Alexandria Engineering Journal 91 (2024) 592–609

Contents lists available at ScienceDirect

Alexandria Engineering Journal

journal homepage: www.elsevier.com/locate/aej

Original Article

Deepfake detection based on cross-domain local characteristic analysis with multi-domain transformer

Muhammad Ahmad Amin [a,*], Yongjian Hu [a], Chang-Tsun Li [b], Beibei Liu [a]

[a] School of Electronic and Information Engineering, South China University of Technology, Guangzhou, 510641, China
[b] School of Information Technology, Deakin University, Geelong Waurn Ponds, VIC 3216, Australia

ARTICLE INFO

Keywords:
Deepfake detection
Multi-domain transformer
Generalization performance
Spectral anomalies
Spatial-frequency domains

ABSTRACT

Deepfake videos present a significant challenge in the current media landscape. While current deepfake detection methods demonstrate satisfactory performance, there is still room for improvement in their ability to generalize and detect unseen scenarios, particularly those involving imperceptible cues. This paper introduces a novel multi-modal deepfake detection model named SpectraVisionFusion Transformer (SVFT), which incorporates spatial and frequency domain statistical artifacts to improve generalization performance. The SVFT framework uses two different backbone encoder models to take advantage of both spatial and frequency domain cues in video sequences, along with a decoder and classifier, for common cross-attention and classification, respectively. The spatial domain branch uses a convolutional transformer-based encoder to analyze facial visual features. In contrast, the frequency domain branch employs a language transformer encoder. Additionally, we introduce a weighted feature embedding fusion mechanism that integrates spectral-based statistical feature embeddings and visual cues to achieve a more comprehensive and balanced spatial-frequency feature representation. By coordinately analyzing these modalities, our model exhibits improved detection and generalization capabilities in unseen scenarios. Our proposed SVFT model achieved 92.57% and 80.63% accuracy in extensive cross-manipulation and dataset evaluation, respectively, while surpassing the performance of traditional and single-domain-based approaches.

1. Introduction

The prevalence of deepfake videos has raised concerns over the genuineness and trustworthiness of multimedia content shared across various media platforms [1]. These videos, often created using advanced AI algorithms, are capable of seamlessly replacing or manipulating the faces of individuals in existing video footage, resulting in highly convincing but fabricated content [2].

The advent of generative neural networks has directed noteworthy advancements in the creation of realistic deepfake content, including variational autoencoders (VAEs) [3] and generative adversarial networks (GANs) [4]. However, even with these advancements, deepfake videos created by these networks often display irregularities in the frequency domain [5–7] and visual elements compared to genuine ones, as the comprehensive frequency spectrum analysis has revealed [8]. Detecting such manipulations [9–20] has become a crucial challenge, requiring innovative approaches that can effectively capture the complex patterns and traits inherent in synthetic media.

Deep learning approaches have gained prominence in recent times owing to their exceptional ability to learn discriminative features. These approaches have explored various domains for feature learning, including spatial perspectives [21–25], frequency analysis, temporal characteristics, and their combinations [26–31]. However, a well-known drawback of these methods is their vulnerability to performance degradation when evaluated on unseen datasets. This is because detection models tend to overfit within specific datasets. To address this challenge, researchers are increasingly focusing on finding more generalized features [24,29,30,32]. However, the ability of deepfake detection methods to generalize across diverse forgeries remains a challenge.

Considering the prospect of generalized detection across unseen forgeries, we aim to enhance deepfake facial forgery detection by improving the learned feature representations with two primary objectives. Firstly, to achieve generalization to unknown forgery patterns,

* Corresponding author.
E-mail addresses: eeahmadamin@mail.scut.edu.cn (M.A. Amin), eeyjhu@scut.edu.cn (Y. Hu), changtsun.li@deakin.edu.au (C.-T. Li), eebbliu@scut.edu.cn (B. Liu).

https://doi.org/10.1016/j.aej.2024.02.035
Received 31 August 2023; Received in revised form 9 January 2024; Accepted 18 February 2024
Available online 27 February 2024

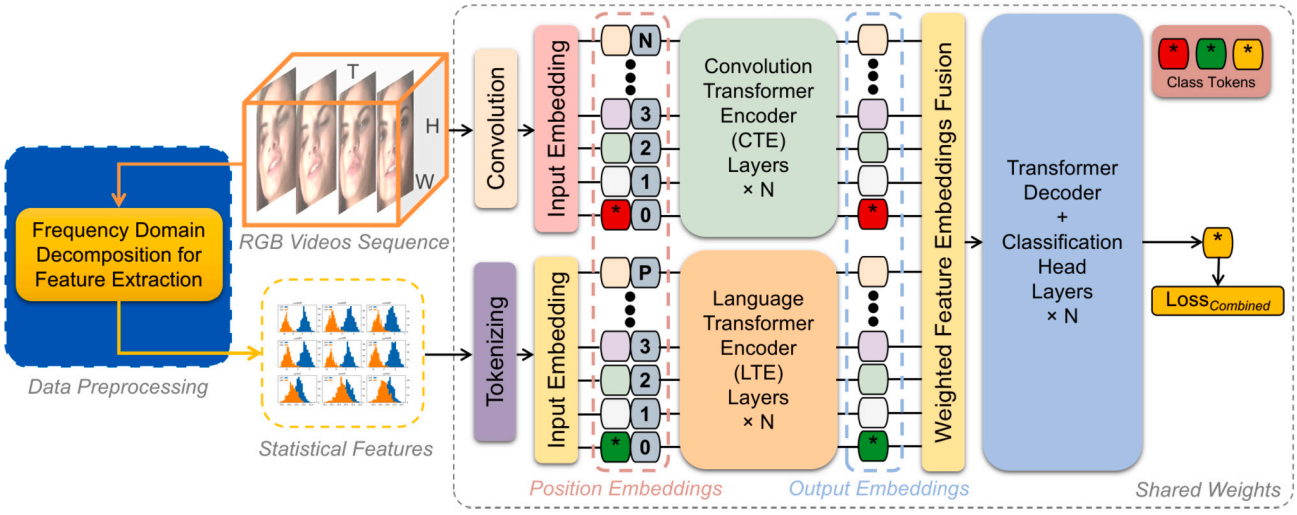


Fig. 1. The SVFT model pipeline. The SVFT method leverages the strengths of the Convolutional Transformer Encoder (CTE) and Language Transformer Encoder (LTE) to capture spatial and frequency cues, enabling joint analysis and common cross-attention of statistical patterns and visual cues.

we explore the domain-invariant discrepancy between authentic and deepfake content with the aim of avoiding overfitting to particular forgery patterns in the training data. Secondly, we identify the intrinsic cues, which are crucial in improving the reasoning of the model about forgery cues. Therefore, our proposed deepfake detection framework, SVFT, analyzes spatial and spectral anomalies to enhance the generalized accuracy and robustness. The SVFT framework integrates two powerful backbone models: the convolutional transformer encoder (CTE) and the language transformer encoder (LTE). CTE excels in capturing local spatial information and modeling complex visual patterns using self-attention mechanisms and convolutional operations. On the other hand, LTE specializes in capturing contextual relationships and semantic meaning within spectral-based statistical features. Our model captures complementary information from spatial and frequency domains and enables a coordinated analysis of unique characteristics by combining these modalities with a common cross-attention decoder and classification head. The collaborative analysis of spatial representations and spectral anomalies helps our model exploit the interplay between visual cues and statistical patterns to enhance overall deepfake detection performance. The proposed SVFT framework is depicted in Fig. 1. Our paper presents several contributions,

- We proposed a new multi-modal deepfake detection transformer model, SpectraVisionFusion Transformer (SVFT), that integrates language and vision encoders, enabling coordinated analysis of the fused complementary cues from facial video sequences and respective frequency domain spectral-based statistical features.
- We introduce a joint training mechanism that arranges real and fake video frames together with their features in a structured order, enhancing the ability of the model to discern between real and deepfake content.
- We also developed a weighted feature embedding fusion mechanism that combines the visual cues from the convolutional encoder and spectral-based statistical features from the language encoder. This fusion, along with a common cross-attention decoder and classifier, enables the joint analysis of fused complementary information, resulting in improved detection performance.
- Finally, we assessed the effectiveness of the SVFT approach on benchmark datasets to showcase its superior performance and enhanced detection capabilities in discerning real and deepfake videos compared to traditional and single-domain-based methods.

In this paper, subsequent sections are as follows: Section 2 recaps related work. Section 3 illustrates our proposed method and delves into the effectiveness of frequency domain analysis. Section 4 details the experimental settings. Section 5 explains the evaluation results and analysis. In Section 6, we provide our conclusion.

2. Related work

In the following section, we will explore the studies conducted on deepfake forensics and synthesis relevant to our methodology.

2.1. Deepfake synthesis

Deepfake multimedia content production has experienced significant advancements since the introduction of adversarial architecture [4] by Goodfellow et al. in 2014. These generative networks are trained via an adversarial process, assuming the dual roles of generator and discriminator, with the objective being to create synthetic facial alterations indistinguishable from genuine ones. Notably, research on GANs has evolved into several breakthroughs that have resulted in new state-of-the-art manipulation methods. These include the benefits of labeled data in training [33], spectral normalization [34], leveraging the Wasserstein distance [35], and style mixing or progressive growth [36]. Over time, facial manipulation algorithms have also improved exponentially. Early attempts relied on landmark-based methods [37,38] that utilized the facial structure to identify a reference face with an identical posture to the target face and then swapped to the target look after acclimating the lighting, geometry, color, and other aspects. However, these methods were limited to altering faces with exact poses. Succeeding efforts [14,39–42] introduced $3D$ face representations to address this limitation, including face reenactment [11], expression manipulation [43], and the Face2Face approach [17]. Still, they could not generate nonexisting parts (e.g., expression) in the reference face, resulting in unnatural cues in the synthesized face. Lately, GAN-based approaches have achieved better face-swapping outcomes. Specifically, Korshunava et al. [44] introduced a transformation method for each target individual, while IPGAN [45], RSGAN [46], and FaceShifter [47] introduced content-agnostic face-swapping tools. Although current deepfake generation methods produce high-quality results, most facial alterations still require better attention to the inner face and a blending phase to achieve a genuinely realistic deepfake. Also, the spectral anomalies in deepfake content are primarily caused by the upsampling procedures used during the generation process [5,6]. Analysis of reconstructed deepfake content reveals distinct spectral artifacts that indicate the upsampling strategies lead the generator networks towards such cues.

2.2. Deepfake forensics

Traditional forensic techniques for detecting forgery employed features such as pattern analysis, local noise assessments, and lighting modeling. However, as deep learning methods advance, facial forgeries have become more visually realistic, making these traditional features less effective. To handle this, recent works propose using high-dimensional semantic cues in the conceptual feature spaces, such as local textures, noise statistics, and frequency information, to pinpoint specific forgery patterns to distinguish deepfake faces from real ones.

Earlier deepfake detection studies have utilized convolutional neural networks (CNN) such as MesoNet [21], Xception [22], RNN [23], and CapsuleNet [24] to extract tampering visual characteristics in an inexplicable manner. The abovementioned methods primarily rely on the spatial domain, with Xception as the baseline method. These standard techniques for detecting tampering involve extracting features from each video frame. However, these conventional methods lack interpretability and perform poorly when applied to different datasets, as they do not clearly explain the underlying operations used to extract the features.

Many researchers have identified the intrinsic differences between real and forged faces for better generalization performance. One notable detector that has been proposed by Li et al. is Face X-ray [25], which effectively extracts blending traces from forged images and produces good results on unknown datasets. Regardless, such low-level traces quickly fade when facing image degradation, resulting in a significant decline in performance. While most deepfake generation approaches are conducted in the spatial domain, they often neglect the importance of enforcing fidelity in the frequency domain. To address this, researchers have proposed methods that combine different domains, such as spatial-frequency [26,27] and spatial-temporal [28]. For example, Liu et al. [26] have studied up-sampling operation in the frequency domain, a standard step in most facial forgery procedures, which can result in mutations in the phase spectrum. To address this, they have presented a novel method, Spatial Phase Shallow Learning (SPSL), that blends spatial pictures and phase spectrum to catch the up-sampling cues of facial manipulation. Yuan Wang et al. [27] have introduced a Spatial-Frequency Dynamic Graph-Based network (SFDG) that uses relation-aware spatial-frequency features to promote generalized forgery detection with a graph-based relation-reasoning approach. Similarly, Hu et al. [28] have presented a frame-temporality two-stream convolution network (FTSC) for compressed deepfake content detection that uses spatial and temporal features. However, despite the improved performance, these approaches mainly rely on the learned forgery patterns presented in the training samples. Thus, they will experience an apparent performance decline for several reasons, such as when dealing with disruptive image degradation, novel forgery patterns, and not coordinately learning spatial, frequency, or temporal features during the training process.

Several studies have attempted to improve deepfake detection by incorporating auxiliary supervision and attention methods to drive the network focus toward local traits in forged areas. For example, Zhao et al. [29] presented a fine-grained deepfake detection framework (Multi-Att) that aggregates high-level semantic and local texture information into attention maps to categorize real and deepfake samples. Regardless, this approach fails to distinguish highly compressed videos with blurry textures. Similarly, Cao et al. [30] employed the reconstruction difference of authentic faces using pixel-level segmentation mapping methods (RECCE) for deepfake detection, which generalizes better to unknown forgery patterns. Nonetheless, these methods rely on dataset-specific forgery patterns or manipulation techniques. Wang et al. [31] introduced a scheme of locally and globally learning image features utilizing the deep convolutional transformer network (CPT) to exploit the global characteristics within a facial image for generalized deepfake detection. However, this method has a limited detection performance on low-quality images.

Despite the vanilla spatial-frequency-temporal fusion ethos, the current methods lack content-aware feature modeling and learning, and when they face unseen forgeries or image perturbations, their performance drops significantly. In contrast, our proposed SVFT approach excavates content-aware frequency clues and enables the comprehensive fusion of spatial and frequency features through common cross-attention and coordinated analysis. This approach provides a promising solution for deepfake detection by considering both spatial and frequency clues in a content-aware manner.

3. Proposed method

Our proposed SVFT approach is outlined in Fig. 1, which comprises four distinct modules. Firstly, a frequency domain analysis calculates the spectral-based statistical features between the three color channels of video frame sequences. By measuring the spectral difference between the color channels of the video frames, we can identify discrepancies that may indicate deepfake manipulations. The mean values, average mean, minimum, maximum, and correlations of these spectral differences are calculated to form a spectral feature vector. Secondly, we use a convolution transformer-based encoder, derived from the convolution vision transformer (CvT) [48], to detect visual patterns indicative and capture high-level features of deepfake manipulations. Indicative visual patterns are unique cues, structures, or characteristics within an image that can be analyzed to detect forgery anomalies. At the same time, high-level features are intrinsic characteristics that exist throughout all the manipulated deepfake content. Thirdly, we use a language transformer-based encoder, which is based on DistilBERT (DBT) [49], to exploit contextual relationships and semantic meaning within the input frequency domain features that could provide valuable insights for deepfake detection. The SVFT core component is the feature fusion mechanism, which merges the encoded visual representations with the spectral-based features extracted from video frames. This fusion of spectral-based features enriches the discriminative power of the model, enabling it to detect subtle anomalies introduced during the deepfake generation process. The weighted fusion mechanism assigns importance to the output of each encoder, helping the model to emphasize the more informative modality in the decision-making process. Finally, we introduce a transformer decoder layer that facilitates cross-model attention. This decoder layer facilitates the model to capture the intricate spectral cues, semantics, and visual patterns indicative of deepfake manipulations. By integrating all these components, our approach provides improved detection and generalized performance.

