MINTIME: Multi-Identity Size-Invariant Video Deepfake Detection

Davide Alessandro Coccomini, Giorgos Kordopatis Zilos, Giuseppe Amato, Roberto Caldelli, *Senior Member, IEEE*, Fabrizio Falchi, Symeon Papadopoulos, and Claudio Gennaro

***Abstract*— In this paper, we present MINTIME, a video deepfake detection method that effectively captures spatial and temporal inconsistencies in videos that depict multiple individuals and varying face sizes. Unlike previous approaches that either employ simplistic a-posteriori aggregation schemes, i.e., averaging or max operations, or only focus on the largest face in the video, our proposed method learns to accurately detect spatio-temporal inconsistencies across multiple identities in a video through a Spatio-Temporal Transformer combined with a Convolutional Neural Network backbone. This is achieved through an Identity-aware Attention mechanism that applies a masking operation on the face sequence to process each identity independently, which enables effective video-level aggregation. Furthermore, our system incorporates two novel embedding schemes: (i) the Temporal Coherent Positional Embedding, which encodes the temporal information of the face sequences of each identity, and (ii) the Size Embedding, which captures the relative sizes of the faces to the video frames. MINTIME achieves state-of-the-art performance on the ForgeryNet dataset, with a remarkable improvement of up to 14% AUC in videos containing multiple people. Moreover, it demonstrates very robust generalization capabilities in cross-forgery and cross-dataset settings. The code is publicly available at: https://github.com/davide-coccomini/MINTIME-Multi-Identity-size-iNvariant-TIMEsformer-for-Video-Deepfake-Detection.**

***Index Terms*— Deepfake detection, computer vision, deep learning, vision transformers, convolutional neural networks.**

Manuscript received 3 May 2023; revised 28 November 2023 and 25 March 2024; accepted 21 May 2024. Date of publication 3 June 2024; date of current version 11 June 2024. This work was supported in part by the NextGeneration EU (EU-NGEU) through the Project SERICS under the National Recovery and Resilience Plan (NRRP) Ministry of University and Research [Ministero dell'Università e della Ricerca (MUR)] Program under Grant PE00000014, in part by the EU–NGEU through Empowering Knowledge Extraction to Empower Learners (EKEEL) under Grant P20227PEPK and Grant ERC PE6_7, in part by the H2020 Project AI4Media under Grant 951911, in part by the Italian MUR (Research Projects of National Relevance [Progetti di Rilevante Interesse Nazionale (PRIN)] 2022) through the FOSTERER Project, and in part by the Junior Star Czech Science Foundation (GACR) under Grant GM 21-28830M. The associate editor coordinating the review of this article and approving it for publication was Dr. Daniel Moreira. *(Corresponding author: Davide Alessandro Coccomini.)*

Davide Alessandro Coccomini, Giuseppe Amato, Fabrizio Falchi, and Claudio Gennaro are with ISTI-CNR, 56124 Pisa, Italy (e-mail: davidealessandro.coccomini@isti.cnr.it; giuseppe.amato@isti.cnr.it; fabrizio.falchi@isti.cnr.it; claudio.gennaro@isti.cnr.it).

Giorgos Kordopatis Zilos is with the Faculty of Electrical Engineering, Czech Technical University in Prague, 160 00 Prague, Czech Republic (e-mail: kordogeo@fel.cvut.cz).

Roberto Caldelli is with CNIT, 50121 Florence, Italy, and also with the Faculty of Technological and Innovation Sciences, Universitas Mercatorum, 00186 Rome, Italy (e-mail: roberto.caldelli@unifi.it).

Symeon Papadopoulos is with ITI-CERTH, 570 01 Thessaloniki, Greece (e-mail: papadop@iti.gr).

This article has supplementary downloadable material available at https://doi.org/10.1109/TIFS.2024.3409054, provided by the authors.

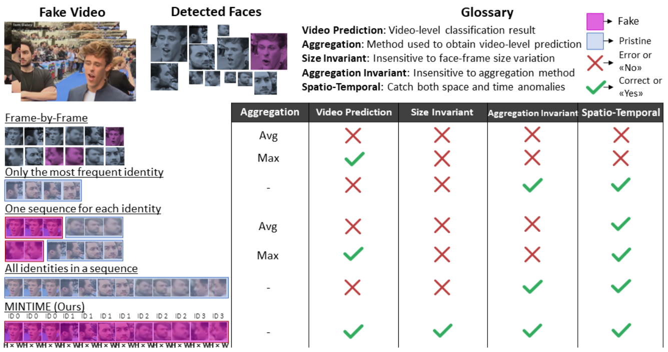
Digital Object Identifier 10.1109/TIFS.2024.3409054

I. Introduction

OVER the last few years, we are witnessing an increasing spread of deepfake images and videos that are becoming more credible and realistic due to the continuous advancement of generative models, such as Generative Adversarial Networks (GANs) [1], Neural Radiance Fields (NeRFs) [2], [3] and Diffusion Models [4]. Such models can generate very high-quality and photorealistic deepfake images and videos, making the verification of such media increasingly difficult. This ultimately makes distinguishing reality from fiction a daunting task at best. To this end, exploiting manipulated images and videos of people has generated several defamation campaigns against public figures, celebrities, or even regular citizens whose lives can be undermined by manipulated content. Moreover, deepfakes may be used for a variety of serious crimes such as extortion, intimidation and damages to democracy [5].

In response to this sudden and uncontrollable development, many efforts have been made to counteract deepfakes by implementing a multitude of deepfake detection systems based on a variety of approaches. However, distinguishing manipulated from pristine content introduces many challenges, and there are still many open issues. In this work, we focus on those issues that we consider crucial for detecting deepfakes 'in the wild' and are starting to attract the research community's attention. Previous studies [6], [7], [8] have pointed out that identifying temporal inconsistencies and anomalous variations between two frames of the same video is fundamental for successful deepfake detection. Much work, however, tends to focus on spatial inconsistencies only, handling videos with a frame-by-frame classification approach and relying on naive aggregation schemes to extract video-level predictions [9], [10], [11], [12]. When videos are analyzed frame-by-frame or divided into separately classified sequences, one needs to face the problem of aggregating frame- or sequence-level predictions into a video-level prediction. We can divide approaches into *a-posteriori aggregation*, i.e., those that use static functions such as average or maximum to obtain the final result, and *internal aggregation*, i.e., those that let the model analyze the entire video to determine the final video-level prediction. Methods based on the first approach are very sensitive to the choice of aggregation function as shown in [8] and [13].

An additional problem resides in a vulnerability that attackers can exploit to deceive a deepfake detector. In cases of videos with multiple distinct people (identities) appearing together [14], an attacker could decide to manipulate only one


Fake Video
Detected Faces
Glossary
Video Prediction: Video-level classification result
Aggregation: Method used to obtain video-level prediction
Size Invariant: Insensitive to face-frame size variation
Aggregation Invariant: Insensitive to aggregation method
Spatio-Temporal: Catch both space and time anomalies
Fake
Pristine
Error or «No»
Correct or «Yes»
ID 0 ID 0 ID 0 ID 1 ID 1 ID 1 ID 1 ID 2 ID 2 ID 2 ID 3 ID 3
H × WH × WH × WH × WH × WH × WH × WH × WH × WH × WH × WH × W

	Aggregation	Video Prediction	Size Invariant	Aggregation Invariant	Spatio-Temporal
Frame-by-Frame	Avg	✗	✗	✗	✗
	Max	✓	✗	✗	✗
Only the most frequent identity	-	✗	✗	✓	✓
One sequence for each identity	Avg	✗	✗	✗	✓
	Max	✓	✗	✗	✓
All identities in a sequence	-	✗	✗	✓	✓
MINTIME (Ours)	-	✓	✓	✓	✓

Fig. 1. The impact of the deepfake detection strategy in a case of video containing multiple identities and variations in the face-frame area ratio. The proposed approach is the only one capable of identifying spatio-temporal inconsistencies and at the same time effectively handling cases of multiple identities, variations in face size and without the use of aggregations that would impact the result.

of them. However, if the detection is carried out *en bloc* for all the detected faces, the negative contribution to the final prediction made by the fake faces could be 'overshadowed' by the non-manipulated ones, thus deceiving the system. One possible approach would be to split the video into several clips, at least one for each identity, but this would require multiple forward passes of the model generating multiple predictions – hence, leading once more to the aggregation problem. These approaches usually rely on a-posteriori aggregation policies to obtain an overall video-level prediction, which significantly affects the final result. To the best of our knowledge, no method in the literature can handle any number of identities based on an internal aggregation scheme. In general, prior works in the literature have put minimal effort into this specific case of multiple-identity videos, which, however, is common in real-world verification problems.

Typically in deepfake detection systems, faces are detected from the input video or image, and before being input to a classification model, they are resized to a uniform size. This may result in vital loss of information, for instance the size of the subject's face with respect to the rest of the scene. Such information could be exploited to build models robust to environmental clutter, e.g., a sudden change of distance from the camera or low-resolution background faces from an original video/image, which could better distinguish traces introduced by the deepfake generation process.

Finally, the deepfake generation model also tends to introduce specific patterns within images and videos. Hence, deepfake detectors often end up learning to recognize only generative models included in the training dataset and are therefore ineffective when dealing with unseen manipulations, demonstrating poor generalization [15], [16], [17], [18], [19], [20], [21], [22]. Furthermore, many approaches may be ineffective in real-world cases because they are trained and validated in very constrained situations where, for example, there is only a single subject in the video or people tend to always stay at the same distance from the camera [23].

