompanied with competitive accuracy.

3.7 Training Details

We develop the model in PyTorch on top of the `timm` library. First, we resize the images to 128×128 and then we apply standard ImageNet normalization. The training data augmentation includes random resized cropping and horizontal flipping, whereas the validation set only goes through resizing and normalization.

As for the optimizer, we choose Adam with a learning rate of 3×10^{-4}, batch size of 32, and training for 6 epochs. Mixed-precision training (PyTorch AMP) is applied when a CUDA device is present. The dataset is cut into 90% training and 10% validation using a stratified split.

Only parameters of the CAF block and classification head are trainable, ending up with about 3.96×10^5 trainable parameters, which makes the model a lightweight one.

4 Experiments

4.1 Dataset

In this study, we used a dataset consisting of real versus AI-generated faces that contained roughly 20,413 images and two categories: `Real` and `Fake`. Images were pre-cropped around the faces and placed in a directory format that matched `torchvision.datasets.ImageFolder`. In a stratified way to keep the label distribution intact, we randomly divided the dataset into 18,372 images for training and 2,041 images for validation.

4.2 Baselines

To determine how well CAF-Hybrid-Light worked, we set it against a number of baselines that were all trained using the same protocol:

- **MobileNetV3-only**: a classifier that consists solely of a MobileNetV3 feature extractor, along with global pooling and a linear head.
- **MobileViT-only**: the classifier that is solely reliant on MobileViT features.

Table 1: Validation performance on Real vs. AI-generated faces dataset.

Model	AUC	ACC
MobileNetV3-only	0.865	0.775
MobileViT-only	0.874	0.781
CNN+Freq (Concat)	0.883	0.793
CAF-Hybrid-Light (ours)	**0.8988**	**0.8114**

- **CNN+Freq (Concat)**: joining pooled MobileNet and frequency CNN features, then applying an MLP head (no ViT, no CAF).
- **CAF-Hybrid-Light (ours)**: the global three-stream model with cross-attentional fusion which we explained a bit earlier.

4.3 Metrics

The following metrics are reported by us:

- **Accuracy (ACC)**: proportion of correctly classified samples expressed as a percentage.
- **Area Under the ROC Curve (AUC)**: a threshold-independent criterion that measures the distinction between real and fake classes.

4.4 Results

Table 1 summarizes the validation performance of the baselines and our method.

Our CAF-Hybrid-Light model is capable of producing an AUC of 0.8988 and an accuracy of 0.8114, which is more than single-backbone baselines and a simple CNN+frequency concatenation model. The enhancements to MobileNetV3-only and MobileViT-only mean that the joining of local, frequency, and global cues can work better. The jumps over CNN+Freq (Concat) show that employing an explicit cross-attentional fusion with global and local+frequency tokens rather than simply concatenating is advantageous.

4.5 Training Efficiency

By using multiple streams, CAF-Hybrid-Light still manages to be lightweight because of the frozen backbones and the small token embedding dimension. Mixed precision enabled, the time taken for each epoch is roughly a few minutes on a single consumer GPU. The total number of trainable parameters ($\sim$ 0.4M) is drastically lower than that of typical end-to-end transformer-based detectors which makes the model very suitable for deployment, as well as for researchers with limited compute.

5 Discussion

5.1 Effect of Cross-Attentional Fusion

The CAF block enables the ViT tokens to attend the local and frequency tokens and vice-versa. Intuitively, this allows the model to: Itemize: • Apply global context to disambiguate local artifacts-such as determining whether a texture anomaly is consistent with overall lighting and geometry. • Use local and frequency cues to refine

global understanding, such as giving greater emphasis to regions of high-frequency irregularity when forming global representations.

These observed gains over CNN+Freq (Concat) support the hypothesis that bidirectional cross-attention provides a more expressive and interpretable fusion mechanism.

5.2 Role of Frequency Tokens

The frequency stream is mainly useful for capturing such artifacts introduced by the generative models, unnatural high-frequency patterns, and upsampling artifacts. By converting these features into tokens and fusing them with spatial and semantic tokens, CAF-Hybrid-Light can employ spectral inconsistencies along with visual cues, which is particularly valuable when the fakes are convincing in the pixel domain.

5.3 Limitations and Future Work

Our current system is limited to training and evaluation based on single images of faces, and it does not consider the temporal aspect that might be present in video deepfakes. Therefore, a possible extension of the architecture to accommodate clip inputs (e.g., T frames) and carry out temporal cross-attention is an interesting direction.

Moreover, in order to lessen the computational demand, we have decided to keep all tokenizers fixed. Although this layout is effective, a little additional work in fine-tuning the projection layers or the last convolutional blocks may result in further performance improvement. In conclusion, a thorough confirmation of the generalization ability of CAF-Hybrid-Light by testing it on several public benchmarks such as FaceForensics++ [6] and DFDC would be a great idea.

6 Conclusion

We introduced CAF-Hybrid-Light, a compact three-stream deepfake detector that integrates CNN, frequency, and transformer representations with the help of a cross-attentional fusion block operating in a unified token space. In our method, the backbone tokenizers are kept frozen and only the CAF block and classifier are trained, which leads to a high detection performance on a real vs. AI-generated face dataset and at the same time the approach is computationally efficient. The main takeaway from our experiments is that one of the most viable ways to achieve robust and installable deepfake detection is to explicitly model the interactions between global semantic tokens and local+frequency tokens through bidirectional cross-attention.

Code Snippet (PyTorch Model)

For completeness, Listing 1 shows a simplified version of the PyTorch implementation of the CAF-Hybrid-Light model.