3.1. Frequency domain decomposition

Generative models can produce synthetic face videos that appear natural and realistic, though the generation of high-frequency details often leads to a systemic bias. Spectral artifacts may arise in the generator due to different upsampling methods, which can direct it toward specific spectral anomalies. However, the discriminator may find it challenging to deal with high-frequency artifacts of low magnitudes as authentic frames often have such high-frequency components. Hence, synthetic video frames generated without direct correlation constraints may contain detectable spectral anomalies and distortions in their frequency spectrum.

3.1.1. Frequency domain spectral analysis

Based on our observations, it is possible to utilize spectral statistical artifacts to identify deepfakes. Specifically, we begin by examining the structure of a $2D$ generative neural network $G\left(x, y; W, H^1\right)$ with respect to the rendering of spatial frequencies, which can be realized as a sequence of convolution layers $Conv_i^l : \mathbb{R}^{d_{l-1}\times d_{l-1}} \rightarrow \mathbb{R}^{d_1\times d_1}$ with a set of parameters space $W \in \mathcal{W}$, input features $H^1 \in \mathbb{R}^{d_0\times d_0}$, and output space $\mathbb{R}^{d\times d}$:

$$H_i^{l+1} = Conv_i^l\left(H^l\right) = \sigma\left(\sum_{c=1}^{C_{(l)}} F_{ic}^l \circledast Up\left(H_c^l\right)\right) \tag{1}$$

In the context of layer-based generative models, the number of output channels is denoted by i, the notation l denotes the depth or layer index, c the number of input channels, and $C_{(l)}$ is the output channel H^l color space at layer l. The output is generated by applying a parametric $2D$ filter with size k_l, $F_{ic}^l \in \mathbb{R}^{k_l \times k_l}$, to the input, H_c^l, in the c-th channel of the i-th output channel of the l-th $Conv$ layer is denoted by $Conv_i^l \in \mathbb{R}^{d_l \times d_l}$ with spatial dimension d_l. The resulting output is then passed through a nonlinearity operator, $\sigma(.)$, and a convolution operator, $\circledast$. The output for each layer, l, of the network is indexed by i and has a spatial dimension of d_l. The upsampling operator, $Up(.)$, is used to increase the spatial dimensions of the output.

$$H_i^{l+1} = \sum_{c=1}^{C_{(l)}} F_{ic}^l \circledast Up\left(\sigma\left(H_c^l\right)\right) \tag{2}$$

According to Khayatkhoei et al., [6], restricting the activation function σ to rectified linear units (ReLU) results in the generative network producing a result space with a piece-wise linear structure, particularly in a miniature vicinity of H^1. Consequently, it can be assumed that the generative network is end-to-end linear. Thus $\sigma(.)$ can be excluded from Equation (2). This implies that our analysis is confined to the location of a particular single sample. Also, it should be noted that $Up(.)$ is a fixed shift-invariant linear function that can be integrated with H^l. However, it is not removed but instead considered a preprocessing process on the intake to each layer, leading to the following result,

$$H_i^{l+1} = \sum_{c=1}^{C_{(l)}} F_{ic}^l \circledast H_c^l \tag{3}$$

We apply $2D$ discrete Fourier transform (DFT) to derive the spatial frequency spectrum components of any $2D$ filter in the l-th $Conv$ layer of a generated deepfake video frame, in Equation (3),

$$\widetilde{H}_i^{l+1} = DFT\left(H_i^{l+1}\right) = DFT\left(\sum_{c=1}^{C_{(l)}} F_{ic}^l \circledast H_c^l\right) \tag{4}$$

In the Fourier space, the convolution theorem states that the convolution of two functions is equivalent to the convolution in real space. Thus, we can streamline Equation (4) by utilizing this theorem,

$$\widetilde{H}_i^{l+1} = \sum_{c=1}^{c_{(l)}} \widetilde{F}_{ic}^l \times \widetilde{H}_c^l = \left\langle \widetilde{F}_i^l, \widetilde{H}^l \right\rangle \tag{5}$$

In equation (5), the variables $\widetilde{F}_i^l = \left(\widetilde{F}_{i1}^l, \ldots, \widetilde{F}_{iC}^l\right)^T$ represents a sequence of spatial frequency components found in the frequency spectrum of $2D$ trainable filters of the l-th $Conv$ layer with filter size k_l and spatial dimension d_l. Additionally, the $\widetilde{H}^l = \left(\widetilde{H}_1^l, \ldots, \widetilde{H}_C^l\right)^T$ represents the synthesized output deepfake video frame. Equation (5) further indicates that each channel in the following layer is a composite of all channels from the prior layer using different groups of elements in the frequency domain.

Thus, improving the output spectrum of each deepfake frame in a neighborhood can only be achieved by adjusting the filter $\widetilde{F}_i^l$ spectrum. The filters aim to extract the informative components from the intake spectrum, which is aliased by upsampling. Accordingly, the maximum spatial frequency in each $Conv^l$ layer should be capped by the Nyquist to have the highest correlation between the channels of a generated deepfake frame. To forge a fresh deepfake frame $\widetilde{H}_i^{l+1}$ in a video sequence, we assume the coordinates of the new frame as $\widetilde{F}_i^l$, with the previous frame $\widetilde{H}^l$ serving as a reference. Each vector $\widetilde{F}_i^l$, $i = 1, \ldots, C_{(l+1)}$ is trained independently and applied to reduce the loss of each $\widetilde{H}_i^{l+1}$ component in the preceding layer. The generative networks foresee $\widetilde{H}^l$ as adequate for constructing a color channel per separate coefficient vector $\widetilde{H}_i^l$, $i = 1, \ldots, C_{(l+1)}$. Consequently, the output frame should appear natural in the spatial domain by stacking three generated color channels of a deepfake video frame (red, green, and blue). According to Theorem 1 in [6], the outer layers of the generative network, which have larger spatial dimensions d_l, primarily generate high-frequency content with filter size k_l specified, the more significant the size of d_l, the higher the correlation in the filter spectrum and, therefore, the slighter its capacity. Thus, the outward layers liable for rendering high-frequency components in deepfake frames are additionally capped in their spectrum corresponding to the internal layers with a minor size of d_l. Although a more significant filter dimension, k_l, can be employed in the outward layers to offset the impact of a bigger size of d_l, still the lower-frequency components will have a bigger end-to-end filter dimension than high-frequency components, therefore, a more negligible spectrum correlation. It is essential to note that all layers in a generative model can generate without aliasing but fall short for high-frequency content with low magnitude.

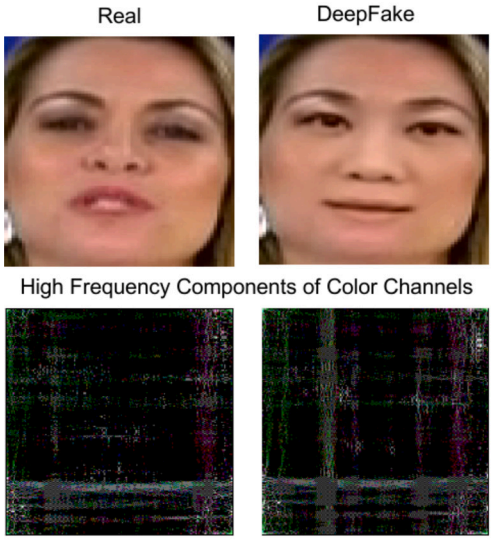


Fig. 2. Comparing the high-frequency components of real and deepfake video frames to identify differences in the spectrum of the same subject.

When generating deepfake video frame sequences, it is vital to consider the application of direct correlation between the generated color channels. In case it fails to do that, it might only cause slight changes in spatial frequency components, which only shift minor attributes in the visualization of the rendered deepfake video and can not be seen from the naked eye. However, these changes will yield significant spectrum aberrations in the frequency domain. Fig. 2 illustrates an example of spectral anomalies and the difference between a real video sample frame and a generated deepfake sample frame of the same subject in the frequency spectrum. It is crucial to acknowledge that these potential limitations during the generation process can be exploited to improve the detection performance.

3.1.2. Frequency domain spectral features

Our analysis has led us to suggest vital yet simple statistical features that can pinpoint spectral anomalies in deepfake video frames. These features include the minimum $\left(Min_{Avg.}\right)$, maximum $\left(Max_{Avg.}\right)$, and mean $\left(Mean_{Avg.}\right)$ of average spectrum differences between color channels. Also, the correlation coefficients $\left(Corr_{RG/RB/GB}\right)$ between the spectrum of the color channels in the frequency domain, and we employ the Pearson correlation to compute these values. In order to extract these spectral anomalies based statistical features, we applied the

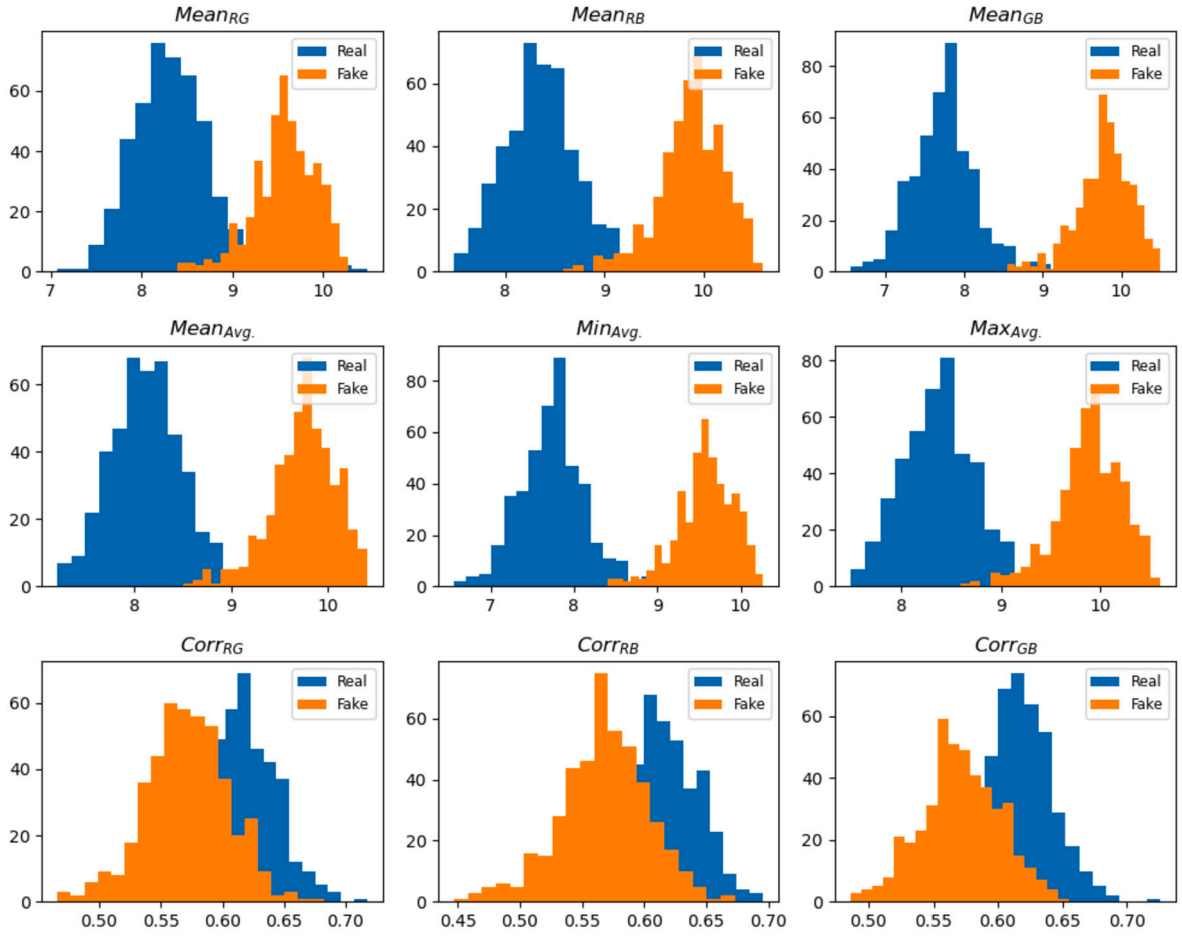


Fig. 3. The statistical features, such as $Mean_{RG}$, $Mean_{RB}$, $Mean_{GB}$, $Mean_{\text{Avg.}}$, $Min_{\text{Avg.}}$, $Max_{\text{Avg.}}$ values distributions and the correlation statistical values distributions $Corr_{RG}$, $Corr_{RB}$, and $Corr_{GB}$, of real (shown in blue) and deepfake (shown in orange) sequences are analyzed through frequency spectrum analysis.