To overcome the above challenges, we present the Multi-Identity size-iNvariant TIMEsformer (MINTIME) for video deepfake detection. The main contributions presented by our approach, which are also illustrated in Figure 1, are:

- Ability to identify inconsistencies in both space and time through the combination of a Spatio-Temporal Transformer and a Convolutional Neural Network (CNN) – unlike other hybrid models designed for deepfake detection [6], [11], [12] that decouples space and time working exclusively in one dimension.
- Ability to handle multiple people in the same video through an Identity-aware attention mechanism, capable of keeping track of the identity to which each face detected in the video refers, combined with a positional embedding technique, namely Temporal Coherent Positional Embedding, which can maintain both spatial and temporal coherence.
- Ability to handle variations in the face-frame area ratio through the introduction of size embeddings that keep track of the ratio between the detected face area and the entire frame at each instant of time.
- Being unaffected by aggregation strategies thanks to an internal aggregation obtained by analyzing the entire video in a single sequence, letting the network infer the video-level prediction by handling appropriately multi-identity videos in a single forward pass. This way, the model directly returns a single prediction for the entire video without requiring additional post-processing.

The performance of the proposed system has been evaluated in a multitude of different contexts, with an improvement of up to 14% AUC in videos containing multiple identities. It has also been validated in cross-forgery and cross-dataset scenarios, outperforming state-of-the-art in all contexts, with improvements in the AUC score of up to 22% in some cases, demonstrating a high level of generalization.

II. Related Work

The growing interest in deepfakes has fuelled the emergence of numerous solutions to detect them using a variety of approaches [24]. Generally speaking, what practically all these methods seek to achieve is to correctly classify a video as pristine or deepfake by learning to distinguish patterns that are often introduced during the deepfake generation process. As we focus on video deepfake detection, we summarize previous works into two main categories based on the type of approaches they adopt.

Space-Only: These methods often treat the video as a series of frames by performing a separate classification per frame and then aggregating them into a final overall classification using an a-posteriori aggregation scheme such as the average or maximum function. These approaches focus primarily on identifying specific traces introduced by deepfake generation methods, namely those of a spatial nature. Even though they are most suited for deepfake image detection, where there is no need for detecting temporal inconsistencies, they are often applied to videos in a frame-by-frame manner. Most are based on well-established deep learning approaches, such as CNNs [9], [10], [25], [26]. For example, the authors in [27] proposed the Central Difference Convolution (CDC) that utilizes both pixel intensity and pixel gradient data to provide a fixed representation of variations in texture and then detect deepfakes. Also, recently some attention-based approaches have been proposed [15], [28], [29], [30]. For example, the authors in [30] proposed an attention-based method capable of analyzing an image in low and high frequency to detect irregular patterns and artifacts. Some previous works also proposed hybrid architectures that combine Vision Transformers with various types of CNNs as in [12], where the features extraction capabilities of EfficientNet [31] are combined with the Cross Vision Transformer [32]. Aggregation, in this case, is done by making the maximum among the predictions obtained on the individual frames. A similar approach but using an XceptionNet [33] is proposed in [11]. One of the best generalization capability was demonstrated by [34], where the authors propose using high-frequency noise to expose discrepancies between authentic and tampered regions, employing three functional modules for effective feature extraction and learning. However, all these methods cannot capture temporal inconsistencies that can be crucial in deepfake detection.

Spatio-Temporal: These approaches aim to capture temporal inconsistencies and obtain a video-level classification without any kind of additional aggregation. Some of them focus on specific types of spatio-temporal artifacts common in the case of deepfake videos, such as anomalous lip movement [18] or inconsistent eye-blinking [35], but they are limited since they do not look for additional artifacts that may be present in other parts of the face. Moreover, others exploit the optical flow of video [36] or analyze the relationships between audio peaks and video content [37]. A recent approach [7] is composed of two stages: firstly, it uses self-supervised learning to capture temporal information from real videos, such as facial movements and expressions, and secondly, it uses these learned representations to train a forgery detector to make real/fake decisions based on these factors. The authors of [38] also proposed a method to detect inter-frame patterns via LSTM networks [39] looking for temporal inconsistencies. Finally, one of the most relevant methods focusing on temporal incoherence within videos is FTCN [6], which uses a Temporal Transformer Encoder at its core. Although these methods are designed to capture spatio-temporal inconsistencies, they do not consider several important nuances of the problem. For example, they construct the input sequence by selecting a single person from the video, even when there is more than one, and do not consider the different face-frame area ratios that may occur or vary in them. An attempt to handle cases of multiple people in the video comes from [12] in which identities are classified separately, and if even one of them is detected as manipulated, then the whole video is considered fake. However, this method is frame-based and relies on an ad hoc aggregation scheme (average or max), thus, requiring a forward pass for each frame. Incorrect handling of these situations can lead to completely ignoring a manipulated subject in favour of a perfectly pristine one, which leads to wrong classification of the overall video. An aspect partially exploited in deepfake detection is the identity of subjects appearing in videos. Some works attempt to detect deepfakes by identifying a person from temporal facial features, specific to how a person moves while talking [40], or by using biometric analysis techniques [41]. However, these latter approaches remain particularly effective mainly in the case of a major manipulation of the person's identity and have limited capability in detecting videos manipulated by other types of deepfake generation methods. They are also often tied to an identity and are ill-suited in cases where the subject under consideration is not a famous person, and there are few videos depicting him/her. In terms of generalization, an exception is made in the case of [42] where the authors proposed an Identity Consistency Transformer capable of detecting the inconsistencies in the identity with good performances in several scenarios.

III. Background

Vision Transformer (ViT) [43] ViT is a type of deep neural network architecture designed for image classification tasks, which relies on the *transformer architecture* [44], originally developed for natural language processing tasks. The core idea is to treat an image as a sequence of patches that are first projected to an embedding space and then fed to a transformer model. The building block of a transformers network is the *attention mechanism* that enables the model to capture dependencies between patch embeddings and learn contextual relationships, enhancing its ability to recognize complex patterns in images that can be exploited for the final estimation. Additionally, the *CLS token*, short for "classification", is a learnable vector that is a model variable. It is processed along with the patch embedding and enriched with relevant information through the attention process. It finally serves as the global representation of the entire patch sequence and is typically used for the final classification. Furthermore, to encode spatial information and capture the sequential arrangement of image patches, *positional embeddings* are employed in a ViT. These are learnable vectors that encode the portions of each patch in the image that are added to the patch embeddings. This enables the model to discern the spatial context within the overall image.

Attention Mechanism: The attention mechanism enables models to exploit similarity between items of input item sequences, so as to capture relevant information. The attention mechanism works by dynamically computing weights based on the similarity of items of two sequences that are then used to compute a weighted sum of a third set, which is the output. More precisely, let three vector sequences, i.e., queries $\mathbf{Q} \in \mathbb{R}^{N \times D}$, keys $\mathbf{K} \in \mathbb{R}^{M \times D}$, and values $\mathbf{V} \in \mathbb{R}^{M \times D}$, where N, M are the corresponding sequence lengths, and D the vector dimensionality. The attention mechanism is applied as

$$\alpha(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \sigma\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{D}}\right)\mathbf{V}, \tag{1}$$

where σ is the softmax function, and $\sqrt{D}$ is a scaling factor. In this formulation, the matrix product between the queries and keys can be understood as the pairwise similarity between their composed vectors. The softmax function normalizes the scores across all keys for each query, so that they add up to one. Finally, the weighted sum of the values is computed using the normalized scores.

ViT Block: This is the main component of a ViT model. It consists of a self-attention layer, i.e., a layer built based

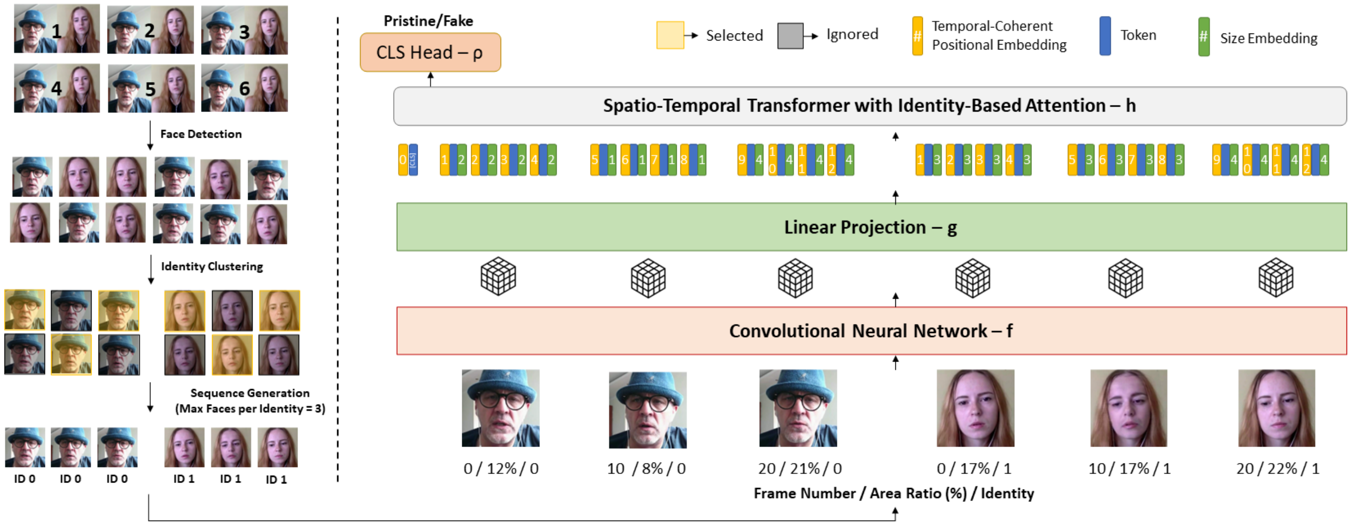


Fig. 2. MINTIME overview: The pre-processing pipeline (left) starts with the detection of faces in the video, follows with the clustering of identities and then creates the input sequence. The sequence of faces is converted into features by the convolutional backbone (right), which once converted into tokens (four tokens for each identity for visualization purposes) and concatenated to the embeddings, pass into the TimeSformer and finally into the classification head for the final classification.