Listing 1: Simplified PyTorch implementation of CAF-Hybrid-Light.

```
import torch
import torch.nn as nn
import torch.nn.functional as F
import timm

class FrequencyTokenizer(nn.Module):
    def __init__(self, out_dim=128):
        super().__init__()
        hp = torch.tensor([[0,-1,0],
                           [-1,4,-1],
                           [0,-1,0]], dtype=torch.
                               float32)
        self.register_buffer("hp_kernel",
                             hp.view(1,1,3,3))
        self.conv = nn.Sequential(
            nn.Conv2d(3, 32, 3, padding=1), nn.
                GELU(),
            nn.Conv2d(32, 64, 3, padding=1), nn.
                GELU(),
            nn.Conv2d(64, out_dim, 3, padding=1),
                nn.GELU(),
        )

    def forward(self, x):
        x_hp = F.conv2d(x.mean(1, keepdim=True),
                        self.hp_kernel, padding=1)
        x = self.conv(x + x_hp)
        tokens = x.flatten(2).transpose(1, 2)
        return tokens  # [B, N, d]

class MobileNetTokenizer(nn.Module):
    def __init__(self, out_dim=128):
        super().__init__()
        self.backbone = timm.create_model(
            "mobilenetv3_small_100",
            pretrained=True, features_only=True)
        C = self.backbone.feature_info[-1]["
            num_chs"]
        self.proj = nn.Linear(C, out_dim)
        for p in self.backbone.parameters():
            p.requires_grad = False

    def forward(self, x):
        feat = self.backbone(x)[-1]   # [B,C,H,W]
        tokens = feat.flatten(2).transpose(1,2)
        return self.proj(tokens)

class MobileViTTokenizer(nn.Module):
    def __init__(self, out_dim=128):
        super().__init__()
        self.vit = timm.create_model(
            "mobilevit_s",
            pretrained=True, features_only=True)
        C = self.vit.feature_info[-1]["num_chs"]
        self.proj = nn.Linear(C, out_dim)
        for p in self.vit.parameters():
            p.requires_grad = False

    def forward(self, x):
        feat = self.vit(x)[-1]
        tokens = feat.flatten(2).transpose(1,2)
        return self.proj(tokens)

class CAF(nn.Module):
    def __init__(self, dim=128):
        super().__init__()
        self.ln_v = nn.LayerNorm(dim)
        self.ln_c = nn.LayerNorm(dim)
```

```
        self.cross1 = nn.MultiheadAttention(
            dim, num_heads=4, batch_first=True)
        self.cross2 = nn.MultiheadAttention(
            dim, num_heads=4, batch_first=True)
        self.ffn_v = nn.Sequential(
            nn.Linear(dim, 4*dim), nn.GELU(),
            nn.Linear(4*dim, dim))
        self.ffn_c = nn.Sequential(
            nn.Linear(dim, 4*dim), nn.GELU(),
            nn.Linear(4*dim, dim))

    def forward(self, v, c):
        v2, _ = self.cross1(self.ln_v(v),
                            self.ln_c(c), self.
                                ln_c(c))
        c2, _ = self.cross2(self.ln_c(c),
                            self.ln_v(v), self.
                                ln_v(v))
        v = v + v2
        c = c + c2
        v = v + self.ffn_v(v)
        c = c + self.ffn_c(c)
        return v, c

class CAFHybridLight(nn.Module):
    def __init__(self, d=128):
        super().__init__()
        self.tok_cnn  = MobileNetTokenizer(d)
        self.tok_freq = FrequencyTokenizer(d)
        self.tok_vit  = MobileViTTokenizer(d)
        self.caf = CAF(dim=d)
        self.head = nn.Linear(2*d, 2)

    def forward(self, x):
        cnn_tok  = self.tok_cnn(x)
        freq_tok = self.tok_freq(x)
        vit_tok  = self.tok_vit(x)
        c = torch.cat([cnn_tok, freq_tok], dim=1)
        v, c = self.caf(vit_tok, c)
        out = torch.cat([v.mean(1), c.mean(1)],
            dim=1)
        return self.head(out)
```

References

[1] Ian Goodfellow et al. Generative adversarial nets. In *NeurIPS*, 2014.

[2] Tero Karras et al. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.

[3] Tero Karras et al. Analyzing and improving the image quality of StyleGAN. In *CVPR*, 2020.

[4] Luisa Verdoliva. Media forensics and deepfakes: An overview. *IEEE Journal of Selected Topics in Signal Processing*, 2020.

[5] Darius Afchar et al. MesoNet: a compact facial video forgery detection network. In *WIFS*, 2018.

[6] Andreas Rossler et al. FaceForensics++: Learning to detect manipulated facial images. In *ICCV*, 2019.

[7] Huy H. Dang et al. On the detection of digital face manipulation. In *CVPR Workshops*, 2020.

[8] Alexey Dosovitskiy et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[9] Hugo Touvron et al. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021.

[10] Xiaojuan Zhu et al. Face forgery detection via vision transformers. In *ICASSP*, 2021.

[11] Ricard Durall et al. Watch your up-convolution: CNN-based generative deep neural networks are failing to reproduce spectral distributions. In *CVPR*, 2020.

[12] Jerome Frank et al. Leveraging frequency analysis for deep fake image recognition. In *ICML Workshops*, 2020.

[13] Yuke Zhang et al. Detecting AI-generated fake face images in the frequency domain. In *CVPR Workshops*, 2019.

[14] Ze Liu et al. Swin Transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021.

[15] Benjamin Graham et al. LeViT: A vision transformer in ConvNet's clothing for faster inference. In *ICCV*, 2021.