$2D$ discrete Fourier transform $DFT\left(x_{R/G/B}\left(k_x,k_y\right)\right)$ to a $2D$ video frame $x_{R/G/B}(p,q)$ as in (6),

$$DFT\left(x_{R/G/B}\left(k_x,k_y\right)\right)=\frac{1}{mn}\sum_{p=0}^{m-1}\sum_{q=0}^{n-1}x_{R/G/B}(p,q)e^{-i2\pi\left(\frac{k_xp}{m}+\frac{k_yq}{n}\right)} \tag{6}$$

Where p and q represent the coordinates of a video frame $x_{R/G/B}(p,q)$. Further, the DFT of all channels is individual. By computing the modulus of $DFT\left(x_{R/G/B}\left(k_x,k_y\right)\right)$ as formulated in (7), we obtain the magnitude spectrum for all three color channels.

$$Spec\left(x_{R/G/B}\left(k_x,k_y\right)\right)=\left|DFT\left(x_{R/G/B}\left(k_x,k_y\right)\right)\right| \tag{7}$$

It can also be denoted as,

$$Spec\left(x_{R/G/B}\left(k_x,k_y\right)\right)=[\Re\left\{DFT\left(x_{R/G/B}\left(k_x,k_y\right)\right)\right\}^2$$
$$+\Im\left\{DFT\left(x_{R/G/B}\left(k_x,k_y\right)\right)\right\}^2]^{\frac{1}{2}} \tag{8}$$

Equation (8) shows the real part of $DFT\left(x_{R/G/B}\left(k_x,k_y\right)\right)$ as $\Re$ and the imaginary part as $\Im$. Additionally, the correlation coefficients $\left(Corr_{RG/RB/GB}\right)$ between the spectra of the color channels $\left(Spec\left(x_{R/G/B}\left(k_x,k_y\right)\right)\right)$ were computed as described in (9).

$$Corr_{RG}=\rho\left(Spec\left(x_R\left(k_x,k_y\right)\right),Spec\left(x_G\left(k_x,k_y\right)\right)\right) \tag{9}$$

Here $Corr_{RG}$ denotes the correlation coefficients between color channels, and can be written as,

$$Corr_{RG}=\frac{Cov\left(Spec\left(x_R\left(k_x,k_y\right)\right),Spec\left(x_G\left(k_x,k_y\right)\right)\right)}{\sqrt{\sigma\left(Spec\left(x_R\left(k_x,k_y\right)\right)\right)\cdot\sigma\left(Spec\left(x_G\left(k_x,k_y\right)\right)\right)}} \tag{10}$$

In Equation (10), Cov represents covariance, while σ represents the standard deviation of both $Spec\left(x_R\left(k_x,k_y\right)\right)$ and $Spec\left(x_G\left(k_x,k_y\right)\right)$. The minimum, maximum, and mean values are calculated by using the average spectrum difference between the color channels, namely $D_{RG/RB/GB}$, mathematically expressed as in Equation (11).

$$D_{RG}=\frac{1}{mn}\sum_{k_x=0}^{m-1}\sum_{k_y=0}^{n-1}\left(\left|Spec\left(x_R\left(k_x,k_y\right)\right)-Spec\left(x_G\left(k_x,k_y\right)\right)\right|\right) \tag{11}$$

The mean ($Mean_{Avg.}$) value is computed as in (12),

$$Mean_{Avg.}=\left(D_{RG}+D_{RB}+D_{GB}\right)/3 \tag{12}$$

The minimum and maximum values of the average spectrum differences $\left(D_{RG/RB/GB}\right)$ of all three color channels are denoted by $Min_{Avg.}$, and $Max_{Avg.}$, respectively, in Fig. 3. Additionally, the mean values of color channel spectrum differences $\left(D_{RG/RB/GB}\right)$ are calculated individually as:

$$Mean_{RG}=Mean\left(D_{RG}\right) \tag{13}$$

$$Mean_{RB}=Mean\left(D_{RB}\right) \tag{14}$$

$$Mean_{GB}=Mean\left(D_{GB}\right) \tag{15}$$

According to the findings presented in Fig. 3, statistical features based on the frequency spectrum show a clear difference between genuine and deepfake frames of the same subject. To validate this claim,

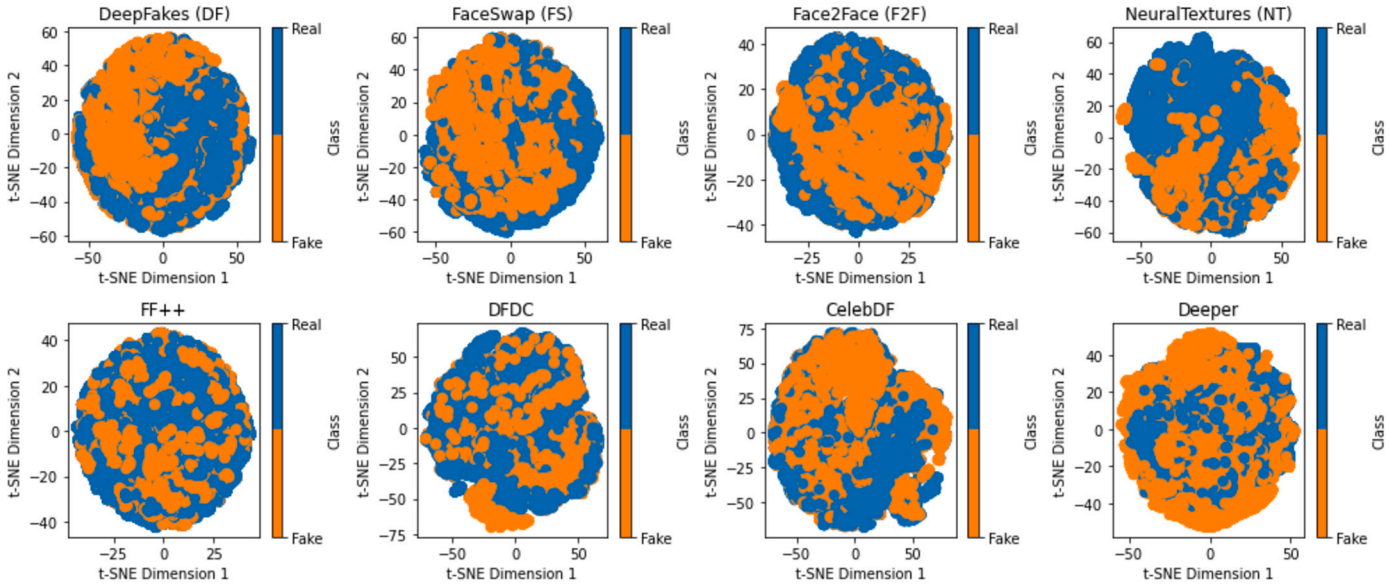


Fig. 4. t-SNE feature distributions between Real and Deepfake video sequences on all the datasets used in our evaluations.

we scrutinized the histogram distributions of the proposed statistical features from the FF++ dataset [50]. Blue represents real face frames, while orange represents fake ones. During the deepfakes generation, interpolation operations alter the spatial distribution of color channels in a generated face frame due to changes in the facial geometry. The statistical anomalies depict the shifts in the pixel value distribution of the real and deepfake color channels in the $2D$ discrete Fourier spectrum revealed by $Mean_{RG}, Mean_{RB}, Mean_{GB}, Mean_{Avg.}, Min_{Avg.}$ and $Max_{Avg.}$ values distributions. Further, analysis of the correlation statistical values distributions at the bottom of Fig. 3, such as $Corr_{RG}$, $Corr_{RB}$, and $Corr_{GB}$, shows that the real video sequences have a high correlation between color channels. In contrast, the deepfake ones show less correlation. This indicates that the generative model fails to reproduce the statistical values of high-frequency components with low magnitude. Therefore, we conclude that the generated deepfake video sequences lack the application of unpretentious correlation constraints between color channels and reveal spectral irregularities. Our proposed spectral anomalies-based statistical features can help differentiate between authentic and deepfake video sequences. The t-SNE feature distributions for all training datasets used in our evaluation are shown in Fig. 4.

3.2. Convolutional transformer encoder branch

The design of the CTE architecture incorporates two novel convolution-based processes into the conventional vision transformer (ViT) [51] framework, i.e., convolutional token embedding and convolutional projection as illustrated in Fig. 5.

In order to establish a hierarchical structure [52] for the vision branch in the SVFT model, akin to CNNs but with an attention mechanism, the CTE model can be partitioned into three stages. The initiation of each stage involves a convolutional token embedding operation, which executes an overlapping convolution process with a specific stride on a reshaped $2D$ token map. Following this, layer normalization is applied to the output per the procedure outlined in [53]. This approach permits the model to grasp local information and reduce sequence size while increasing token feature dimension across each stage, attaining spatial down-sampling, and augmenting feature maps in numbers. The third step in each stage applies the convolutional projection to individual self-attention blocks within the transformer block, which involves performing a depth-wise separable $s \times s$ convolution process [22] on a token map that has been reshaped into a $2D$ format. This strategy improves the ability of the model to grasp the local spatial context and lessens semantic ambiguity in the attention mechanism. The application of convolution stride aids in managing computational complexity. It allows for sub-sampling of the key and value matrices, resulting in improved efficiency of up to 4×4 or more while having minimal impact on performance. The convolutional token embedding and convolutional projection are discussed in detail as follows,

3.2.1. Convolutional token embedding

The convolution token embedding process in CTE is designed to capture local spatial contexts in a multi-stage hierarchy approach. This process models a range of local features, including higher-order semantic primitives and low-level edges. Specifically, the output token map from a previous stage x_{i-1} can be expressed as a $2D$ video sequence frame x_i. This approach allows for the modeling of local spatial contexts in a hierarchical manner, thus improving upon the existing CNNs.

$$x_{i-1} \in \mathbb{R}^{H_{i-1} \times W_{i-1} \times C_{i-1}} \tag{16}$$

In the input to the stage i, our objective is to obtain a functional mapping of x_{i-1} to new tokens $f\left(x_{i-1}\right)$ with a channel size C_i. To achieve this, we employ function $f(\cdot)$, which is a two-dimensional convolution operation with a kernel size of $s \times s$, a stride of $s - o$, and p padding to address boundary conditions. The resulting token map is represented as follows,

$$f\left(x_{i-1}\right) \in \mathbb{R}^{H_i \times W_i \times C_i} \tag{17}$$

Where, H_i represents the height and W_i denotes the width of a new token map,

$$H_i = \left[\frac{H_{i-1} + 2p - s}{s - o} + 1\right], \tag{18}$$

$$W_i = \left[\frac{W_{i-1} + 2p - s}{s - o} + 1\right] \tag{19}$$

Following pre-processing, $f\left(x_{i-1}\right)$ is transformed into an $H_i \times W_i \times C_i$ size and subsequently subjected to layer normalization [53]. This normalization process prepares the input for the succeeding stage i for transformer blocks.

The convolutional token embedding layer adjusts the dimensions of the token feature and number at different stages. Adjusting the convolution parameters can reduce token sequence length and increase feature

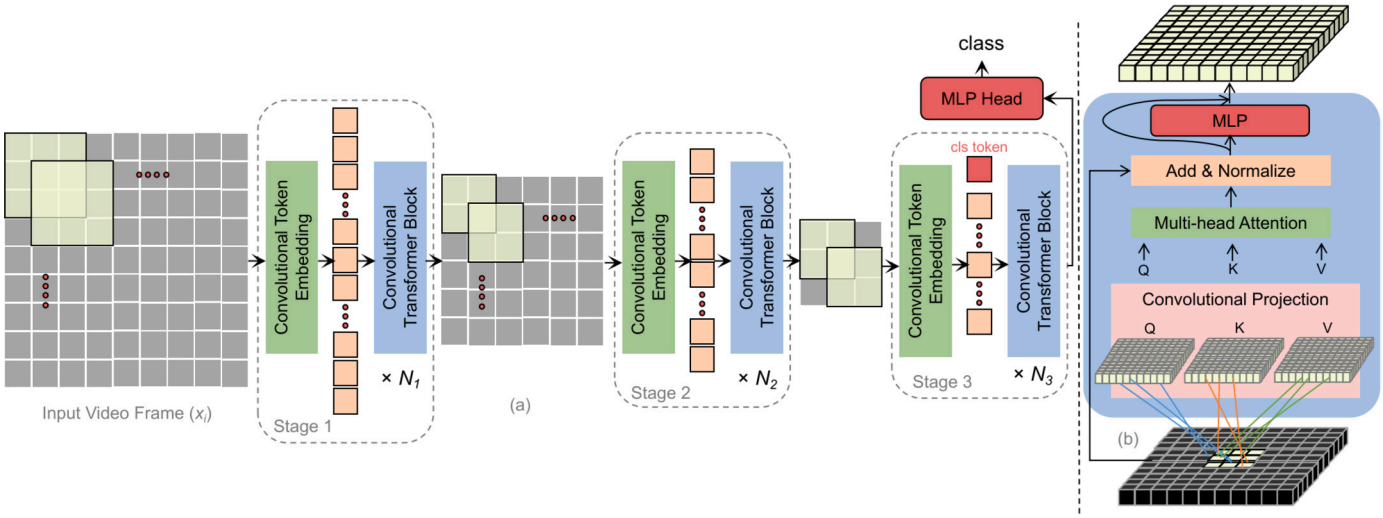


Fig. 5. (a) The proposed CTE architecture comprises a hierarchical multi-stage structure, enabled by the Convolutional Token Embedding layer. (b) Illustration of the convolution projection as the initial layer in the Convolutional Transformer Block.

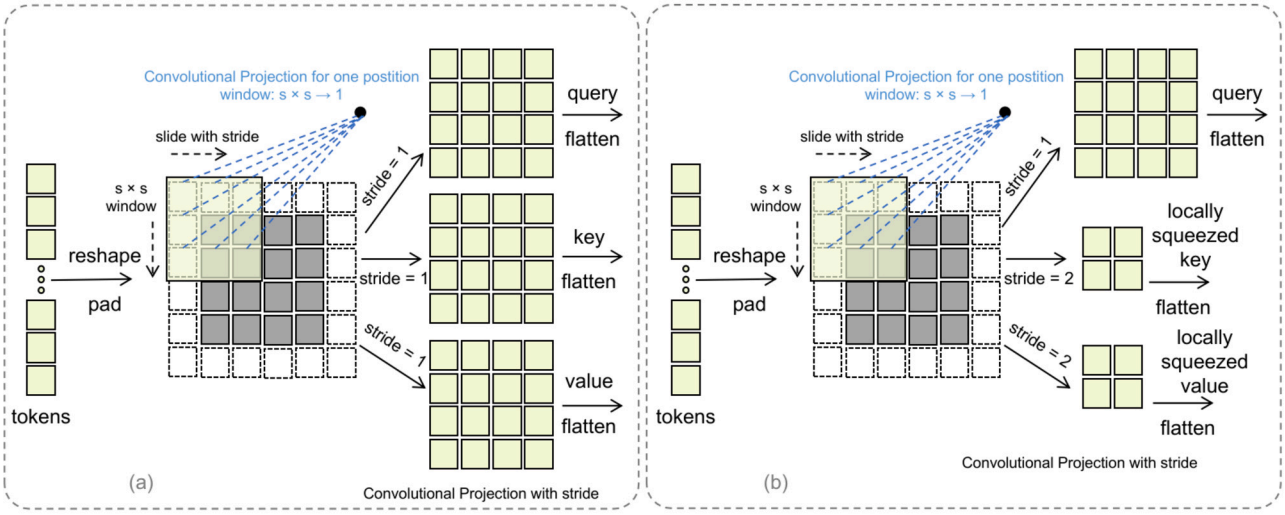


Fig. 6. (a) Two types of Convolutional projection. (a) default convolutional projection, and (b) Squeezed convolutional projection. We use squeezed convolutional projection.

dimension. This allows tokens to represent more complex visual patterns over larger areas, which helps improve the CTE learning process.

3.2.2. Convolutional projection for attention

The convolutional projection layer in the CTE model aims to enhance local spatial context modeling by enabling the under-sampling of key (K) and values (V) matrices as shown in Fig. 5-(b) for better efficiency. This layer uses depth-wise separable $s \times s$ convolutions to reduce computational costs and generalize the transformer block of the ViT model.

Fig. 6-(a), an $s \times s$ convolutional projection is depicted, which involves the initial reshaping of tokens into a $2D$ token map. A depthwise separable convolution layer with the kernel dimension of s is used for the convolutional projection. Eventually, the projected tokens are flattened into a $2D$ form to continue processing.

$$x_i^{q/k/v} = Flatten\left(Conv2d\left(Reshape2D\left(x_i\right), s\right)\right) \tag{20}$$

In $Q/K/V$ matrices, $x_i^{q/k/v}$ represents the input token at layer i, while x_i represents the original undisturbed token before the convolutional projection. The $Conv2d$ process involves a depth-wise separable convolution [22] implemented using a combination of depth-wise $Conv2d$, $BatchNorm2d$, and point-wise $Conv2d$. Here, s symbolizes the convolution kernel size. The new transformer block design, which includes the convolutional projection layer, can easily realize the original position-wise linear projection layer with a 1×1 convolution layer.