on the attention mechanism, and a feed-forward network, i.e., a two-layer Multi-Layer Perceptron (MLP). Typically, several blocks are stacked to build a ViT. More formally, let $\mathbf{z}_i \in \mathbb{R}^{HW \times D}$ be the patch embeddings of an image with H and W the spatial dimensions and D the embedding dimensionality. The processing takes place as follows

$$\begin{aligned}\hat{\mathbf{z}} &= \mathbf{z} + \alpha(\mathbf{z}W^Q, \mathbf{z}W^K, \mathbf{z}W^V)\\ h(\mathbf{z}) &= \hat{\mathbf{z}} + \phi(\hat{\mathbf{z}}W^1)W^2,\end{aligned} \tag{2}$$

where $h(\cdot)$ denotes the ViT block as a function given an input tensor, $W^Q, W^K, W^V \in \mathbb{R}^{D\times D}$ are projection matrices of learnable weights, $W^1 \in \mathbb{R}^{D\times D'}$ and $W^2 \in \mathbb{R}^{D'\times D}$ are the learnable weights of the feed-forward network with D' latent dimension and $\phi(\cdot)$ a non-linear activation function, i.e., GeGLU [45]. During backpropagation, all learnable weights are updated. In our implementation, the multi-head variant of the attention layer is used [44], and Layer Normalization (LN) [46] is applied before the residual connections. They are not displayed in (2) for simplicity in presentation. The output of one block is given as an input to the next block.

IV. Proposed Approach

The proposed Multi-Identity size-iNvariant TIMEsformer (MINTIME) approach is illustrated in Figure 2. It receives as input a video containing one or more identities and detects whether the video has been manipulated. Our method first processes the input video to extract sequences of face images in an adaptable manner. Then, the extracted sequences are propagated through a network architecture, consisting of a CNN backbone network and a Spatio-Temporal Transformer, that are able to effectively capture the spatio-temporal inconsistencies within its content. MINTIME is versatile and able to efficiently adapt to a multitude of real world complex settings. In the following, we present in detail the proposed pipeline and the novelties of our network architecture.

A. Preprocessing and Feature Extraction

1) Pre-Processing: Given an input video v, we need to detect the depicted faces and extract various details necessary for our system. To this end, a frame-by-frame face detection [47] is initially applied that detects the location within video frames where faces are displayed. For each of the detected faces, we derive the following information: (i) the face image tensor $\mathbf{x}_i \in \mathbb{R}^{3\times H\times W}$, where 3 is the number of the RGB channels and H and W are the height and width of the cropped face image respectively, (ii) the timestamps $t_i \in \mathbb{N}$, i.e., the index of the frame where they are extracted, and (iii) the face-frame area ratio $s_i \in (0., 1.]$, i.e., the ratio between the area of the face image and the frame. Additionally, we apply a face clustering scheme [13] in order to group the detected faces and link them with an identity. In that way, each detected face corresponds to an identity index $I_i \in \mathbb{N}$, which is the cluster id where it belongs. In the end, each arbitrary face i corresponds to a tuple $f_i = (\mathbf{x}_i, I_i, s_i, t_i)$ containing all the necessary information for later processing.

To enable the model to handle multiple identities in one video, we need to generate an input face sequence composed of faces from several identities. Since the different videos have a different number of faces, we need to select a fixed number of faces to generate sequences with fixed slots so as to facilitate vectorization and batch processing. To this end, the identities are ordered according to the face-frame area ratio of their faces (higher ratios first). Top-ranked identities are generally prioritized by allocating a higher number of slots to them in the sequence. This is done to drive our model to focus more on faces that cover a larger area in the video frames and, therefore, likely be more relevant compared with the smaller ones. However, as evidenced in Table I, the system takes into account all identities with the first two that are equally treated. In our implementation we only consider a maximum of 4 identities but in the case of a higher number of identities the slot distribution can be customized as desired. This is just

TABLE I

NUMBER OF SLOTS ($N = 16$ IN TOTAL) ASSIGNED TO EACH IDENTITY BASED ON THE NUMBER OF CONSIDERED IDENTITIES, TO COMPOSE THE INPUT SEQUENCE

Considered Identities	Identity			
	1	2	3	4
1	16	N/A	N/A	N/A
2	8	8	N/A	N/A
3	6	6	4	N/A
4	6	6	2	2

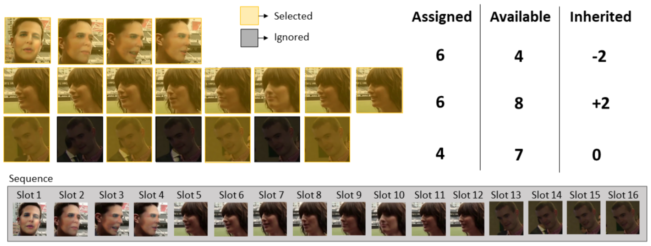


Fig. 3. Sequence sampling when the number of assigned faces in the sequence is inherited through identities. The numbers on the right indicate, for each identity, how many faces can be used (the assignment is based on hyperparameters), how many are available and how many have been inherited from or left to another identity. A negative value in the "Inherited" column means that some of the available slots have not been used; a positive one means that some slots have been inherited from the previous identity.

an implementation choice but the system is anyway designed to deal with any slot distribution.

The sequence length N, i.e., the available slots (16 is usually chosen), and the number of identities K used for the sequence generation is fixed, and we treat them as hyperparameters of our system. The number of slots assigned to each identity in the case of multi-identity videos is also a hyperparameter and can be customized based on the number of identities available in the considered video. If there is only one identity in the video, its faces fill the entire sequence; otherwise, the sequence is filled by distributing the slots to the various identities prioritizing larger ones, as mentioned above. In cases where there are not enough faces to fill the input sequence for one identity, then faces from other identities are used, or the sequence is padded with empty images. On the contrary, in the case of longer identity sequences, uniform sampling is performed, taking into account some faces spread as much uniformly as possible in time. Figure 3 illustrates an example of the sequence generation process. The output of this process is a set of tuples $\mathcal{F} = \{(\mathbf{x}_1, I_1, s_1, t_1), (\mathbf{x}_2, I_2, s_2, t_2), \ldots, (\mathbf{x}_N, I_N, s_N, t_N)\}$ consisting of all tuples of individual frames considered in the sequence.

2) Feature Extraction: To extract features from the detected face images, we use a CNN backbone that maps the image tensors $\mathbf{x}_i$ of the faces in the input sequence into the feature space. Then, the output of the CNN backbone is propagated to the Spatio-Temporal Transformer for further processing. The CNN backbone can be denoted as a function $f : \mathbb{R}^{3\times H\times W} \rightarrow \mathbb{R}^{C\times H'\times W'}$, where C are the channel dimensions of the feature space, i.e., the dimensionality of the output feature maps. Hence, the resulting features maps are $f(\mathbf{x}_i) \in \mathbb{R}^{C\times H'\times W'}$, where H' and W' are its spatial dimensions resulting from the mapping and depend on the employed backbone network. In the rest of the paper, we omit the accents in the notation of H' and W' for a more compact presentation.

With the use of the CNN backbone, we generate comprehensive representations that capture the low-level spatial information of face images and facilitate the analysis of face sequences by the Spatio-Temporal Transformer. In our experiments, we benchmark the performance of two CNN backbones, i.e., an EfficientNet-B0 [31], which has been combined with Transformer networks in prior work [12], and the popular XceptionNet [33] used in several approaches [11].

B. Spatio-Temporal Transformer

To implement our Spatio-Temporal Transformer, we used a variation of the classical ViT, namely the TimeSformer [48], proposed for video analysis and designed to capture not only spatial information but also the temporal evolution of the scene. We extend it with several adaptations for the proper analysis of our input face sequences. More precisely, we build our model using the divided space-time attention, which incorporates separate spatial and temporal attention mechanisms applied to the input in turns. It splits frames into non-overlapping patch tokens and first calculates the attention between the corresponding patches in the previous and next frames to capture temporal dynamics, and then calculates the attention between the patches of the same frame to capture spatial information. To the best of our knowledge, this is the first time that this architecture has been employed for DeepFake Detection.

1) Face Tokens: Having extracted the feature maps for all faces in a sequence, we then generate face tokens for our Spatio-Temporal Transformer – unlike the original TimeSformer [48] where tokens are generated from frame patches. In particular, to generate our face tokens, we concatenate, reshape and project with a linear layer all the feature maps extracted from the faces of our sequence through the feature extraction step. This can be considered as the function $g : \mathbb{R}^{N\times C\times H\times W} \rightarrow \mathbb{R}^{NHW\times D}$, where D is the dimensionality of output face tokens. Hence, the final face tokens are denoted as $\mathbf{z}_f \in \mathbb{R}^{NHW\times D}$.