3.2.3. The CTE model operations over three stages

The first stage of the CTE model starts with a convolutional embedding layer as shown in Fig. 5-(a), where the subsequent Multi-Head Self-Attention (MHSA) operation enables the model to capture global contextual information (by utilizing 64 output channels with a stride of 4 and 7×7 kernel size), and two Multi-Layer Perceptron (MLP) layers further process the extracted features. In the second stage, the CTE model focuses on refining a frame x_0 features F_{x_0} obtained from the previous stage. It also starts with a convolutional embedding layer employing 192 output channels with a stride of 2, and a 3×3 kernel size. A convolutional projection layer follows this embedding step. Then MHSA is applied by the model to incorporate a global context. The two MLP layers are subsequently employed to further enhance the extracted features. In the third stage, the model performs additional feature refinement. A convolutional embedding layer with 384 output channels, a 3×3 kernel size, and a stride of 2 is used. Similarly, this is succeeded

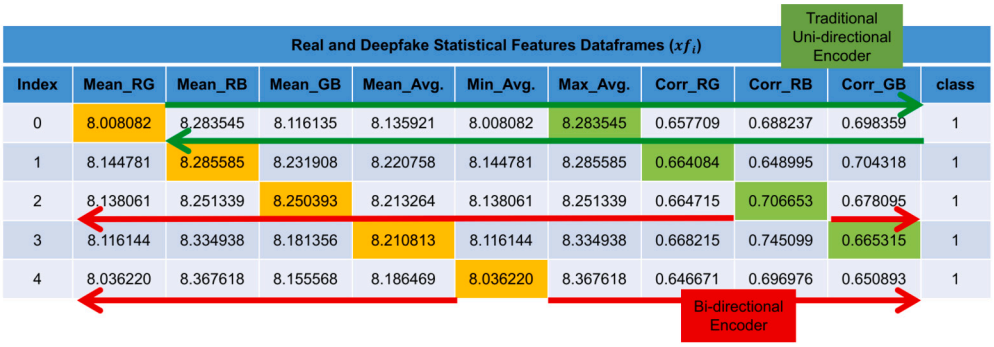


Fig. 7. The LTE bi-directional language modeling.

Table 1
Comprehensive Details about The CTE Model Parameters.

Stages	Layer Name	CTE	Output Size
Stage 1	**Conv. Embedding**	7 × 7, 64, **stride** 4	56 × 56
	Conv. Projection	[3 × 3, 192] × 2	
	MHSA	[H1 = 1 , D1 = 64] × 2	56 × 56
	MLP	[R1 = 4] × 2	
Stage 2	**Conv. Embedding**	3 × 3, 192, **stride** 2	28 × 28
	Conv. Projection	[3 × 3, 768] × 2	
	MHSA	[H2 = 3 , D2 = 192] × 2	28 × 28
	MLP	[R2 = 4] × 2	
Stage 3	**Conv. Embedding**	3 × 3, 384, **stride** 2	14 × 14
	Conv. Projection	[3 × 3, 1024] × 2	
	MHSA	[H3 = 6 , D3 = 384] × 2	14 × 14
	MLP	[R3 = 4] × 2	
Classifier	**Linear**	2	1 × 1
Complexity	4.53 GFLOPs		

by a convolutional projection layer and then MHSA is applied again to capture global contextual dependencies. Notably, this stage incorporates ten MLP layers to further process the refined features. Following the three stages, the CTE model concludes with a head layer responsible for classification or output feature embeddings F_{x_0} extraction. The output size of this layer is 1×1, and it employs a linear transformation. In the case of the CTE model employed as an individual classification network, the resulting output consists of two classes, representing the classification categories of real and deepfake.

3.2.4. CTE model parameters and specifications

The CTE adopted the convolution vision transformer (CvT-13) model architecture [48]. The CTE model encompasses a total of 19.98 million parameters. Also, we implement the Vanila CTE model for the deepfake detection evaluation. Table 1 gives comprehensive details about the CTE parameters.

3.3. Language transformer-based encoder

The LTE model is derived from a distilled version of the BERT model [54], previously mentioned as DBT [49]. Unlike most language models that are unidirectional and can only traverse a context window of features from right to left or vice versa, the LTE model employs bi-directional language modeling. This approach allows for a more holistic view of the feature sequence, enabling the simultaneous consideration of all features on either side of a given feature value in contextual language modeling. As illustrated in Fig. 7, this offers a distinct advantage over unidirectional models. The LTE model consists of six transformer blocks, each block consisting of a multi-head self-attention mechanism and a fully connected feed-forward network layer. A residual connection [52] and layer normalization [53] are used around these sub-layers to maintain consistency. Specifically, the function $Sublayer(xf_i)$ implemented by each sub-layer produces an output of $LayerNorm(xf_i + Sublayer(xf_i))$, with the output dimensions of 512 to facilitate the residual connections between layers. These blocks are then stacked atop one another to form the overall architecture of the encoder model. The layer operations can be viewed in Fig. 8, and further details are as follows.

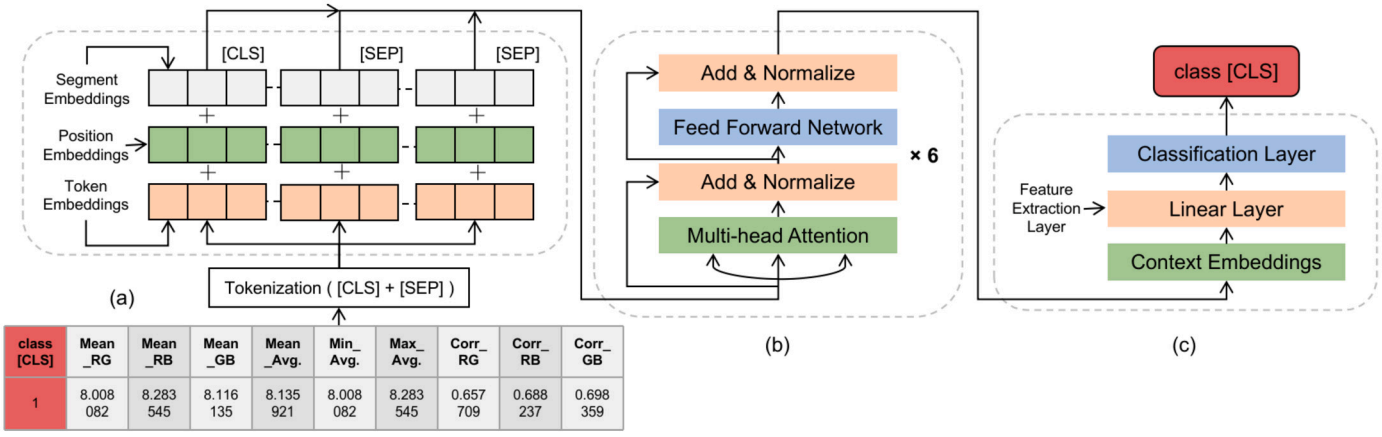
3.3.1. Embedding layer

We take the input spectral features xf_o extracted from a video frame x_o and transform them into token embeddings and subwords using a DBT-based tokenizer [49], which utilizes WordPiece embeddings [55] and has a vocabulary of 30522 tokens. This process, as shown in Fig. 8-(a), helps us capture the contextual and semantic meaning of the input tokens and enables us to unambiguously represent a sequence of feature values in one token sequence. The initial token in each sequence is consistently a distinctive classification token $[CLS]$, and its associated final hidden state serves as the comprehensive illustration of the sequence feature for classification. We apply segment embeddings to feature tokens to distinguish between two different sequences of tokens in a single input. We differentiate the sequences of features in two manners: by isolating them with a special token $[SEP]$ and adding an embedding to indicate their sequence ($AorB$). Our input embedding is depicted as E (Fig. 9), with the final hidden vectors of the special token and the i-th input token as $C \in R^H$ and $T_i \in R^H$, respectively. Additionally, we incorporate position embeddings along with token and segment embeddings to indicate token sequence position. This step involves combining the token, segment, and position embeddings to construct the input representation of a token. As shown in Fig. 9, this construction allows us to grasp the meaning of input tokens and provide a more accurate representation for classification and feature extraction tasks.

3.3.2. Transformer encoder

The LTE model consists of a series of transformer blocks that function as the basic components of the encoder model. Each transformer block [56] has two sub-operational layers: a self-attention layer and a feed-forward neural network layer, as shown in Fig. 8-(b). The self-attention layer in each transformer block facilitates the model to concentrate selectively on various portions of the input sequence in order to capture contextual information and dependencies between tokens. This approach provides greater flexibility and accuracy in modeling complex relationships within the input data. Meanwhile, the attention output undergoes a non-linear transformation through the feed-forward neural network layer.

The LTE architecture relies on the self-attention mechanism as a key component. Specifically, it facilitates the model to grasp the interrelationships between tokens in the input feature series. Through self-attention, the model calculates attention scores between each pair of tokens, which are then utilized to calculate the weighted sums of the token embeddings. These attention scores reflect the relative importance or relevance of each token within the feature sequence. The self-attention process is carried out in parallel multiple times using multi-head attention ($MHSA_{\mathrm{LTE}}$). This enables each attention head



class [CLS]	Mean_RG	Mean_RB	Mean_GB	Mean_Avg.	Min_Avg.	Max_Avg.	Corr_RG	Corr_RB	Corr_GB
1	8.008082	8.283545	8.116135	8.135921	8.008082	8.283545	0.657709	0.688237	0.698359

Fig. 8. The Language Transformer Based Encoder (LTE) - model architecture. (a) Embedding Layer (b) Transformer Encoder (c) Classification Head or Output Feature Embedding Extractor.

class [CLS]	Mean_RG	Mean_RB	Mean_GB	Mean_Avg.	Min_Avg.	Max_Avg.	Corr_RG	Corr_RB	Corr_GB
1	8.008082	8.283545	8.116135	8.135921	8.008082	8.283545	0.657709	0.688237	0.698359

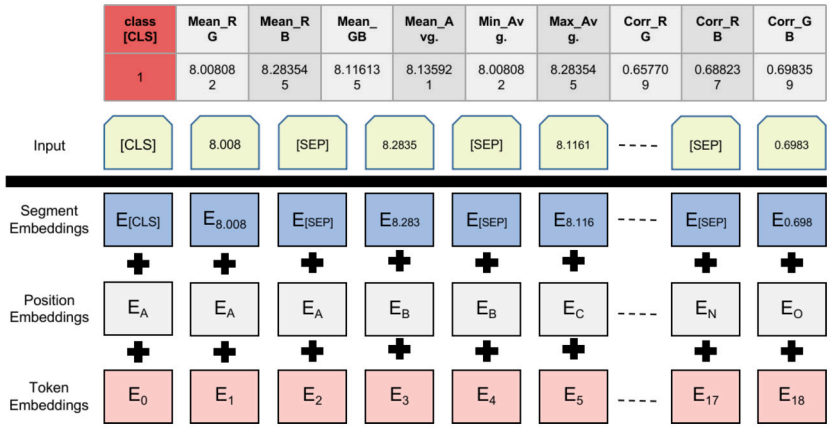


Fig. 9. Input embedding illustration for LTE. The input embeddings for LTE are the sum of token, segmentation, and position embeddings.

to capture different aspects of the input sequence, creating various representations of the context. The $MHSA_{\mathrm{LTE}}$ approach is advantageous as it projects queries, keys, and values times using learned linear projections for dimensions of d_k, d_k, and d_v. The attention function is employed alongside individual projection, generating output values with the dimension of d_v. Afterward, these values are combined and projected again, concluding in the final values illustrated in Fig. 10-(b).

Assuming that q and k are independent random variables with a variance of 1 and mean of 0, the dot product, $q \cdot k$ is $\sum_{i=1}^{d_k} q_i k_i$, of the two, has a mean of 0 and a variance of d_k. This explains why the dot products get larger. The multi-head attention process enables the model to heed cues from various illustrations of subspaces, each focusing on different positions. It overcomes the limitation of a single attention head, which tends to average out the information.

$$MHSA_{\mathrm{LTE}}(Q,K,V) = Concat\left(head_1,\ldots,head_h\right)W^O \tag{21}$$

where head in $_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right)$

In (21), the parameter matrices of projections are $W^O \in \mathbb{R}^{hd_v \times d_{\mathrm{model}}}$, $W_i^Q \in \mathbb{R}^{d_{\mathrm{model}} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{\mathrm{model}} \times d_k}$, and $W_i^V \in \mathbb{R}^{d_{\mathrm{model}} \times d_v}$. The multi-head attention is achieved by employing scaled dot-product attention multiple times, as depicted in Fig. 10-(a). The input includes queries and keys with dimension d_k and values with dimension d_v. We compute the dot products of the query with all keys, next, divided by $\sqrt{d_k}$, and then, finally, we determine the weights on the values by passing it through a SoftMax function. The attention function is typically calculated on a matrix of queries (Q) in practice. The keys and values matrices, K and

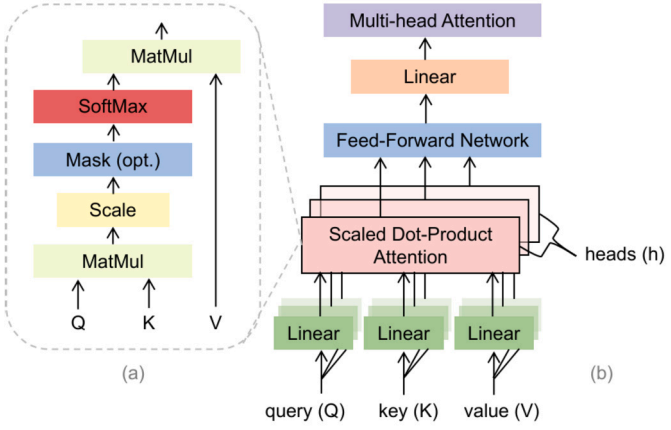


Fig. 10. (a) Scaled Dot-Product Attention, and (b) The LTE Multi-Head Attention ($MHSA_{\mathrm{LTE}}$) is comprised of multiple attention layers that operate simultaneously.

V, are also combined. The output matrix is computed according to the following procedure in (22).

$$Attention_{\mathrm{LTE}}(Q,K,V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{22}$$

In the LTE model, the attention output is subjected to a fully connected layer and point-wise feed-forward network layer, which adds non-linearity to the model and enables it to capture intricate patterns

Table 2
LTE Model Parameters.

Parameters	LTE
Number of layers (transformer blocks)	6
Total number of self-attention heads	12
Number of hidden units	768
Total number of trainable parameters	66 Million
Complexity	17.92 GFLOPs

and relationships. In our encoder, each layer has a feed-forward network that is utilized identically in every position, in addition to the attention sub-layers. This feed-forward network incorporates two linear operations isolated by a ReLU activation function.

$$FFN_{\mathrm{LTE}}\left(xf_0\right) = \max\left(0, xf_0 W_1 + b_1\right) W_2 + b_2 \tag{23}$$

While the linear operations remain consistent across various positions, the parameters used differ from layer to layer. This can also be understood as two convolutions with a kernel size of 1.

3.3.3. Normalization, pooling, and output layers

Further, layer normalization is applied after each sub-layer, such as the feed-forward and self-attention layers. Normalizing the output of the sub-layer is essential as it leads to more stable training and better generalization. The LTE model also includes a pooling layer that aggregates the representations of all the tokens into a fixed-size representation. The pooled representation captures the meaning of the input feature sequence as a whole. The output layer is typically task-specific, such as feature embedding extraction or classification. Here, the LTE output embeddings F_{xf_0} are extracted and combined with the CTE model for joint analysis through a common decoder.

3.3.4. Language encoder model parameters

The LTE model has around 66 million parameters, which is approximately 40% of the size of the BERT base model. Table 2 gives more details about the model parameters.

3.4. Weighted feature embeddings fusion

The weighted feature embeddings fusion mechanism, illustrated in Fig. 1, is implemented by combining the CTE and LTE output embeddings, F_{x_i} and F_{xf_i}, following the acquisition of multi-modality representations through the input video sequences x_i and respective statistical features xf_i in batches to the convolution and language encoders of the SVFT model, respectively. This layer facilitates a comprehensive comprehension of real and deepfake characteristics, thereby enhancing detection performance.

Additionally, in the feature fusion process, weights are utilized to blend the features from each encoded output and regulate their relative significance in the final representation. These weights (α, β) are scalar values that can be adjusted based on the assigned importance of each modality. Here, α represents the weighted representation of CTE, which is 0.7, and similarly, β represents the LTE weighted value of 0.3. The weighted fusion aggregation of the encoded outputs is determined by (24).

$$\mathrm{FF} = \left(\alpha \times \mathrm{F}_{x_i}\right) + \left(\beta \times \mathrm{F}_{xf_i}\right) \tag{24}$$

3.5. SVFT decoder and classification head layers

The SVFT decoder model [56], as mentioned in Fig. 1, utilizes fused feature embeddings (FF) from convolution and language encoder models. It then applies common cross-attention using the decoder and the final classification output is generated using the classification head layer. The step-by-step functioning procedure is outlined as follows.

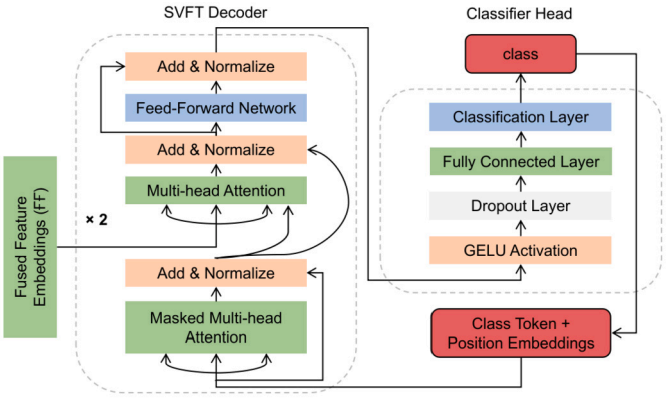


Fig. 11. SVFT Decoder and Classification Head Layers.

3.5.1. SVFT decoder layer

The decoder layer in the SVFT model plays an integral function in processing the input fused features FF and generating an output representation for the classification head. The decoder layer applies two operations to the fused features in succession, such as the self-attention mechanism and feed-forward network. The self-attention helps the model to concentrate on distinct parts of the FF cues, capturing dependencies and relationships between different deepfake characteristics and cues. The feed-forward neural network operations include nonlinear activations and linear transformations. This allows the model to perform complex computations and generate a more informative and higher-level discriminative representation of the FF cues. Here decoder layer parameters such as output feature vectors have a dimensionality of 2 with two attention heads. The operations in the decoder layer and classification head layer of the SVFT model are shown in Fig. 11. In the SVFT decoder, the multi-head self-attention process computes attention scores (α) for each position i based on the given input FF cues, as follows:

$$\alpha_i = \mathrm{Softmax}\left(\frac{Q_i \cdot K_i^T}{\sqrt{d_k}}\right) \tag{25}$$

Here, Q_i and K_i represent the query and key embeddings for position i, and the key embeddings dimension by d_k. The attention scores (α) are then utilized to compute the weighted aggregate of value embeddings (V) to obtain the attended output $\left(O_i\right)$ for position i:

$$O_i = \sum\left(\alpha_i \cdot V_i\right) \tag{26}$$

In the next SVFT decoder operation, layer normalization (LN) is applied after the self-attention operation to normalize the intermediate representations. Given an input FF cues, the layer normalization operation is defined as:

$$LN(\mathrm{FF}) = \gamma\left(\frac{\mathrm{FF} - \mu}{\sqrt{\sigma^2 + \epsilon}}\right) + \beta \tag{27}$$

Here, learnable scales and shift parameters γ and β are used alongside the mean and standard deviation of FF(μ and σ) and a small constant ϵ for numerical stability.

At the last stage of operation in the decoder, the FF cues are passed to a feed-forward network (FFN). FFN includes two linear transformations isolated by a GELU activation function.

$$FFN(\mathrm{FF}) = GELU\left(\mathrm{FF}.W_1 + b_1\right) W_2 + b_2 \tag{28}$$

Here, FF represents the input features to the feed-forward network, GELU is the Gaussian Error Linear Unit activation function, b_1 and b_2 are biasing vectors, and W_1 and W_2 are weight matrices.

Table 3
SVFT Layer Modules and Their Parameters.

Layer Modules	Total Parameters	Trainable Parameters
LTE	66,955,010	66,955,010
CTE	19,613,250	19,613,250
Decoder Layer	10,302	10,302
Classifier Layer	6	6
GELU Activation	0	None
Dropout	0	None
Total	86,578,568	86,578,568
Complexity	25.89 GFLOPs	

3.5.2. Fully connected and classification layers

The fully connected and classification layers in the SVFT model work together to transform the features from the decoder layer into a suitable representation for classification. These layers aim to capture the necessary information and make accurate predictions for the given classification task by applying linear transformations, non-linear activations, and regularization techniques like dropout. The fully connected layer uses a linear transformation to grasp non-linear affinities between fused features FF in a lower-dimensional space. The fully connected layer has an input dimension of 2, corresponding to the resultant dimensionality of the decoder layer and representing the number of output classes. After the fully connected layer, the output goes through a GELU activation function and a dropout layer. The GELU function introduces non-linearity, which can help capture complex patterns in the feature data. The GELU activation function [57] is defined as:

$$\mathrm{GELU(FF)} = 0.5\left(1 + \mathrm{erf}\left(\frac{\mathrm{FF}}{\sqrt{2}}\right)\right) \tag{29}$$

Here, FF represents the input cues to the GELU activation function. During training, the dropout layer plays a crucial role in preventing overfitting and enhancing generalization by randomly setting a fraction of the input elements to zero. This approach mitigates network reliance on specific input features and ensures it performs well on new, unseen data, dropout operation is,

$$\mathrm{FF}_{\mathrm{drop}} = \mathrm{Dropout}\ (p)(\mathrm{FF}) \tag{30}$$

Here, p represents the probability of a component being zeroed. Finally, the output goes via an additional linear layer to generate the logits of each class. The fully connected layer parameters depend on the input and output sizes, precisely the number of output classes. The linear layer in the SVFT model uses a linear transformation to the input tensor and is defined as:

$$y = \mathrm{Linear}\ \left(\mathrm{FF}_{\mathrm{drop}}, W, b\right) = \mathrm{FF}_{\mathrm{drop}}\, W^T + b \tag{31}$$

Here, $\mathrm{FF}_{\mathrm{drop}}$ represents the input feature embeddings, y denotes the output class probability, b represents the bias vector, and W represents the weight matrix. The GELU activation function and dropout layer do not have trainable parameters but contribute to the overall model behavior.

3.5.3. SVFT model parameters and specification

The SVFT model parameters represent the count of learnable parameters in the network architecture. These parameters are the values that the model adjusts throughout the training cycle to optimize its performance for deepfake classification tasks Table 3 provides a comprehensive breakdown of all parameters. The decoder layer has 10302 parameters. These parameters are utilized for the decoder component of the model, which applies cross-attention and transforms the fused features. The GELU activation function does not introduce any learnable parameters. The decoder output undergoes this non-linear activation function. Similarly, the dropout layer does not introduce any new learnable parameters, it has a dropout rate of 0.25. It randomly sets elements of the input tensor to zero during training, which helps in preventing overfitting. The fully connected layer has six parameters. These parameters include weights and biases and are responsible for transforming the decoder output to the desired output classes. There are 86,578,568 trainable parameters in the SVFT model.

4. Experimental settings

4.1. Datasets

Experiments were performed on four prominent benchmark datasets to evaluate the efficacy of the proposed SVFT model: such as FaceForensics++ (FF++) [50], Deepfake Detection Challenge (DFDC) [58], Celeb-DF-v2 (CDF) [59], and DeeperForensics-1.0 (Deeper) [60]. A total of 1000 videos were selected from each dataset and partitioned into distinct subsets, training, validation, and test, in distribution by following the [50]. The training, validation, and testing datasets contain 720, 140, and 140 videos, respectively.

FF++: FF++ is an early and widely recognized video dataset that comprises 1,000 authentic face videos sourced from YouTube and 4000 associated manipulated videos. Each real video clip is associated with four types of forgeries: DeepFakes (DF), Face2Face (F2F), FaceSwap (FS), and NeuralTextures (NT). Various compression ratios are applied to the videos, resulting in subsets of raw quality (raw or c0), high-quality (c23) with mild compression, and low-quality with heavy compression (c40) videos. Given its diverse content and extensive scale, FF++ is commonly utilized for model training.

DFDC: DFDC stands out as one of the enormous Deepfake datasets, comprising 104,500 fake videos and 23,654 authentic videos captured by 3426 actors and actresses. The fake videos exhibit remarkably realistic appearances.

CDF: The CDF dataset contains 5639 high-quality deepfake and 590 authentic videos. The fake videos are forged using improved methods of face swapping, resulting in exceptionally realistic face transformations. The dataset benefits from enhanced facial appearance.

Deeper: Deeper is one of the enormous datasets publicly available for real-world face manipulation detection. This dataset enhances the original videos from FF++ by applying advanced tampering techniques to change facial identities. A careful selection of videos ensures diversity and quality within the dataset.

4.2. Evaluation metric

We utilize the Area Under the Receiver Operating Characteristic Curve (AUC) as an evaluation metric following the previous works [26], [27], [29], [30], [31], which is standard practice in most studies. AUC reflects the overall performance under different detection thresholds, independent of the threshold values.

4.3. Training and testing settings

This section discusses the details of the parameters used in the CTE, LTE, and SVFT models training and testing. The CTE model also serves as the baseline model for the deepfake classification task, and we implemented it as a separate method for comparison with our SVFT model.

Models Initialization: The CTE model was initialized from the pre-trained CvT [48] backbone weights trained on ImageNet1k [61]. This choice was motivated by the success of CvT in capturing hierarchical features in image data, as demonstrated through its strong performance in image understanding tasks. All parameters of the CTE model were fine-tuned on deepfake datasets using the data transformations concerning the mean and standard deviation of training datasets, Table 4 provides the details. Similarly, the LTE framework is based on the DBT base model [49], with its backbone initialized using the DBT weights. The DBT base model is pre-trained on a substantial amount of general-purpose knowledge, utilizing the BooksCorpus (800 million words) and

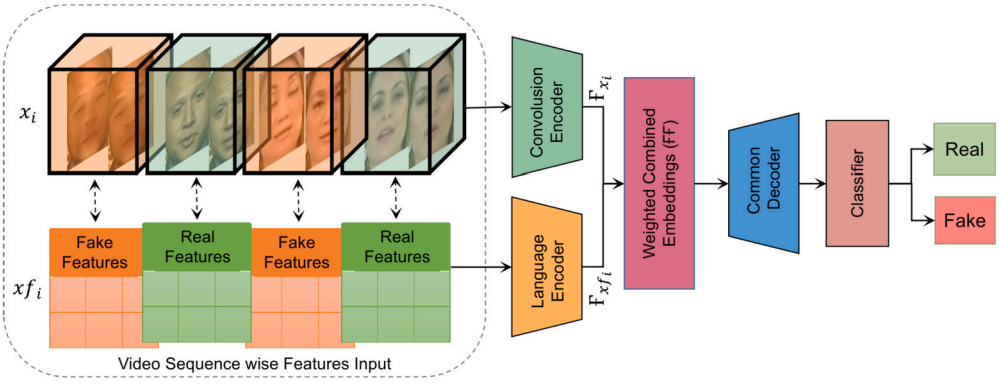


Fig. 12. Visual Depiction of Structured Training Process.

English Wikipedia (2,500 million words). To align the LTE with our specific deepfake detection task, it underwent further training and fine-tuning on deepfake datasets. Finally, the SVFT model was initialized with two pre-trained encoders (CTE and LTE), both trained on deepfake datasets. The model also incorporates a standard transformer decoder layer with a classifier layer, initialized with random weights, ensuring adaptability to our task.

Training and Testing Hyperparameters: The CTE model underwent training with the AdamW optimizer over a total of 30 epochs. A learning rate of $5e^{-5}$ was carefully selected through a grid search, balancing convergence speed and stability. To dynamically adjust the learning rate during training, a CosineAnnealingLR scheduler was employed, utilizing a cosine annealing schedule with $T_{max} = 30$, $eta_{min} = 0$, and $last_{epoch}$ =-1. The choice of a batch size of 128 for training and validation loaders, and 512 for test loaders, was driven by computational efficiency considerations. In the LTE branch, most hyperparameters mirrored those of CTE, with the exception of a weight decay of 0.001, strategically employed for regularization. The SVFT model inherited hyperparameters from CTE and LTE, maintaining consistency, except for a learning rate of 0.001 and a weight decay of 0.01, carefully chosen to suit the unique characteristics of the fusion model. These hyperparameter choices were guided by empirical experiments, ensuring a balance between model convergence, generalization, and regularization.

Training Strategy: The CTE and LTE backbones and SVFT model training process were executed using the Fabric, which enables distributed training using the lightning framework of Pytorch. The barrier synchronization point is used for the distributed training setup, ensuring coordinated execution across multiple processes. Further, the LTE training process strategy involves tokenization and numericalization before fine-tuning the model. In the first step, we tokenized the frequency domain statistical data using the DBT-based tokenizer with the maximum length of input allowed, 512, and the vocabulary size of 30522. After tokenization, the feature data was numerically encoded to prepare it for model input. The numericalization is performed on the tokenized data using the same tokenizer. The data is converted into a torch-compatible format, including input IDs, attention masks, and labels. Like the steps mentioned, the SVFT model adopted the input strategies applied to both CTE and LTE backbone training. Overall, we employed a comprehensive methodology for training CTE, LTE, and SVFT using mixed-precision training with "bf16-mixed" [62], enabling efficient utilization of computational resources while maintaining satisfactory model accuracy and computational cost.