Furthermore, we employ a CLS token $\mathbf{z}_0 \in \mathbb{R}^{1\times D}$, which is processed by our network along with the other face tokens. This captures all relevant information from the face tokens and is used to derive the final video-level prediction for an input video. In that way, we enable our system to perform internal aggregation for the final prediction, without relying on an a-posteriori aggregation scheme. Our final face tokens are

$$\mathbf{z} = \begin{bmatrix} \mathbf{z}_0 \\ \mathbf{z}_f \end{bmatrix} \in \mathbb{R}^{(NHW+1)\times D} \tag{3}$$

where $[\cdot]$ indicates concatenation.

Additionally, we enhance our face tokens with the two embedding schemes that capture temporal order and relative face-frame size, as depicted in Figure 4. These embeddings, explained in more detail in the next lines, are based on both the spatial position of the patches extracted from each face and on the temporal position of the face in the video.

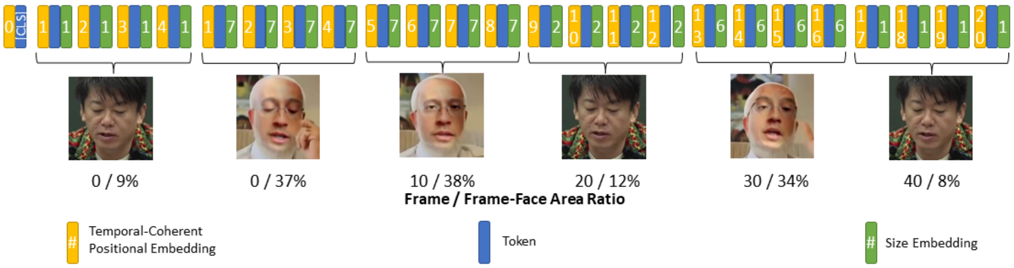


Fig. 4. Overview of Temporal Coherent Positional Embedding and Size Embedding on a two-identities video frame. The identities' faces are detected at different frames and this is translated into TCPEs equally between the ones appearing together in the scene and different TCPEs when the faces appear in two distinct frames. For example the first two faces both appear in frame 0 so they have the same TCPE but different SE because they have diverse face-frame area ratios. The third and the fourth face, on the other hand, are detected in different frames (10 and 20 respectively) and so they have different TCPEs. The same situation for the last two faces, detected in frames 30 and 40 respectively. Also, each face has a diverse face/frame area ratio, conducting different size embeddings for each face.

Temporal Coherent Positional Embedding (TCPE): To encode temporal information into our tokens, we adopt the positional embedding scheme used in the ViT [44]. For their application, we order faces based on their timestamps t_i in chronological order. Since two faces can derive from the same video frame, their temporal encoding should be identical and not relative to their position in the face sequence. Therefore, in that case, we use the same positional embedding, which corresponds to the minimum index in the face sequence. In that way, we maintain local and global temporal coherence for the frames of each identity and across the frames of all identities in the video, respectively. We use trainable positional embeddings that are added to our tokens.

Size Embedding (SE): To inject size information into our tokens, we build a similar approach as the positional embeddings, where, instead of positions, we exploit the face-frame area ratio to retrieve the embedding tokens. Since the face-frame area ratio s_i is a percentage in the (0%, 100%] range, we split this range into 20 intervals of 5% each. Then, for each interval, a size embedding vector is determined, which is used for all faces with a ratio falling within the corresponding interval. For example, if a face has an area covering the 16% of the entire frame, the size embedding of the fourth interval, i.e., (15%, 20%], will be used. In that way, we encode the relative face size to our face tokens, enabling our network to learn to leverage such information. Similar to positional embeddings, we use trainable size embeddings that are added to our tokens.

2) Identity-Aware Spatio-Temporal Attention: In our approach, we want to capture spatio-temporal relations in the face tokens of the input sequences. Hence, following divided space-time attention from [48], we employ the attention mechanism (1) in two ways so as to capture: (i) the spatial relations between tokens of the same face, i.e., spatial attention, and (ii) the temporal patterns analysing face tokens derived from a particular image location in time, i.e., temporal attention.

For spatial attention, our goal is to process each frame independently. Let $\mathbf{z} \in \mathbb{R}^{(NHW+1)\times D}$ represent a token sequence comprising face tokens, denoted as $\mathbf{z}_i \subset \mathbf{z}$, where $\mathbf{z}_i \in \mathbb{R}^{1\times D}$. For each token $\mathbf{z}_i$, we define a set of related face tokens $\mathbf{z}_i^s \in \mathbb{R}^{HW\times D}$. This set consists of all the face tokens extracted from the same face image, characterized by a *specific identity* and a *specific timestamp* according to $\mathbf{z}_i$. Note that $\mathbf{z}_i^s$ does not contain tokens extracted from other face images, even if they originate from the same identity but in different timestamps. The attention mechanism is applied between the CLS token $\mathbf{z}_0$ and all tokens in our sequence $\mathbf{z}$, including itself. For all other tokens, the attention mechanism is applied only with the CLS token $\mathbf{z}_0$ and the corresponding ones in the $\mathbf{z}_i^s$. More formally, the spatial attention is given by

$$s\alpha(\mathbf{z}_i) = \begin{cases} \alpha(\mathbf{z}_0 W_s^Q, \mathbf{z}W_s^K, \mathbf{z}W_s^V) & i = 0 \\ \alpha(\mathbf{z}_i W_s^Q, \begin{bmatrix} \mathbf{z}_0 \\ \mathbf{z}_i^s \end{bmatrix} W_s^K, \begin{bmatrix} \mathbf{z}_0 \\ \mathbf{z}_i^s \end{bmatrix} W_s^V) & i \neq 0 \end{cases}, \quad (4)$$

where $s\alpha(\cdot)$ denotes the spatial attention for an input token, and $W_s^Q, W_s^K, W_s^V \in \mathbb{R}^{D\times D}$ are learnable weights. This process is similar to the space attention used in TimeSformer [48].

Similarly, for temporal attention, we aim at capturing the temporal inconsistencies for each region in detected faces. Since we target videos with multiple identities, we need to process the face images from each identity separately, without mixing information from different identities. Hence, for each face token $\mathbf{z}_i \in \mathbb{R}^{1\times D}$, we consider the concatenated set $\mathbf{z}_i^I \in \mathbb{R}^{N^I\times D}$ of all tokens of the corresponding image regions where the same identity is displayed, where N^I is the total number of faces of the identity. Again, for the CLS token, the attention scores are computed considering all face tokens. All other tokens consider only the CLS and the corresponding ones in the $\mathbf{z}_i^I$. To this end, the attention output is given by:

$$id - t\alpha(\mathbf{z}_i) = \begin{cases} \alpha(\mathbf{z}_0 W_t^Q, \mathbf{z}W_t^K, \mathbf{z}W_t^V) & i = 0 \\ \alpha(\mathbf{z}_i W_t^Q, \begin{bmatrix} \mathbf{z}_0 \\ \mathbf{z}_i^I \end{bmatrix} W_t^K, \begin{bmatrix} \mathbf{z}_0 \\ \mathbf{z}_i^I \end{bmatrix} W_t^V) & i \neq 0 \end{cases} \quad (5)$$

where $id - t\alpha(\cdot)$ denotes the identity-aware temporal attention for an input token, and $W_t^Q, W_t^K, W_t^V \in \mathbb{R}^{D\times D}$ are learnable weights. Figure 5 illustrates how attention is calculated exclusively by tokens referring to identity 0 faces (green), ignoring those referring to identity 1 faces (red) and vice versa, while all attend to the CLS token. The more intense the colour, the higher the attention score for the pair of tokens. The grey squares are token pairs for which the attention scores are not

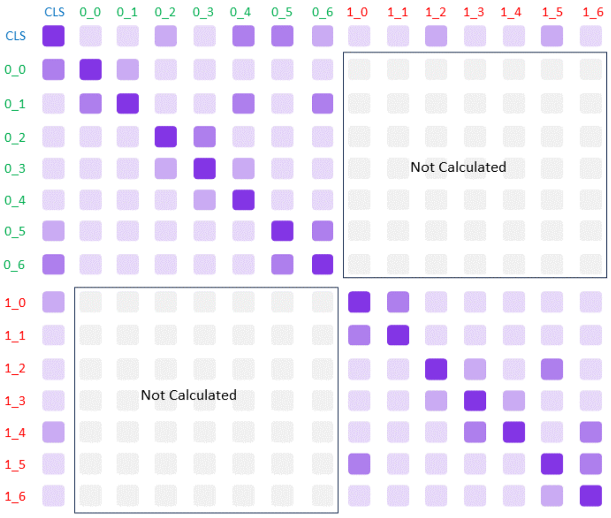


Fig. 5. Identity-aware Attention on tokens for two-identities video. The naming X_Y stands for the identity ID and token ID respectively. The more intense the colour, the higher the attention value for each couple of tokens. The attention is not computed between couples of tokens referring to different identities.

calculated. This is because they are pairs of tokens derived from two distinct identities; hence, the identity-aware attention mechanism is not applied.

To this end, the equation (2) for the transformer block h of our Spatio-Temporal Transformer can be reformulated as

$$\begin{aligned}\hat{\mathbf{z}} &= \mathbf{z} + id - t\alpha(\mathbf{z})\\ \tilde{\mathbf{z}} &= \hat{\mathbf{z}} + s\alpha(\hat{\mathbf{z}})\\ h(\mathbf{z}) &= \tilde{\mathbf{z}} + \phi(\tilde{\mathbf{z}}W^1)W^2,\end{aligned} \tag{6}$$

In that way, our transformer blocks analyse the face tokens in both space and time, capturing relevant information, without mixing the processing of tokens belonging to different identities. Regarding implementation, we follow ViT [43] and TimeSformer [48] and use the multi-head attention, LN before the residual connections, and GeGLU activation function for ϕ. Our Spatio-Temporal Transformer consists of a total of L such transformer blocks.