The SVFT Structured Training Process: In the SVFT structured training, we aim to enhance the ability of the model to discern the characteristics that differentiate real and fake videos by exposing the model to both types of data on common subjects. The training process begins by organizing the video sequences and corresponding features, where real videos and their corresponding features are paired and succeeded by the fake video sequence and their features, which are then provided as input to the SVFT model. This permits the model to understand the distinguishing features and patterns that differentiate fake videos from real ones. A visual depiction is shown in Fig. 12. The process of alternating between real and fake video sequences continues throughout the training procedure. This interleaved training scheme approach further enriches the learning process by providing intrinsic information and cues for the model to differentiate between real and fake videos.

Table 4
Dataset Characteristics: Mean and Standard Deviation.

Datasets	Mean	Standard Deviation
FF++	[0.6034, 0.4359, 0.3801]	[0.2232, 0.1801, 0.1705]
DF	[0.6051, 0.4498, 0.3932]	[0.2209, 0.1863, 0.1834]
FS	[0.6035, 0.4361, 0.3806]	[0.2220, 0.1784, 0.1687]
F2F	[0.6050, 0.4372, 0.3815]	[0.2228, 0.1793, 0.1699]
NT	[0.6042, 0.4354, 0.3796]	[0.2246, 0.1806, 0.1707]
DFDC	[0.5127, 0.3541, 0.3195]	[0.2182, 0.1828, 0.1811]
CDF	[0.5424, 0.3566, 0.2991]	[0.2186, 0.1634, 0.1449]
Deeper	[0.5462, 0.3911, 0.3308]	[0.2445, 0.1716, 0.1507]

4.4. Data augmentations and transformations

The proposed SVFT model utilizes RetinaFace [63], a single-stage face detector, to extract the input facial frames from videos. Additionally, we employ the DLib face detector with 81 facial landmarks to ensure that the detected facial area encompasses all landmark points. Frames that do not meet this criterion are discarded. The facial video sequence is then resized to 224×224 to accommodate the CTE model, which serves as one of the backbones of the SVFT framework. Two sets of data transformations are used for training, validation, and test datasets.

The first transformation type aims to augment the training data and involves random horizontal flipping, random cropping, random resizing, conversion to tensor, and normalization. Random resizing scales the input image randomly, while random cropping extracts a 224×224 pixel patch from the resized frame. Afterward, the frame is transformed to a PyTorch tensor and normalized utilizing the mean and standard deviation values prescribed in Table 4. The second transformation process is performed on the validation and test datasets, including conversion to tensor and normalization by employing the same mean and standard deviation values. This approach ensures that the validation and test datasets undergo the exact normalization strategy as the training dataset, thus ensuring consistency in the computed metrics.

4.5. Loss function

During the training of all three models, CTE, LTE, and SVFT, the binary cross entropy loss ($BCEL_{SVFT}$) [64] was computed for each ex-

ample in the training datasets, and the average loss across all examples were employed to correct the model parameters through the backpropagation process. The objective is to reduce the loss function value to enhance the accuracy of the model in binary classification tasks.

$$BCEL_{SVFT} = -(y \cdot \log(p) + (1-y) \cdot \log(1-p)) \quad (32)$$

where y represents the true binary label and the p in $[0, 1]$ symbolizes the predicted probability for the positive class.

5. Results and evaluations

In order to assess the generalization capability of our proposed deepfake models, we executed various experiment scenarios using cross-manipulation and cross-dataset setups. This allowed us to simulate unseen forgery methods and datasets separately. Specifically, we evaluated the detection ability and robustness of our model under different compression settings and against unseen perturbations. We conducted within and cross-manipulation evaluations using the FF++ dataset [50] and its subsets, and cross-dataset performance evaluations were performed by training the models on FF++ [50] and testing on DFDC [58], CDF [59], and Deeper [60] datasets. Furthermore, we evaluated compression-based performance on the FF++ manipulations subsets and analyzed against four unseen perturbations using the FF++ dataset (Brightness Change, Contrast Change, Random Block Occlusion, and Gaussian Blur). Methods with the "※" symbol are trained using publicly provided codes to produce results under the exact environment as ours for comparison.

5.1. Evaluation on unseen cross manipulations

We aim to demonstrate the significance of our proposed baseline CTE and SVFT methods by testing their generalization capabilities on the cross-manipulation method in the FF++ dataset [50]. The dataset comprises four deepfake attack methods - F2F, FS, DF, and NT - each yield significantly different results. In real-world scenarios, the deepfakes are usually unknown. To replicate such situations, we split the FF++ datasets into target and reference based on distinct manipulation techniques. We then trained our deepfake detection models on each subset of the FF++ dataset and tested them on all the remaining sub-datasets. We used the raw version of FF++ for training and testing and performed a comparison of our method performance with six state-of-the-art methods. Table 5 displays the comparison results. We also showed the learned features embedding space by the LTE branch with the t-SNE [65] feature distributions in Fig. 4, illustrating the clustering of the learned features.

The findings presented in Table 5 demonstrate that our proposed methodology consistently outperforms its competitors in most cross-manipulation settings by a significant margin. Specifically, when trained on the DF dataset and evaluated on FS, F2F, and NT, our approach showcases performance gains of 17.43%, 5.25%, and 12.86%, respectively, in terms of AUC. Moreover, our model remains effective under the four intra-domain settings, which are highlighted in gray. These results suggest that our method is capable of uncovering adaptive frequency features and establishing connections between essential forged clues in the spatial and frequency domains through joint learning. Although some instances show that our SVFT method slightly underperforms when trained on F2F and NT, it is likely because the manipulation patterns only exist in certain small regions. On the other hand, our CTE method outperforms the latest methods in many cases such as SFDG [27] and RECCE [30], while CTE only utilizes the visual modality, highlighting the effectiveness of incorporating frequency domain statistical modality to aid visual modality in deepfake detection. Our SVFT method achieves better results compared to methods utilizing statistical-visual multimodalities, indicating its ability to effectively capture fine-grained statistical-visual inconsistencies and explore correlations in non-critical phoneme-viseme regions with better generalization capability against unseen forgery methods. This aspect is crucial in practical applications, emphasizing the importance of our proposed methodology.

Table 5
Our Proposed Methods are Evaluated for Intra and Cross-manipulation on the FF++ Dataset. We compare our Methods with the state-of-the-art Methods in terms of AUC (%).

Train	Methods	Test				Avg.
		DF	FS	F2F	NT	
DF	Xception※[22]	99.32	49.05	73.60	73.61	**73.90**
	Face X-ray※[25]	98.71	63.32	60.06	69.82	**72.98**
	SPSL※[26]	99.35	48.14	67.86	73.88	**72.31**
	MultiAtt [29]	99.92	40.61	75.23	71.08	**71.71**
	RECCE※[30]	99.19	57.42	74.39	85.04	**79.01**
	SFDG [27]	99.73	75.34	86.45	86.64	**87.03**
	CTE (Ours)	**99.70**	**73.00**	**78.00**	**90.10**	**85.70**
	SVFT (Ours)	**99.99**	**92.77**	**91.70**	**99.90**	**96.09**
FS	Xception※[22]	66.45	99.40	88.83	71.32	**81.50**
	Face X-ray※[25]	63.02	98.44	93.83	94.57	**87.47**
	SPSL※[26]	46.42	99.93	98.90	97.80	**85.76**
	MultiAtt [29]	64.13	99.67	66.39	50.10	**70.07**
	RECCE※[30]	66.66	99.76	73.66	57.46	**74.39**
	SFDG [27]	81.71	99.53	77.30	60.89	**79.85**
	CTE (Ours)	**76.40**	**99.91**	**85.60**	**74.40**	**84.08**
	SVFT (Ours)	**84.10**	**99.99**	**91.78**	**81.93**	**89.70**
F2F	Xception※[22]	80.33	76.25	99.47	69.66	**81.43**
	Face X-ray※[25]	45.82	96.12	98.15	95.76	**83.96**
	SPSL※[26]	60.06	98.58	99.48	98.70	**89.21**
	MultiAtt [29]	86.15	60.14	99.13	64.59	**77.50**
	RECCE※[30]	88.04	67.35	98.93	74.16	**82.12**
	SFDG [27]	97.38	73.54	99.36	72.61	**85.72**
	CTE (Ours)	**81.80**	**76.28**	**99.20**	**80.90**	**84.05**
	SVFT (Ours)	**84.90**	**92.90**	**99.92**	**82.00**	**89.68**
NT	Xception※[22]	79.98	73.17	81.36	99.15	**83.42**
	Face X-ray※[25]	70.51	91.77	91.03	92.54	**86.46**
	SPSL※[26]	57.01	99.43	99.67	99.90	**89.00**
	MultiAtt [29]	87.23	48.22	75.33	98.66	**77.36**
	RECCE※[30]	90.20	58.06	76.65	97.17	**80.52**
	SFDG [27]	91.73	83.58	70.85	99.74	**86.47**
	CTE (Ours)	**94.10**	**80.50**	**79.80**	**99.95**	**88.58**
	SVFT (Ours)	**92.00**	**92.80**	**91.70**	**99.99**	**94.87**

5.2. Evaluation on cross-dataset settings

This section evaluates our methods in a cross-dataset setting to assess the generalization performance. Cross-dataset evaluation can be challenging in realistic scenarios due to the unfamiliar domain void between the training and testing sets. For deepfake detection models, domain generalization ability is an essential criterion. To this end, we conducted cross-dataset experimentation wherein we trained proposed detection models on FF++ [50] (raw), which includes all four types of manipulation datasets, and evaluated them on three unseen datasets, namely DFDC [58], CDF [59], and Deeper [60]. Table 6 reports the AUC results of several methods. Our approach achieves promising results compared to recent methods on all three unseen datasets with unknown post-processing distortions. Furthermore, the t-SNE feature distributions presented in Fig. 4 illustrate the embedding space of the learned features through the LTE branch. Our approach exhibits more compact feature distributions for intra-class (real or fake) instances when compared to other existing methods. The inter-class instances are also more distinguishable, which substantially enhances the AUC when assessing unknown deepfakes.

In particular, the CDF [59] and DFDC [58] datasets are considered challenging due to the various manipulations and perturbations used to generate deepfake videos. Our SVFT approach shows significant improvement in generalization performance, with a 7.94% and 2.29% increase in the AUC metric compared to the second-place SPSL [26]

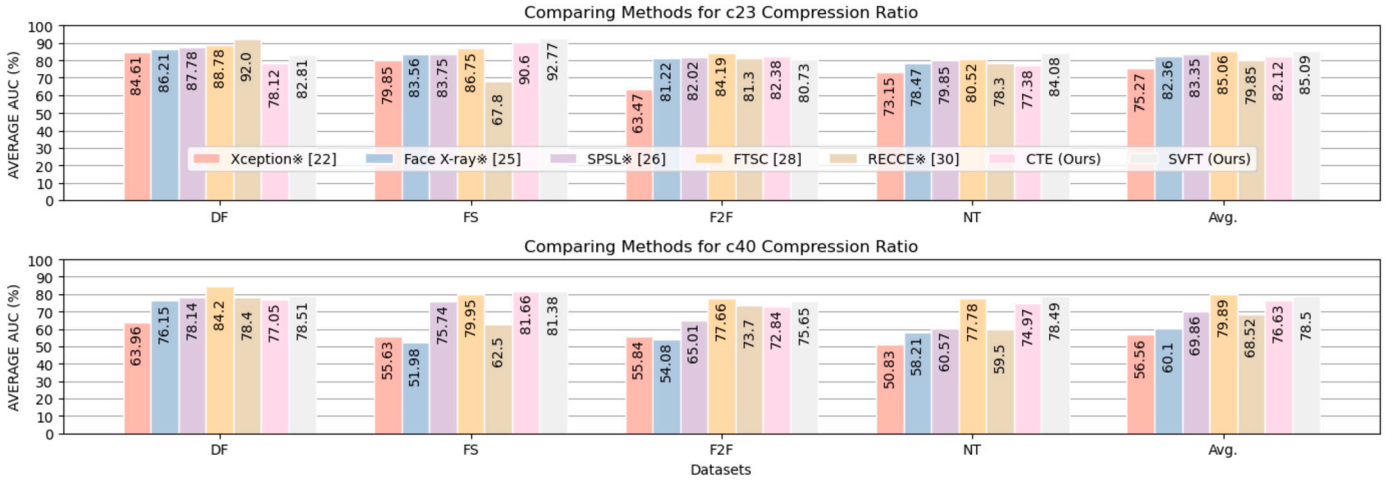


Fig. 13. Evaluation of our Proposed Method against Compressed Versions of the FF++ Dataset (c23 and c40). Results are compared with state-of-the-art methods in terms of AUC (%).

Table 6
Our Proposed Methods are Compared to State-of-the-art Methods in terms of AUC (%) on Cross-Dataset Evaluation of DFDC, CDF, and Deeper Datasets.

Train	Methods	Test			Avg.
		DFDC	CDF	Deeper	
FF++ (raw)	Xception※[22]	67.90	59.46	69.81	**65.06**
	Face X-ray※[25]	70.01	74.20	72.30	**72.17**
	SPSL※[26]	75.56	76.88	71.41	**74.62**
	MultiAtt [29]	69.56	67.44	67.34	**68.11**
	RECCE※[30]	68.34	68.94	88.70	**75.32**
	CPT [31]	73.68	72.43	78.19	**74.76**
	SFDG [27]	73.64	75.83	92.10	**80.52**
	CTE (Ours)	**79.92**	**77.72**	**73.18**	**76.94**
	SVFT (Ours)	**83.50**	**78.12**	**80.27**	**80.63**

and SFDG [27] models, respectively, as stated in Table 6. Unexpectedly, the baseline CTE method marginally outperformed the others on the DFDC and CDF datasets. On the Deeper dataset, our baseline CTE and SVFT approaches underperformed in comparison with the state-of-the-art from RECCE [30], CPT [31], and SFDG [27] methods, with the latter showing the best detection performance, at least 3.40% on average from the second-place RECCE [30]. Overall, our method performed better on average in the cross-dataset domain. It is noteworthy that multimodal deepfake detection methods generally produce better results than unimodal methods. There are several possible reasons to explain these results.

The results obtained can be attributed to three primary factors. (1) The supervised learning approach effectively utilizes intra and cross-category cues. (2) The multi-model feature learning method can integrate the internal affinities of global and local regions, thereby acquiring more thorough indications concerning intrinsic cues. (3) The RGB cues and high-frequency statistical anomalies are utilized to enhance the generalization capability of the model to counteract interference from varying qualities. These reasons make our SVFT method competitive and superior compared to existing methods, which tend to overfit specific forgery patterns presented in the training samples. The experimental results strongly indicate the importance of exploiting spatial, temporal, and frequency domain information. Our method is more likely to tap into subtle inconsistencies and intrinsic complementary features with content-aware semantic attention. The coordinated analysis and interaction between spatial and frequency domain statistical values help in reasoning about generally forged cues, thus enhancing the generalization performance of our proposed SVFT model on unseen deepfakes.

5.3. Evaluation on compression manipulations

The impact of compression on deepfake video detection is significant as it can hinder the development of a generalized approach. Therefore, we evaluate the robustness performance of our models in scenarios with two ratios. The evaluation comparison is reported in Fig. 13. Our models were trained on the FF+ (raw) [50] dataset and evaluated using c23 and c40 compression to simulate practical scenarios. However, most research works use videos with compression factors c23 and c40 for training, which could lead to improved performance.

As shown in Fig. 13, many methods experience a significant decrease in accuracy when it comes to cross-compression detection. This applies to methods such as [22,25,26,28,30]. Our proposed method, however, surpasses these methods by a notable margin. This is due to the fact that our SVFT model utilizes two streams. The first stream is a frame-level convolution transformer encoder network. The second stream is a frequency spectrum-based network that incorporates spectral-dependent statistical features resistant to compression. Thus, our method is able to generate robust results that are not affected by compression. Our proposed SVFT method exhibits superior performance on c23 videos with a higher number of forgery artifacts, as opposed to c40 videos which tend to lose more artifacts due to more extensive compression resulting in fewer forged traces. In the FaceSwap (FS) setting, our methods achieve an AUC of 92.77% and 81.66% (trained on FF++ raw and evaluated on c23 and c40), surpassing the second-best performing method FTSC [28] by a significant gap of 6.02% and 1.71%, respectively. Notably, FTSC [28] is explicitly designed to detect compressed videos. Although our SVFT performs better in the c23 category in the NeuralTexture set, it loses to FTSC [28] by a small margin on the average AUC score of c40. However, our method surpasses other recent state-of-the-art approaches in all settings for compression, with an average AUC of 85.09% on c23 and 78.50% on c40. In Deepfake and Face2Face compressed video detection, FTSC [28] and RECCE [30] outperformed our method.

5.4. Robustness evaluation on perturbations

To ensure that a detector possesses good generalization, it should also be able to withstand expected video degradations. As most multimedia content is sourced from the Internet, the data is likely to undergo unfamiliar distortions during transfers. To assess the robustness of our proposed methods against unseen perturbations, we applied four types of distortions - Brightness Change (BC), Contrast Change (CC), Random Block Occlusion (RBO), and Gaussian Blur (GB) - to the FF++ (raw) [50] dataset, creating a perturbations dataset. Using the original FF++ [50] dataset (raw), we trained our models and tested them on

Table 7
Perturbations Types and Intensity Levels.

No.	Perturbation Types	Five Intensity Levels
1	**Brightness Change**	**Factors**: -0.1, -0.2, -0.3, -0.4, -0.5
2	**Contrast Change**	**Factors**: 1.0, 2.0, 3.0, 4.0, 5.0
3	**Random Block Occlusion**	**Number of Blocks**: 5, 10, 15, 20, 25 **Block size**: 20x20, **Fill value**: 0
4	**Gaussian Blur**	**Kernel size**: 3x3, 5x5, 7x7, 9x9, 11x11 **Sigma**: $(0.3 \cdot ((\text{kernel size} - 1) \cdot 0.5 - 1)) + 0.8$

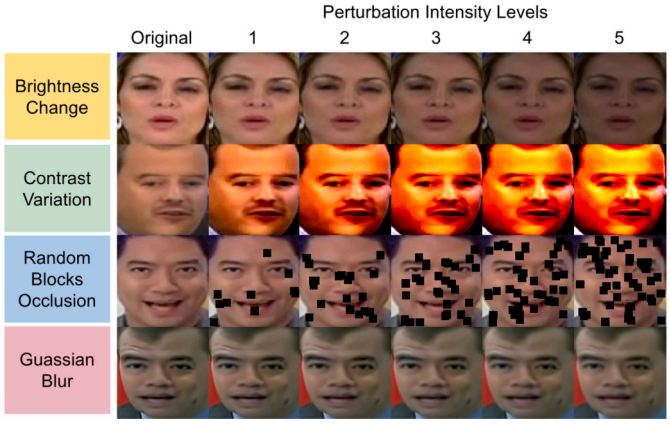


Fig. 14. Demonstration of perturbation effect. From left to right (5 levels): original image, Brightness Change, Contrast Change, Block Occlusion, and Gaussian Blur.

the FF++ [50] dataset with various perturbations. Each perturbation type was applied to five different intensity levels, and the illustrations per level for every disturbance are envisioned in Fig. 14, as detailed in Table 7. Fig. 15 shows the anti-perturbation performance of our models against the state-of-the-art methods with respect to the AUC metric for all intensity levels for each degradation type. The average AUC values of robustness performance of all degradation types are compared against recent state-of-the-art in Table 8.

When evaluating the performance across four disturbance categories, our model demonstrated superior results in two categories: Brightness Change and Block Occlusion. Moreover, we achieved comparable results to the top-performing Xception model [22] in Color Contrast (94.14% versus 98.67%). Notably, when considering Block-wise disturbance, we observed that block masking resulted in a variation in brightness, potentially damaging high-frequency image components. Despite this, the detection performance of our model remained at the forefront due to the interaction of different modalities. It is noteworthy that prior methods experience a significant decrease in performance when confronted with Gaussian blur at higher levels of damage, which destroys frequency statistics. This degradation suggests that accentuating distinct manipulation patterns observed in the training dataset is susceptible to typical perturbations. Our proposed SVFT model surpasses most texture-based techniques due to the inherent complementary cues learned from two streams, providing a sufficient discrimination basis for detection. Moreover, it shows that high-level semantic features are more robust to perturbations. This helped our model outperform their counterparts and exhibit a favorable average performance among the four perturbations, demonstrating its robustness with an AUC of 91.17%.

5.5. Ablation studies

This section presents the ablation as a means to scrutinize the strategies that influence the performance of our presented SVFT framework and training process. Specifically, we develop the following: (1) the Impact of the Weighted Concatenation on the Performance, (2) the Impact of Mixed-Precision Training on Computation Resources, Training time, and Accuracy, (3) the Impact of the LTE Branch on Performance, and (4) Impact of Structure Training Strategy.

5.5.1. Impact of the weighted concatenation on the performance

In the SVFT model, as illustrated in Fig. 1, we use a two-branch framework to assemble an intricate feature representation of visual and statistical cues in the spatial-frequency domain. The spatial branch concentrates on RGB artifacts, while the frequency domain branch focuses more on statistical anomalies and inconsistencies. To mitigate the effects of both branches, we perform an ablation analysis on the values of α and β in Equation (24), as shown in Table 9. The best performance on the DFDC dataset is achieved with a setting of $\alpha = 0.7$ and $\beta = 0.3$. This also suggests that the spatial embeddings are more critical than the statistical embedding to assemble a biased joint representation of spatial-frequency cues, which again establishes that the joint representation emanated from the face manipulation method can function as crucial indications for deepfakes identification.

5.5.2. Impact of mixed-precision training

In this analysis, we compare the training performance of two methods, CTE and SVFT, using different precision modes: 32-bit True Precision and Automatic Mixed Precision (bf16-mixed). We analyze the training time, computation cost, and AUC values for each method and precision mode to determine the effectiveness of automatic mixed precision. We trained the CTE and SVFT models for 30 epochs using both 32-bit True Precision and Automatic Mixed Precision. The training times, computation costs, and AUC values were recorded for each combination. A comparison is depicted in Fig. 16.

The results demonstrate that training with Automatic Mixed Precision offers notable advantages over 32-bit True Precision regarding training time and computation cost. In the CTE case, training time was reduced by approximately 35% when using Automatic Mixed Precision, resulting in a more efficient training process. Additionally, the computation cost decreased by around 3 GB, indicating a reduction in memory usage. Similarly, for SVFT, training time decreased by approximately 24% with Automatic Mixed Precision, leading to faster convergence and model training. The computation cost was also reduced by around 1.2 GB, further optimizing resource utilization. The achieved AUC values of 100% for both precision modes indicate that the discriminative performance of the models remained unaffected by the precision mode used. Further, mixed-precision training with reduced numerical precision leads to faster convergence during the optimization process; secondly, in terms of model stability, the lower-precision data types can introduce a certain level of noise during training. Paradoxically, this noise can act as a regularizer, preventing the model from overfitting to the training data and improving its generalization performance on unseen data, contributing to enhanced model stability without compromising the detection performance.

5.5.3. Impact of LTE branch on performance

To investigate the significance of the frequency domain representation through statistical anomalies via the LTE branch, we can observe the experimental results of different evaluations in Tables 5, 6, 8, and Fig. 13. The frequency domain spectral features adopted in the LTE branch of the SVFT model have helped achieve enhanced generalized detection performance on average AUC metric in the cross-manipulation evaluation, such as 96.42% on Deepfakes compared to single stream-based CTE, which is 85.70%. Further, the trend continues for FS, F2F, and NT, with average gains of 5.62%, 5.63%, and 6.29%, respectively. The accuracy of the deepfake detection task on cross datasets, compression sets, and perturbation sets peek similar patterns.

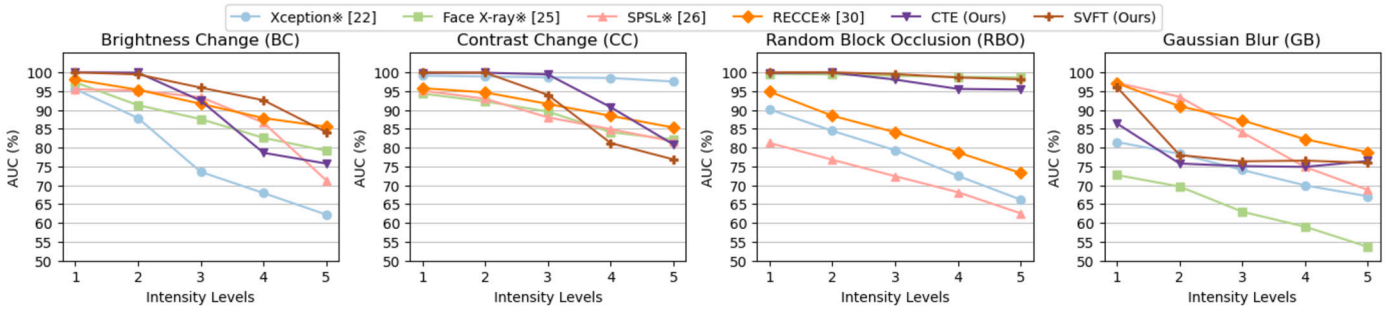


Fig. 15. The AUCs (%) scores over five intensity levels for each perturbation type.

Table 8
Robustness Against Four Types of Unseen Perturbations. The average AUC (%) obtained over five different intensity levels is reported. We also calculate the AUCs over all perturbations for each method.

Train	Methods	Original	Average AUC over Five Levels				Avg.
			BC	CC	RBO	GB	
FF++ (raw)	Xception*[22]	99.30	77.43	98.67	78.52	74.19	**82.20**
	Face X-ray*[25]	99.90	87.60	88.50	99.11	63.80	**87.25**
	SPSL*[26]	98.32	89.49	88.93	72.40	84.61	**83.86**
	RECCE*[30]	99.32	91.74	91.19	83.88	87.29	**88.52**
	CTE (Ours)	**99.99**	**89.35**	**94.14**	**97.76**	**77.73**	**89.75**
	SVFT (Ours)	**99.99**	**94.42**	**90.39**	**99.25**	**80.61**	**91.17**

Table 9
Ablation Study on the SVFT feature fusion layer, specifically on the weights of the spatial domain-based CTE branch (represented by α) and the frequency domain-based LTE branch (represented by β). The evaluation results on the DFDC dataset were measured using the AUC metric (%).

α	0.9	0.8	**0.7**	0.6	0.5	0.4	0.3	0.2	0.1
β	0.1	0.2	**0.3**	0.4	0.5	0.6	0.7	0.8	0.9
AUC	80.08	80.71	**83.50**	80.25	80.37	79.80	79.80	78.53	77.89

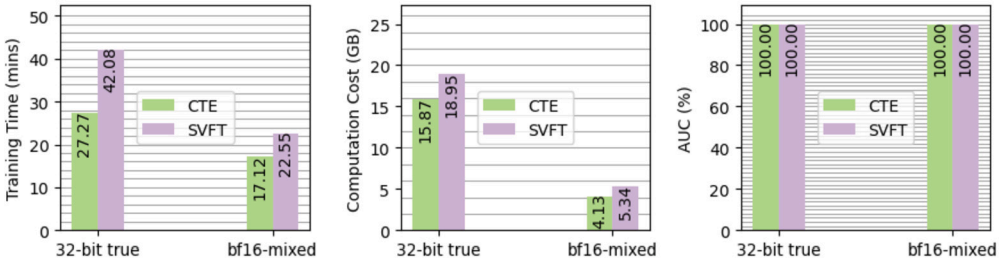


Fig. 16. Mixed-Precision Evaluation Comparison with respect to Computation Resources, Training Time and Accuracy.

5.5.4. Impact of the structure training strategy

It should be emphasized that the efficacy of the structure training approach is contingent upon several critical factors. The diversity and quality of training datasets are essential for the model to generalize well to unseen data. To assess the usefulness of the suggested training process, we conducted two experiments and calculated the equal error rate (EER). In the first case, we employed traditional randomization for training, and in the second case, we applied our proposed training structure. After 30 training epochs, in the first case, we got the EER of 0.1471, while on our proposed training methodology, we got less EER value of 0.0796 with the same AUC score of 99.99% on the intra-evaluation of the FF++ [50] dataset. These results suggest that a balanced representation of real and fake videos and a diverse range of characteristics and scenarios can help the model develop a robust understanding of the differences between the two types and have proven effective in our evaluations.

6. Conclusion

In conclusion, we introduce a novel approach for deepfake detection, SpectraVisionFusion Transformer (SVFT), which combines the analysis of frequency domain statistical features and spatial artifacts to improve the generalization capability. The coordinated analysis of visual and spectral information has proven effective in detecting deepfakes in diverse scenarios, as demonstrated by the improved accuracy, resilience to adversarial attacks, and better generalized performance. Further, to verify the contribution of our method, the empirical validation of our SVFT framework was conducted on various deepfake datasets. The SVFT achieved outstanding performances compared to state-of-the-art methods on intra- and cross-manipulation settings with improved average performance in terms of AUC in DF (9.03%), FS (2.23%), F2F (0.47%), and NT (8.4%). In robustness evaluation, although our method performed well with an increased compression ratio applied to deepfake images, the detection performance was affected

significantly. Further, when we applied the four unseen perturbations, our method performed well overall but saw a significant drop in performance when contrast or blur variations were applied, indicating our method's limitation. On the cross-dataset evaluation, our methods also performed consistently, such as on DFDC (83.50% vs. competition 75.56%), on the CDF (78.12% vs. competition 76.88%), but on the Deeper, our SVFT model underperformed (80.27% vs. competition 92.10%), but overall our method perform the best. We believe that our proposed approach contributes to advancing deepfake detection techniques, providing valuable insights for addressing the challenges posed by synthetic media and ensuring the integrity of visible content in the digital age. The extensive evaluation of benchmark datasets also demonstrated the efficacy of our proposed model in distinguishing between legitimate and deepfake videos, surpassing traditional approaches with their robust and enhanced detection capabilities, thereby indicating superior generalization performance. However, future work will further explore the impact analysis of compression, contrast, and blur on performance in our proposed framework and how it can be mitigated to improve performance in real-world cases. Besides that, the overall framework complexity and network compression will be studied.