3) Model Output: Following the common practice [43], the final prediction is derived through a single-layer classification head, which takes as input the CLS token $\mathbf{z}_0^L$ obtained from the last transformer block and returns the confidence score for classification. Formally, this process can be described as

$$\hat{y} = \rho(\mathbf{z}_0^L) \tag{7}$$

where $\rho(\cdot)$ denotes our classifier and $\hat{y}$ is the final prediction and classification output. The CLS token has collected global information from all the face tokens and, in our case, from all the identities considered.

V. Evaluation

A. Dataset

The majority of video deepfake detection datasets available mostly contain homogeneous videos with somewhat standard framing and contexts and filming only one person at a time. For that reason, a study of the deepfake detection video datasets in the literature is first conducted to find one that would include temporal inconsistencies, various face-frame area ratios, videos with several people in the same scene at the same time, and varied in terms of deepfake generation methods and perturbations. For these reasons, ForgeryNet [49] is chosen as the most suitable dataset for our experiments. The training set consists of 163,176 videos (73,678 pristine and 89,498 fake), while the validation set, which has been taken as our test set (the ForgeryNet test set is unlabeled, so it cannot be used) comprises 14,048 videos (6,205 pristine and 7,843 fake); the frame rate ranges between 20 and 30 fps and the duration is rather different. The training and validation sets contain mostly single-identity videos with a good percentage of multi-identity sequences; for instance, in the training set, 11.7% of videos display two identities while 3.1% have three or more identities. A similar distribution is observed for the validation set (details can be found in the supplementary material). Notably, the possibility that the same actor appears both in the train and validation set is avoided by the procedure adopted by the authors of the ForgeryNet dataset. In fact, the authors first split the identities of the original videos into two subsets, training and evaluation, roughly according to a proportion of 7:3 [49]. This guarantees that any person appearing in a training video is not included in the evaluation set, and later, the evaluation subset is further divided into validation and test with an approximate ratio of 1:2. We also analyzed the face-frame area ratio of videos in this dataset, discovering its variety with videos containing faces covering an area up to almost even 100% of the entire frame. Comparing it with the videos in the DFDC dataset, the faces in the ForgeryNet videos have a more varied face-frame area ratio with a variance of 288.6, compared with only 5.4 in the case of DFDC. This means that in the ForgeryNet dataset, the people are filmed at a less standard distance, which is crucial for validating the impact of our Size Embedding. The tampered videos in the dataset have been manipulated using eight distinct methods: (1) FaceShifter [50], (2) FS-GAN [51], (3) DeepFakes [52], (4) BlendFace, (5) MMReplacement, (6) DeepFakes-StarGAN-Stack, (7) Talking-Head Video [53], and (8) ATVGNet [54]. Each video is manipulated using only one of these methods. The number of fake videos for each method can vary with more frequent manipulations such as FaceShifter or Talking-headVideo and less frequent ones such as DeepFakes-StarGAN-Stack and MMReplacement. These manipulations can be broadly categorized into two groups: *ID-Remained* and *ID-Replaced*. In the *ID-Remained* category, manipulations (composed by 7 and 8) are focused on altering the subject's face without altering the identity; in the *ID-Replaced* category (composed by methods from 1 to 6), the subject's face is substituted with a different one. To perform a more comprehensive analysis of the proposed method, we also conducted some experiments on DFDC since, even if it is composed of many more standard videos, there are 549 multi-identity test videos that offer additional cases for our approach.

B. Implementation Details

Pre-Processing: For face detection, we use the publicly available MTCNN face detector [47][1] with a similarity threshold of 45% and a minimum cluster size ratio of 20% of the whole faces. Faces are resized to an equal height and width of 224 pixels. For face clustering, we employ the scheme from [13] that applies the DBSCAN [55] clustering on face embeddings extracted with an InceptionResnetV1 [56] pretrained on VGGFace2 dataset [57] based on FaceNet [58]. For better generalization, the face sequences undergo several data augmentations during the training phase, in which many random perturbations are applied, inspired by [49]. In particular, each time that a sequence is given as input to the model, 31 randomly selected transformations are applied uniformly for all the video, such as image compression, various types of blur techniques, image flip and invert, colour editing, random noise, cutout and others.

Training Setup: MINTIME was trained in two main versions, a) MINTIME-EF, which uses an EfficientNet-B0 [31] as feature extractor and trained on DFDC dataset [59] as in [12], and b) MINTIME-XC, which uses an XceptionNet [33] as feature extractor (inspired by [11]) and trained on ForgeryNet images as in [49]. The first version is a lighter one and used as a baseline and it was trained keeping fixed the whole convolutional backbone excluding the last 2 blocks, with a batch size of 8. The MINTIME-XC was trained end-to-end with a batch size of 32. Both models are trained for 30 epochs. The optimizer used is SGD with a learning rate of 0.01, which decays to 0.0001 using a cosine scheduler. The weight decay was set to 0.0001 as in [48]. All models were trained considering a maximum of two identities per input sequence, which does not limit the possibility of using more or fewer identities at inference time. The maximum number of faces we put in the input sequence was set to 16. As previously stated, the number of faces considered for each identity to fill the input sequence is a hyperparameter and our experiments were based on the configuration of Table I. The hardware used to perform the training consisted of up to four NVIDIA A100.

Reproduced Approach: Since the SlowFast [60] model trained in ForgeryNet [49] was not available, we retrained it starting from the model pretrained on Kinetics 400 [61] in order to see how it behaved in certain contexts not reported in the original paper. This is the only training conducted in which a single identity per video is considered in order to emulate what has been done in the original ForgeryNet paper [49].

C. Metrics

To evaluate the performance of our model, we used in most contexts the AUC and accuracy calculated as $Acc = \frac{TP+TN}{TP+FP+TN+FN}$ where TP, TN, FP and FN stand for True Positives, True Negatives, False Positives and False Negatives respectively. This is because they are widely used in the literature by previous methods and because they are highly indicative of detection performance in a binary classification context such as this. Some additional metrics are occasionally used in the results, namely the False Positive Rate (FPR) calculated as $FPR = \frac{FP}{TN+FP}$ and the mean and standard deviation (STD) of accuracies. The FPR is used as a metric because it is important to have a system in the real world which does not lead to many false detections. The mean and STD are used to give an idea of the models' generalization among the forgery methods. We also used the True Positive Rate (TPR) calculated as $TPR = \frac{TP}{TP+FN}$ and True Negative Rate (TNR) calculated as $TNR = \frac{TN}{TN+FP}$ in order to evaluate the capability of the models to correctly classify both fake and pristine videos respectively. Exploiting these metrics we also calculated the Balanced Accuracy (BA) as $BA = \frac{TPR+TNR}{2}$. All the evaluations are conducted considering a threshold of 0.5.

TABLE II

VIDEO-LEVEL EVALUATION ON FORGERYNET VALIDATION SET. THE IDENTITIES COLUMN IS THE NUMBER OF CONSIDERED IDENTITIES FOR THE INFERENCE. † INDICATES THAT THE MODEL HAS BEEN TRAINED IN OUR SETUP. $*$ INDICATES THAT THE RESULT IS TAKEN FROM [49]

Model	#IDs	AUC	Acc	#Params
Cross Convolutional ViT [12]	1	75.45	69.65	101M
SlowFast R-50† [60]	1	90.86	82.59	34M
SlowFast R-50* [60]	1	93.88	**88.78**	34M
X3D-M* [62]	1	93.75	87.93	3M
MINTIME-EF	1	90.13	81.92	74M
MINTIME-EF	2	90.45	82.28	74M
MINTIME-EF	3	90.28	82.05	74M
MINTIME-XC	1	93.20	85.96	85M
MINTIME-XC	2	**94.25**	87.64	85M
MINTIME-XC	3	94.10	86.98	85M

TABLE III

EVALUATION ON MULTI-IDENTITY ONLY VIDEOS OF FORGERYNET VALIDATION SET. THE MODELS ARE ALL TRAINED IN OUR SETUP

Model	#IDs	AUC	Acc
Cross Convolutional ViT [12]	1	59.78	52.08
SlowFast R-50 [60]	1	80.92	72.63
MINTIME-EF	2	89.56	81.21
MINTIME-XC	2	**94.12**	**86.68**

VI. EXPERIMENTAL RESULTS

A. Comparison With the State of the Art

1) ForgeryNet Evaluation: According to Table II, MINTIME-XC outperforms the state-of-the-art on the ForgeryNet validation set in terms of AUC and is almost on par with SlowFast R-50 in terms of accuracy, which is, however, limited to consider a single identity in the classification phase (the value #ID indicates the number of identities considered by every model in each specific test case, with only MINTIME providing the multi-identity option, without affecting the number of considered videos). All MINTIME models are also robust in analyzing videos considering a variable number of identities without a significant loss of overall performance. To consider also a frame-by-frame method, we tested the Cross Convolutional ViT [12], which can be considered as a baseline. The method significantly underperforms compared with the

[1] https://github.com/timesler/facenet-pytorch

TABLE IV

VIDEO-LEVEL EVALUATION ON FORGERYNET VALIDATION SET IN TERMS OF TNR FOR THE PRISTINE VIDEOS, TPR ON EACH FORGERY METHOD AND ON THE OVERALL FAKE VIDEOS, FPR, MEAN AND STANDARD DEVIATION (STD) AND BA. THE MODELS ARE ALL TRAINED IN OUR SETUP

Model	**TNR ↑**	**TPR on Forgery Method ↑**								**TPR ↑**	**FPR ↓**	**Mean ↑**	**STD↓**	**BA↑**
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**					
SlowFast R-50 [60]	84.65	69.70	71.71	81.19	81.35	**78.67**	**88.43**	88.96	**92.05**	80.98	15.34	81.86	7.64	82.81
MINTIME-EF	85.84	70.05	69.75	74.55	82.05	78.14	79.59	91.49	77.03	79.44	14.16	78.72	7.09	82.64
MINTIME-XC	**88.15**	**79.94**	**84.64**	**82.17**	**84.05**	77.59	85.37	**92.03**	79.91	**84.03**	**14.06**	**83.76**	**4.49**	**86.09**

others, highlighting the importance of capturing the temporal inconsistencies, particularly in the case of ForgeryNet, where the faces are not manipulated in all the frames.