CRediT authorship contribution statement

Muhammad Ahmad Amin: Conceptualization of this study, Methodology, Software, Data curation, Writing - Original draft preparation. **Yongjian Hu:** Supervision for Overseeing Experiments and Paper Drafting. **Chang-Tsun Li:** Collaborative Supervision for Experiments and Paper Drafting. **Beibei Liu:** Co-Supervision for Experimental Work and Paper Drafting.

Declaration of competing interest

Authors don't have any conflicts of interest.

Acknowledgements

The research work (Grant No. 2022GH15) has received funding from the Science and Technology Foundation of Guangzhou Huangpu Development District.

References

[1] S. Karnouskos, Artificial intelligence in digital media: the era of deepfakes, IEEE Trans. Technol. Soc. 1 (3) (2020) 138–147, https://doi.org/10.1109/TTS.2020.3001312.

[2] V.G. Ivanov, Y.R. Ignatovskiy, Deepfakes: prospects for political use and threats to the individual and national security, RUDN J. Public Adm. 7 (2020) 379–386, https://doi.org/10.22363/2312-8313-2020-7-4-379-386.

[3] Y. Pu, Z. Gan, R. Henao, X. Yuan, C. Li, A. Stevens, L. Carin, Variational autoencoder for deep learning of images, labels and captions, in: Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16, Curran Associates Inc., Red Hook, NY, USA, 2016, pp. 2360–2368, https://proceedings.neurips.cc/paper_files/paper/2016/file/eb86d510361fc23b59f18c1bc9802cc6-Paper.pdf.

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative Adversarial Networks, vol. 63, Association for Computing Machinery, New York, NY, USA, 2020, pp. 139–144.

[5] K. Schwarz, Y. Liao, A. Geiger, On the frequency bias of generative models, in: M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, J.W. Vaughan (Eds.), Advances in Neural Information Processing Systems, vol. 34, Curran Associates, Inc., 2021, pp. 18126–18136, https://proceedings.neurips.cc/paper_files/paper/2021/file/96bf57c6ff19504ff145e2a32991ea96-Paper.pdf.

[6] M. Khayatkhoei, A. Elgammal, Spatial Frequency Bias in Convolutional Generative Adversarial Networks, vol. 36, 2022, pp. 7152–7159.

[7] T. Dzanic, K. Shah, F. Witherden, Fourier spectrum discrepancies in deep network generated images, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, vol. 33, Curran Associates, Inc., 2020, pp. 3022–3032, https://proceedings.neurips.cc/paper_files/paper/2020/file/1f8d87e1161af68b81bace188a1ec624-Paper.pdf.

[8] M.A. Amin, Y. Hu, H. She, J. Li, Y. Guan, M.Z. Amin, Exposing deepfake frames through spectral analysis of color channels in frequency domain, in: 2023 11th International Workshop on Biometrics and Forensics (IWBF), 2023, pp. 1–6.

[9] Welcome - faceswap, https://faceswap.dev/.

[10] H. Averbuch-Elor, D. Cohen-Or, J. Kopf, M.F. Cohen, Bringing portraits to life, ACM Trans. Graph. 36 (6) (2017), https://doi.org/10.1145/3130800.3130818.

[11] S. Suwajanakorn, S.M. Seitz, I. Kemelmacher-Shlizerman, Synthesizing obama: learning lip sync from audio, ACM Trans. Graph. 36 (4) (2017), https://doi.org/10.1145/3072959.3073640.

[12] Y. Deng, J. Yang, S. Xu, D. Chen, Y. Jia, X. Tong, Accurate 3d face reconstruction with weakly-supervised learning: from single image to image set, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE Computer Society, Los Alamitos, CA, USA, 2019, pp. 285–295.

[13] F.-J. Chang, A.T. Tran, T. Hassner, I. Masi, R. Nevatia, G. Medioni, Deep, landmark-free fame: face alignment, modeling, and expression estimation, Int. J. Comput. Vis. 127 (6–7) (2019) 930–956, https://doi.org/10.1007/s11263-019-01151-x.

[14] Y. Nirkin, Y. Keller, T. Hassner, FSGANv2: improved subject agnostic face swapping and reenactment, IEEE Trans. Pattern Anal. Mach. Intell. 45 (01) (2023) 560–575, https://doi.org/10.1109/TPAMI.2022.3155571.

[15] L. Li, J. Bao, H. Yang, D. Chen, F. Wen, Advancing high fidelity identity swapping for forgery detection, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 5073–5082.

[16] K. Liu, I. Perov, D. Gao, N. Chervoniy, W. Zhou, W. Zhang, Deepfacelab: integrated, flexible and extensible face-swapping framework, Pattern Recognit. 141 (C) (2023), https://doi.org/10.1016/j.patcog.2023.109628.

[17] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, M. Nießner, Face2face, Commun. ACM 62 (1) (2018) 96–104, https://doi.org/10.1145/3292039.

[18] S.A. Samy, P. Anbalagan, Disturbance observer-based integral sliding-mode control design for leader-following consensus of multi-agent systems and its application to car-following model, Chaos Solitons Fractals 174 (2023) 113733, https://doi.org/10.1631/FITEE.2200181.

[19] S. Arockia Samy, R. Ramachandran, P. Anbalagan, Y. Cao, Synchronization of nonlinear multi-agent systems using a non-fragile sampled data control approach and its application to circuit systems, Front. Inf. Technol. Electr. Eng. 24 (4) (2023) 553–566, https://doi.org/10.1016/j.chaos.2023.113733.

[20] S. Samy, R. Raja, X. Bai, J. Alzabut, R. Swaminathan, G. Rajchakit, Asymptotic pinning synchronization of nonlinear multi-agent systems: its application to tunnel diode circuit, Nonlinear Anal. Hybrid Syst. 49 (2023) 1–17, https://doi.org/10.1016/j.nahs.2023.101366.

[21] D. Afchar, V. Nozick, J. Yamagishi, I. Echizen, MesoNet: a compact facial video forgery detection network, in: 2018 IEEE International Workshop on Information Forensics and Security (WIFS), 2018, pp. 1–7.

[22] F. Chollet, Xception: deep learning with depthwise separable convolutions, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1800–1807.

[23] D. Güera, E.J. Delp, Deepfake video detection using recurrent neural networks, in: 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2018, pp. 1–6.

[24] H.H. Nguyen, J. Yamagishi, I. Echizen, Capsule-forensics networks for deepfake detection, Adv. Comput. Vis. Pattern Recognit. (2022) 275–301, https://doi.org/10.1007/978-3-030-87664-7_13.

[25] L. Li, J. Bao, T. Zhang, H. Yang, D. Chen, F. Wen, B. Guo, Face x-ray for more general face forgery detection, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society, Los Alamitos, CA, USA, 2020, pp. 5000–5009.

[26] H. Liu, X. Li, W. Zhou, Y. Chen, Y. He, H. Xue, W. Zhang, N. Yu, Spatial-phase shallow learning: rethinking face forgery detection in frequency domain, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society, Los Alamitos, CA, USA, 2021, pp. 772–781.

[27] Y. Wang, K. Yu, C. Chen, X. Hu, S. Peng, Dynamic graph learning with content-guided spatial-frequency relation reasoning for deepfake detection, in: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society, Los Alamitos, CA, USA, 2023, pp. 7278–7287.

[28] J. Hu, X. Liao, W. Wang, Z. Qin, Detecting compressed deepfake videos in social networks using frame-temporality two-stream convolutional network, IEEE Trans. Circuits Syst. Video Technol. 32 (3) (2022) 1089–1102, https://doi.org/10.1109/TCSVT.2021.3074259.

[29] H. Zhao, T. Wei, W. Zhou, W. Zhang, D. Chen, N. Yu, Multi-attentional deepfake detection, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society, Los Alamitos, CA, USA, 2021, pp. 2185–2194.

[30] J. Cao, C. Ma, T. Yao, S. Chen, S. Ding, X. Yang, End-to-end reconstruction-classification learning for face forgery detection, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 4103–4112.

[31] T. Wang, H. Cheng, K.P. Chow, L. Nie, Deep convolutional pooling transformer for deepfake detection, ACM Trans. Multimed. Comput. Commun. Appl. 19 (6) (2023), https://doi.org/10.1145/3588574.

[32] S. Lin, C.-T. Li, A.C. Kot, Multi-domain adversarial feature generalization for person re-identification, IEEE Trans. Image Process. 30 (2021) 1596–1607, https://doi.org/10.1109/tip.2020.3046864.

[33] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, Improved techniques for training gans, in: Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16, Curran Associates Inc., Red Hook, NY, USA, 2016, pp. 2234–2242, https://proceedings.neurips.cc/paper_files/paper/2016/file/8a3363abe792db2d8761d6403605aeb7-Paper.pdf.

[34] T. Miyato, T. Kataoka, M. Koyama, Y. Yoshida, Spectral normalization for generative adversarial networks, in: International Conference on Learning Representations, 2018, https://openreview.net/forum?id=B1QRgziT-.
[35] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks, Proc. Mach. Learn. Res. 70 (2017) 214–223, https://proceedings.mlr.press/v70/arjovsky17a.html.
[36] T. Karras, T. Aila, S. Laine, J. Lehtinen, Progressive growing of GANs for improved quality, stability, and variation, in: 6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings, 2018, arXiv:1710.10196, https://iclr.cc/Conferences/2018.
[37] D. Bitouk, N. Kumar, S. Dhillon, P. Belhumeur, S.K. Nayar, Face swapping: automatically replacing faces in photographs, ACM Trans. Graph. 27 (3) (2008) 1–8, https://doi.org/10.1145/1360612.1360638.
[38] H.-X. Wang, C. Pan, H. Gong, H.-Y. Wu, Facial image composition based on active appearance model, in: 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, 2008, pp. 893–896.
[39] C. Cao, Y. Weng, S. Zhou, Y. Tong, K. Zhou, FaceWarehouse: a 3D facial expression database for visual computing, IEEE Trans. Vis. Comput. Graph. 20 (3) (2014) 413–425, https://doi.org/10.1109/TVCG.2013.249.
[40] Z. Geng, C. Cao, S. Tulyakov, 3d guided fine-grained face manipulation, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society, Los Alamitos, CA, USA, 2019, pp. 9813–9822.
[41] J.H. Hong, H. Kim, M. Kim, G.P. Nam, J. Cho, H.-S. Ko, I.-J. Kim, A 3d model-based approach for fitting masks to faces in the wild, in: 2021 IEEE International Conference on Image Processing (ICIP), 2021, pp. 235–239.
[42] Y. Nirkin, I. Masi, A.T. Tuăn, T. Hassner, G. Medioni, On face segmentation, face swapping, and face perception, in: Proceedings - 13th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2018, 2018, pp. 98–105, arXiv:1704.06729, https://doi.org/10.1109/FG.2018.00024.
[43] I. Masi, A.T. Tran, T. Hassner, J.T. Leksut, G. Medioni, Do we really need to collect millions of faces for effective face recognition?, in: Proceedings - Computer Vision – ECCV, 2016, pp. 579–596.
[44] I. Korshunova, W. Shi, J. Dambre, L. Theis, Fast face-swap using convolutional neural networks, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 3697–3705.
[45] J. Bao, D. Chen, F. Wen, H. Li, G. Hua, Towards Open-Set Identity Preserving Face Synthesis, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 6713–6722.
[46] R. Natsume, T. Yatagawa, S. Morishima Rsgan, Face swapping and editing using face and hair representation in latent spaces, in: ACM SIGGRAPH 2018 Posters, SIGGRAPH '18, Association for Computing Machinery, New York, NY, USA, 2018.
[47] L. Li, J. Bao, H. Yang, D. Chen, F. Wen, Advancing High Fidelity Identity Swapping for Forgery Detection, 2020, pp. 5073–5082.
[48] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, L. Zhang, Cvt: Introducing Convolutions to Vision Transformers, 2021, pp. 22–31.
[49] V. Sanh, L. Debut, J. Chaumond, T. Wolf, Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter, NeurIPS workshop, arXiv:1910.01108 [abs], http://arxiv.org/abs/1910.01108.
[50] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, M. Niessner, FaceForensics++: learning to detect manipulated facial images, in: Proceedings of the IEEE International Conference on Computer Vision 2019-October, 2019, pp. 1–11, arXiv:1901.08971, https://doi.org/10.1109/ICCV.2019.00009.
[51] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. 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, ICLR, 2021, https://openreview.net/forum?id=YicbFdNTTy.
[52] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.
[53] J. Ba, J.R. Kiros, G.E. Hinton, Layer normalization, arXiv:1607.06450 [abs], https://api.semanticscholar.org/CorpusID:8236317.
[54] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1 (Long and Short Papers), Association for Computational Linguistics, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, 2019, pp. 4171–4186.
[55] W. Zhang, Y. Feng, F. Meng, D. You, Q. Liu, Bridging the gap between training and inference for neural machine translation, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 4334–4343.
[56] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, Curran Associates Inc., Red Hook, NY, USA, 2017, pp. 6000–6010, https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.
[57] D. Hendrycks, K. Gimpel, Gaussian error linear units (gelus), ICLR. URL, https://api.semanticscholar.org/CorpusID:125617073.
[58] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, C.C. Ferrer, The DeepFake Detection Challenge (DFDC) Dataset, arXiv:2006.07397v4, http://arxiv.org/abs/2006.07397.
[59] Y. Li, X. Yang, P. Sun, H. Qi, S. Lyu, Celeb-DF: a large-scale challenging dataset for DeepFake forensics, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2020, pp. 3204–3213, arXiv:1909.12962, https://doi.org/10.1109/CVPR42600.2020.00327.
[60] L. Jiang, R. Li, W. Wu, C. Qian, C.C. Loy, Deeperforensics-1.0: a large-scale dataset for real-world face forgery detection, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 2886–2895.
[61] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: a large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.
[62] P. Micikevicius, S. Narang, J. Alben, G.F. Diamos, E. Elsen, D. Garcia, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh, H. Wu, Mixed precision training, in: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, ICLR, 2018, https://openreview.net/forum?id=r1gs9JgRZ.
[63] J. Deng, J. Guo, E. Ververas, I. Kotsia, S. Zafeiriou, RetinaFace: single-shot multi-level face localisation in the wild, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 5202–5211.
[64] Z. Zhang, M.R. Sabuncu, Generalized cross entropy loss for training deep neural networks with noisy labels, in: Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS'18, Curran Associates Inc., Red Hook, NY, USA, 2018, pp. 8792–8802, https://proceedings.neurips.cc/paper_files/paper/2018/file/f2925f97bc13ad2852a7a551802feea0-Paper.pdf.
[65] L. van der Maaten, G. Hinton, Visualizing data using t-sne, J. Mach. Learn. Res. 9 (86) (2008) 2579–2605, http://jmlr.org/papers/v9/vandermaaten08a.html.