Furthermore, in Table III, we evaluate the performances of the models considering only the videos in which more than one identity occurs. In this setup, MINTIME-XC significantly outperforms our reproduced state-of-the-art SlowFast R-50 model by correctly classifying most of the videos considered. Interestingly, SlowFast R-50, trained considering only one identity per video, performs poorly on multi-identity videos as it is heavily influenced by the choice of the single identity to be analyzed. Also, the frame-by-frame method, namely Cross Convolutional ViT, completely fails to manage the multi-identity case, reporting very low performance on this subset close to random guessing. This poor performance derives from both the chosen identity and the aggregation function used to merge the individual face predictions into a single video-level prediction. For this experiment, we consider the maximum prediction obtained for the faces in each video. This highlights the necessity for models capable of handling multiple identities in the same video. In our case, we consider identities containing faces that cover larger areas of the video to be more important. However, we conduct a deeper analysis of the impact of this choice in our ablation study, where we demonstrate that our approach is not sensitive to this choice. We also conduct experiments to highlight the impact of the TCPE, the Identity-aware Attention, and the Size Embedding as further analysis.

In Table IV, we report the TNR obtained on pristine videos, the TPR obtained on the eight different deepfake methods present in ForgeryNet with its mean and std, the global TPR obtained considering all the fake videos without specifying the manipulation method and the global FPR considering all the pristine videos. The results make clear how both our models, especially MINTIME-XC, achieve highly competitive detection performance in all forgery types. SlowFast R-50 appears to be superior to the other competing models in recognizing video manipulated by method 8, namely ATVG-Net [54]. Nevertheless, according to the mean and STD, the model exhibits much higher variability in performance, unlike our MINTIME-XC, which remains consistent across different methods and achieves a lower FPR and higher overall TPR.

2) DFDC Evaluation: Although DFDC is not the best choice to validate robustness on the multi-identity case or the impact of size embedding due to the homogeneity of videos contained in it, for completeness we conducted experiments to evaluate the ability of our model to identify deepfake videos on this dataset as well. We train MINTIME-XC on the DFDC training set and compare it with several state-of-the-art methods in Table V. It can be seen that MINTIME-XC also performs reasonably well on this dataset, reporting results similar to other previous methods but without outperforming them. We also perform an evaluation considering only multi-identity videos, which, in the DFDC test set, are 549 (355 of which are pristine videos), and we notice that the performances remain at a high level, particularly in terms of AUC and Accuracy.

TABLE V

PERFORMANCES ON DFDC OFFICIAL TEST SET. ◇ INDICATES THAT THE METHOD HAS BEEN TESTED ONLY ON MULTI-IDENTITY VIDEOS CONSIDERING A MAXIMUM OF TWO IDENTITIES FOR THE SEQUENCE CONSTRUCTION

Model	**Acc**	**F1-Score**	**AUC**
XN-avg [25]	84.6	-	-
I3D [63]	80.8	-	-
LSTM [39]	79.0	-	-
TEI [64]	87.0	-	-
S-IML-T [65]	85.1	-	-
STIL [66]	**89.8**	-	-
MINTIME-XC (our)	86.4	87.0	95.2
MINTIME-XC ◇ (our)	84.7	79.6	93.1
Convolutional ViT [67]	-	77.0	84.3
Efficient ViT [12]	-	83.8	91.9
Conv. Cross ViT [12]	-	88.0	95.1
Selim EfficientNet B7 [68]	-	90.6	97.2
ViT with distillation [69]	-	**91.9**	**97.8**

TABLE VI

CROSS-FORGERY EVALUATION ON FORGERYNET VALIDATION SET. X3D-M AND SLOWFAST R-50 RESULTS ARE TAKEN FROM [49]

Method	**Forgery type (Training)**	**ID-replaced**		**ID-remained**	
		AUC	**Acc**	**AUC**	**Acc**
X3D-M [62]	**ID-replaced**	92.91	87.92	65.59	55.25
SlowFast R-50 [60]		92.88	**88.26**	64.83	52.64
MINTIME-EF		83.86	80.18	86.98	79.03
MINTIME-XC		**93.66**	86.58	**88.43**	**84.02**
X3D-M [62]	**ID-remained**	62.87	55.93	95.40	88.85
SlowFast R-50 [60]		61.50	52.70	95.47	87.96
MINTIME-EF		66.26	63.13	95.02	89.22
MINTIME-XC		**68.53**	**64.01**	**97.26**	**92.08**

3) Generalization Analysis: As pointed out in [21], the generalization capability of a deepfake detection model is crucial in order to be effectively deployed in the wild. In Table VI, we report the results obtained from the proposed models by training them on a subset of forgery methods and then testing them on the remaining ones. In particular, the models were trained with videos altered with the so-called Identity-Remained techniques, i.e., approaches that preserve

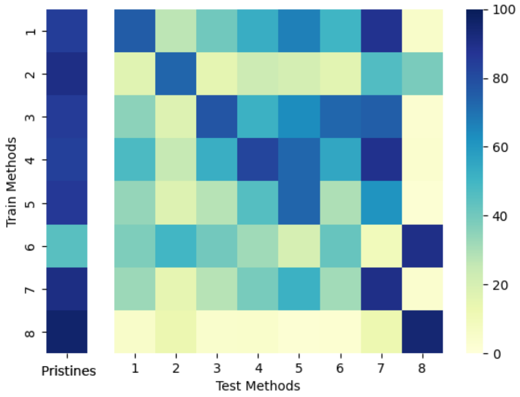


Fig. 6. Accuracy of MINTIME-XC on ForgeryNet for each method in cross-forgery training and testing scenario. The first column indicates the percentage of pristine videos correctly classified. The other columns indicate the correctly detected fake videos for each forgery method.

the identity of the manipulated subject depicted in the video. Then, the trained models are tested on videos altered with both Identity-Remained and Identity-Replaced techniques, i.e., approaches that replace the identity of the manipulated subject depicted in the video. We also repeat the process but training with videos of Identity-Replaced methods and testing on both. According to Table VI, MINTIME-XC demonstrates the best generalization capability to unseen methods, outperforming competing methods by a significant margin in almost all cases. It is worth noting that when training on Identity-Remained samples the performance on Identity-Replaced test videos is strongly compromised for all models (see second row of Table VI); in the dual case (see first row of Table VI), the same happens but not for the MINTIME models that provide a superior degree of generalization.

We conduct further analysis to see how the model behaves when trained on a single forgery method and tested on all others. For this experiment, the training set is composed by selecting all pristine videos and all the videos generated with a single forgery technique. The trained models are then tested with videos from all eight forgery methods. The results obtained are summarized in the heatmap in Figure 6 in which the first column reports the accuracy achieved by each specific trained model on pristine test samples (namely correctly classified pristine videos), while the other columns present the performance obtained by each specific trained model on the fake test samples for each of the eight different forgery methods (namely correctly classified fake videos manipulated with a specific method). It can be seen that a good generalization capability is evident in most considered contexts. In fact, although the model achieves higher accuracy on training methods in all contexts, it still manages to recognize many videos edited with techniques unseen during training, which is crucial in real-world verification tasks. The different behaviour for Method 6 is probably related to the low amount of videos manipulated with this method used during training, while Method 8 differs from the others probably because of its very particular artifacts that make it very hard for detection when training on samples coming from different forgery methods.

Finally, we evaluate MINTIME-XC trained on the ForgeryNet training set and tested on the DFDC Preview test set [70]. In Table VII, we provide a comparison with other methods tested in this context. Although our model is trained on a different dataset from other prior works, it achieves a highly competitive level of generalization with a high AUC score in a cross-dataset context.

TABLE VII

Cross-Dataset Comparison on DFDC Preview Test Set. The Previous Methods Are Trained on FaceForensics++, While MINTIME-XC Is Trained on ForgeryNet. The Identities Column Is the Number of Considered Identities During the Inference

Model	Trainset	#IDs	AUC
Face X-ray [19]	FF++	1	65.50
Patch-based [71]	FF++	1	65.60
DSP-FWA [72]	FF++	1	67.30
CSN [7]	FF++	1	68.10
Multi-Task [73]	FF++	1	68.10
CNN-GRU [74]	FF++	1	68.90
Xception [33]	FF++	1	70.90
CNN-aug [20]	FF++	1	72.10
LipForensics [36]	FF++	1	73.50
FTCN [6]	FF++	1	74.00
RealForensics [7]	FF++	1	75.90
HF Features [34]	FF++	1	**79.70**
MINTIME-EF	ForgeryNet	2	68.57
MINTIME-XC	ForgeryNet	2	**77.92**

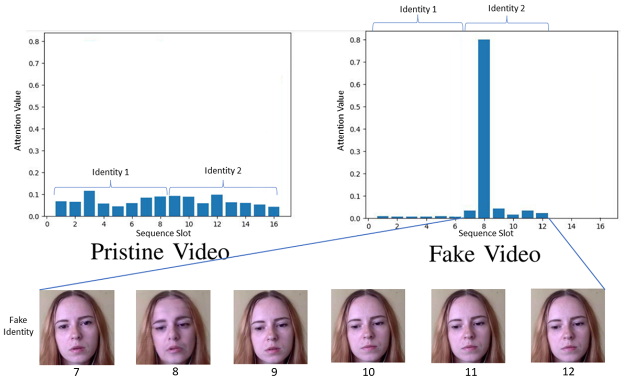


Fig. 7. Attention values computed on the 16 input faces for a pristine video (left) and a fake video (right) both containing two identities; in bottom rows, sample attention maps (only 4 faces for each identity) are presented.

B. Qualitative Evaluation

All our models have been trained to perform binary classification of the entire video. However, in the case of deepfake videos, a hypothetical final user might be interested not only in knowing whether the video has been manipulated but also at what instant and if there is more than one tampered person. These are typical requirements when such systems are provided to end users (e.g., journalists) [8].

We can derive such information by analyzing the attention values obtained on the various faces that compose the input sequence. Indeed, it has been empirically shown that when the video is pristine, there are no relevant alarms of detection, as shown in Figure 7 (left), while in the presence of a deepfake video, the model pays more attention on the faces containing traces, as shown in Figure 7 (right). Analyzing the attention values makes it rather straightforward to trace which identity

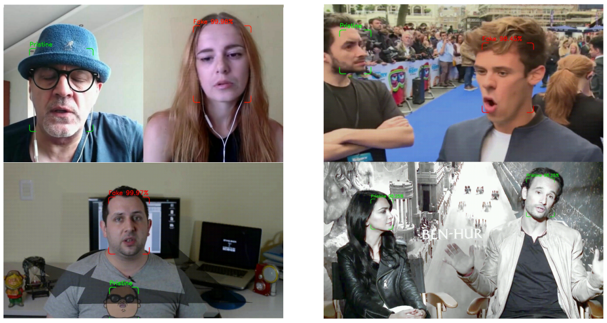


Fig. 8. Shots of outcomes obtained with MINTIME in different multi-identity contexts.

TABLE VIII
EVALUATION OF MINTIME WITH AND WITHOUT THE USAGE OF SIZE EMBEDDING ON FORGERYNET VALIDATION SET

Model	#IDs	SE	AUC	Acc
MINTIME-XC	2	✗ ✓	89.42 **94.25**	82.57 **87.64**

has been manipulated and at what instant(s) the trace is present.

Examples of outcomes from the model are shown in Figure 8. In all cases, the proposed model is able to identify the fake identity, if any, even in crowded situations. An interesting case is the one where a cartoon face picture is involved (Figure 8 bottom-left), but the model still manages to realize that the manipulated face is that of the man. The face detector detects both the cartoon face, which is originally represented on the t-shirt of the man, and the face of the man, but then the deepfake detector only evaluates the second one as manipulated (as it really is) considering the cartoon face as "pristine" in the sense of unmodified. In the case that a human face is altered by means of a cartoon face the modification would have likely been detected by the method.

C. Ablation Study

To better understand the impact of the novelties introduced in our architecture on the performance, we perform a number of ablation experiments, modifying our model and disabling its components in part.

Impact of Size Embeddings: Size-embedding is introduced to handle certain cases that rarely occur in a deepfake detection dataset but can be very common in the real world. In Table VIII, we show a comparison between MINTIME-XC trained with and without the usage of Size Embeddings on the ForgeryNet validation set. As can be seen, even in a more standard context, such as a dataset created specifically for deepfake detection, the introduction of Size Embeddings yields better results in terms of both accuracy and AUC.

To verify if the model is robust independently from the face-frame area ratio, we also constructed four sub-datasets considering the quartile distribution of face-frame area ratios of the largest face in the videos, and we tested the model with and without the usage of Size Embeddings reporting the results in Table IX. It is evident that the model has similar AUC values on all sub-datasets, and the impact of Size Embeddings is clearly apparent, particularly in situations where the face covers a small area of the frame, i.e., between 0 and 20%.

TABLE IX
IMPACT OF SIZE EMBEDDING ON SUB-DATASETS COMPOSED BY VIDEOS FROM FORGERYNET TEST SET WITH DIFFERENT FACE-FRAME AREA RATIOS

Model	Face-Frame ratio	Pristine	Fake	SE	AUC
MINTIME-XC	[0, 5%]	1345	2166	✗ ✓	87.39 **90.88**
MINTIME-XC	(5, 10%]	2000	1494	✗ ✓	90.28 **93.91**
MINTIME-XC	(10, 20%]	1570	1961	✗ ✓	90.90 **95.89**
MINTIME-XC	(20, 100%]	1796	1716	✗ ✓	90.51 **91.97**

TABLE X
EVALUATION OF MINTIME-XC WITH AND WITHOUT THE USAGE OF IDENTITY-AWARE TECHNIQUES ON MULTI-IDENTITY VIDEOS ONLY FROM FORGERYNET VALIDATION SET. THE IDENTITIES COLUMN INDICATES THE NUMBER OF IDENTITIES CONSIDERED DURING INFERENCE

Model	#IDs	TCPE	IA	AUC
MINTIME-XC	2	✗ ✓	✗ ✓	93.29 **94.12**
MINTIME-XC	3	✗ ✓	✗ ✓	90.57 **93.32**

TABLE XI
THE IMPACT OF IDENTITIES SORTING TECHNIQUES AT INFERENCE TIME ON MULTI-IDENTITY VIDEOS ONLY FROM FORGERYNET VALIDATION SET. ALL THE METHODS ARE TRAINED WITH SIZE-BASED APPROACH

Model	#IDs	Sorting Method	AUC	Acc
SlowFast R-50 [60]	1	Random Frequence-Based Size-Based	85.97 84.58 80.92	77.02 76.04 72.63
MINTIME-XC	2	Random Frequence-Based Size-Based	93.73 94.08 **94.12**	86.25 86.43 **86.68**

Impact of Identity-Aware Mechanisms: Considering only videos containing more than one person in the same scene, we can see in Table X that the proposed model performs best when Multi-Identity Attention and TCPE are used, demonstrating that these mechanisms contribute to better manage multi-identity cases.

Impact of Identity Reordering Policy: We also investigated in Table XI how the identity reordering policy could affect the performance of our model, and we made sure that we avoided inducing any bias during the training phase. To do this, we conducted several tests using different techniques. It is important to decide in which order to insert the various identities when constructing the input sequence. During training, we used the average of the areas of the faces associated with each identity as a criterion, which gave more weight to the most prominent identities in the scene (Size-based). We also tried using the number of faces associated with an identity as a criterion (Frequency-based), and finally, we compared a random criterion. Our results showed that the three different strategies did not significantly affect the performance of our proposed model, except for Random, which had a slightly higher loss. This may be because the fake identity could be discarded in favour of other identities in the scene when reordering the identities randomly. When considering a single identity per video, the results are strongly affected by the identity reordering policy, increasing the probability of missing the manipulated identity. In SlowFast experiments, it can be seen that this also leads to better results in random ordering simply by chance in the choice of the fake identity. This highlights again the importance of analyzing multiple identities correctly.

VII. Conclusion

In this study, we addressed various challenges associated with video deepfake detection in real-world scenarios and proposed a novel solution to overcome these obstacles. Our proposed model, MINTIME, achieves state-of-the-art performance on the ForgeryNet dataset and exhibits a high level of generalization in cross-forgery and cross-dataset settings, often surpassing previous approaches. MINTIME effectively manages videos featuring multiple individuals without necessitating any form of prediction aggregation, accommodates faces with varying sizes relative to the frame area, and simultaneously captures spatial and temporal inconsistencies by employing a modified TimeSformer. Moreover, the attention values of the trained models offer easy interpretability, allowing for more fine-grained predictions and extracting useful information for end-users. As a direction for future research, it would be interesting to investigate how to leverage the audio component of videos, as a complementary input to the spatial and temporal components. Furthermore, additional up-to-date datasets and more challenging real-world scenarios in the multi-identity context need to be investigated.

References

[1] I. Goodfellow et al., "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27. Red Hook, NY, USA: Curran Associates, 2014, pp. 1–9.

[2] R. Martin-Brualla, N. Radwan, M. S. M. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, "NeRF in the wild: Neural radiance fields for unconstrained photo collections," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 7210–7219.

[3] X. Zhang, P. P. Srinivasan, B. Deng, P. Debevec, W. T. Freeman, and J. T. Barron, "NeRFactor: Neural factorization of shape and reflectance under an unknown illumination," *ACM Trans. Graph.*, vol. 40, no. 6, pp. 1–18, Dec. 2021.

[4] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10674–10685.

[5] M. Van Huijstee, P. Van Boheemen, and D. Das, *Tackling Deepfakes in European Policy*. Scientific Foresight Unit (STOA), European Parliamentary Research Service (EPRS), 2021.

[6] Y. Zheng, J. Bao, D. Chen, M. Zeng, and F. Wen, "Exploring temporal coherence for more general video face forgery detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 15024–15034.

[7] A. Haliassos, R. Mira, S. Petridis, and M. Pantic, "Leveraging real talking faces via self-supervision for robust forgery detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 14930–14942.

[8] S. Baxevanakis et al., "The mever deepfake detection service: Lessons learnt from developing and deploying in the wild," in *Proc. 1st Int. Workshop Multimedia AI Against Disinformation*, 2022, pp. 1–10.

[9] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "MesoNet: A compact facial video forgery detection network," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2018, pp. 1–7.

[10] B. Bayar and M. C. Stamm, "A deep learning approach to universal image manipulation detection using a new convolutional layer," in *Proc. 4th ACM Workshop Inf. Hiding Multimedia Secur.*, Jun. 2016, pp. 5–10.

[11] S. A. Khan and D.-T. Dang-Nguyen, "Hybrid transformer network for deepfake detection," in *Proc. Int. Conf. Content-Based Multimedia Indexing*, Sep. 2022, pp. 8–14.

[12] D. A. Coccomini, N. Messina, C. Gennaro, and F. Falchi, "Combining efficientnet and vision transformers for video deepfake detection," in *Image Analysis and Processing (ICIAP)—Part III*. Lecce, Italy: Springer, 2022, pp. 219–229.

[13] P. Charitidis, G. Kordopatis-Zilos, S. Papadopoulos, and I. Kompatsiaris, "Investigating the impact of pre-processing and prediction aggregation on the deepfake detection task," in *Proc. Truth Trust Online Conf. (TTO)*, Oct. 2020, pp. 1–11.

[14] T.-N. Le, H. H. Nguyen, J. Yamagishi, and I. Echizen, "OpenForensics: Large-scale challenging dataset for multi-face forgery detection and segmentation in-the-wild," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10097–10107.

[15] H. Dang, F. Liu, J. Stehouwer, X. Liu, and A. K. Jain, "On the detection of digital face manipulation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5780–5789.

[16] Y. Qian, G. Yin, L. Sheng, Z. Chen, and J. Shao, "Thinking in frequency: Face forgery detection by mining frequency-aware clues," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Springer, 2020, pp. 86–103.

[17] L. Chai, D. Bau, S.-N. Lim, and P. Isola, "What makes fake images detectable? Understanding properties that generalize," in *Proc. ECCV*. Springer, 2020, pp. 103–120.

[18] A. Haliassos, K. Vougioukas, S. Petridis, and M. Pantic, "Lips don't lie: A generalisable and robust approach to face forgery detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 5037–5047.

[19] L. Li et al., "Face X-ray for more general face forgery detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5000–5009.

[20] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, "CNN-generated images are surprisingly easy to spot…for now," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8692–8701.

[21] D. A. Coccomini, R. Caldelli, F. Falchi, C. Gennaro, and G. Amato, "Cross-forgery analysis of vision transformers and CNNs for deepfake image detection," in *Proc. 1st Int. Workshop Multimedia AI Against Disinformation*, Jun. 2022, pp. 52–58.

[22] D. A. Coccomini, R. Caldelli, F. Falchi, and C. Gennaro, "On the generalization of deep learning models in video deepfake detection," *J. Imag.*, vol. 9, no. 5, p. 89, Apr. 2023.

[23] B. Zi, M. Chang, J. Chen, X. Ma, and Y.-G. Jiang, "WildDeepfake: A challenging real-world dataset for deepfake detection," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 2382–2390.

[24] M. Westerlund, "The emergence of deepfake technology: A review," *Technol. Innov. Manag. Rev.*, vol. 9, no. 11, pp. 39–52, 2019.

[25] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Niessner, "FaceForensics++: Learning to detect manipulated facial images," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1–11.

[26] C.-C. Hsu, C.-Y. Lee, and Y.-X. Zhuang, "Learning to detect fake face images in the wild," in *Proc. Int. Symp. Comput., Consum. Control (IS3C)*, Dec. 2018, pp. 388–391.

[27] J. Yang, A. Li, S. Xiao, W. Lu, and X. Gao, "MTD-net: Learning to detect deepfakes images by multi-scale texture difference," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 4234–4245, 2021.

[28] C. Wang and W. Deng, "Representative forgery mining for fake face detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14918–14927.

[29] Z. Hanqing, T. Wei, W. Zhou, W. Zhang, D. Chen, and N. Yu, "Multi-attentional deepfake detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 2185–2194.

[30] C. Miao, Z. Tan, Q. Chu, H. Liu, H. Hu, and N. Yu, "F2Trans: High-frequency fine-grained transformer for face forgery detection," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 1039–1051, 2023.

[31] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.

[32] C. R. Chen, Q. Fan, and R. Panda, "CrossViT: Cross-attention multiscale vision transformer for image classification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 347–356.

[33] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1800–1807.

[34] Y. Luo, Y. Zhang, J. Yan, and W. Liu, "Generalizing face forgery detection with high-frequency features," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 16312–16321.

[35] Y. Li, M.-C. Chang, and S. Lyu, "In ictu oculi: Exposing AI created fake videos by detecting eye blinking," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2018, pp. 1–7.

[36] I. Amerini, L. Galteri, R. Caldelli, and A. Del Bimbo, "Deepfake video detection through optical flow based CNN," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 1205–1207.

[37] T. Mittal, U. Bhattacharya, R. Chandra, A. Bera, and D. Manocha, "Emotions don't lie: An audio-visual deepfake detection method using affective cues," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 2823–2832.

[38] I. Amerini and R. Caldelli, "Exploiting prediction error inconsistencies through LSTM-based classifiers to detect deepfake videos," in *Proc. ACM Workshop Inf. Hiding Multimedia Secur.*, New York, NY, USA, 2020, pp. 97–102.

[39] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[40] D. Cozzolino, A. Rössler, J. Thies, M. Nießner, and L. Verdoliva, "ID-reveal: Identity-aware DeepFake video detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 15088–15097.

[41] S. Agarwal, H. Farid, T. El-Gaaly, and S.-N. Lim, "Detecting deep-fake videos from appearance and behavior," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2020, pp. 1–6.

[42] X. Dong et al., "Protecting celebrities from DeepFake with identity consistency transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 9458–9468.

[43] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," 2021, *arXiv:2010.11929*.

[44] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[45] N. Shazeer, "GLU variants improve transformer," 2020, *arXiv:2002.05202*.

[46] J. Lei Ba, J. Ryan Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.

[47] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Process. Lett.*, vol. 23, no. 10, pp. 1499–1503, Oct. 2016.

[48] G. Bertasius, H. Wang, and L. Torresani, "Is space-time attention all you need for video understanding?" in *Proc. ICML*, vol. 2, no. 3, 2021, p. 4.

[49] Y. He et al., "ForgeryNet: A versatile benchmark for comprehensive forgery analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 4358–4367.

[50] L. Li, J. Bao, H. Yang, D. Chen, and F. Wen, "Faceshifter: Towards high fidelity and occlusion aware face swapping," 2020, *arXiv:1912.13457*.

[51] Y. Nirkin, Y. Keller, and T. Hassner, "FSGAN: Subject agnostic face swapping and reenactment," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7183–7192.

[52] I. Perov et al., "DeepFaceLab: Integrated, flexible and extensible face-swapping framework," 2021, *arXiv:2005.05535*.

[53] O. Fried et al., "Text-based editing of talking-head video," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 1–14, Aug. 2019.

[54] L. Chen, R. K. Maddox, Z. Duan, and C. Xu, "Hierarchical cross-modal talking face generation with dynamic pixel-wise loss," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7824–7833.

[55] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu, "Density-based clustering in spatial databases: The algorithm GDBSCAN and its applications," *Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 169–194, 1998.

[56] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-V4, inception-resnet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 4278–4284.

[57] Q. Cao, L. Shen, W. Xie, O. Parkhi, and A. Zisserman, "VGGFace2: A dataset for recognising faces across pose and age," in *Proc. 13th IEEE Int. Conf. Autom. Face Gesture Recognit.*, May 2018, pp. 67–74.

[58] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 815–823.

[59] B. Dolhansky et al., "The DeepFake detection challenge (DFDC) dataset," 2020, *arXiv:2006.07397*.

[60] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "SlowFast networks for video recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6201–6210.

[61] W. Kay et al., "The kinetics human action video dataset," 2017, *arXiv:1705.06950*.

[62] C. Feichtenhofer, "X3D: Expanding architectures for efficient video recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 200–210.

[63] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4724–4733.

[64] Z. Liu et al., "TEINet: Towards an efficient architecture for video recognition," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 7, 2020, pp. 11669–11676.

[65] X. Li et al., "Sharp multiple instance learning for deepfake video detection," in *Proc. 28th ACM Int. Conf. Multimedia*, Seattle, WA, USA, C. W. Chen, R. Cucchiara, Z. Zhang, and R. Zimmermann, Eds. ACM, 2020, pp. 1864–1872.

[66] Z. Gu et al., "Spatiotemporal inconsistency learning for deepfake video detection," in *Proc. 29th ACM Int. Conf. Multimedia*, Chengdu, China, 2021, pp. 3473–3481.

[67] D. Wodajo and S. Atnafu, "Deepfake video detection using convolutional vision transformer," 2021, *arXiv:2102.11126*.

[68] S. Seferbekov. (2020). *DFDC 1st Place Solution*. [Online]. Available: https://github.com/selimsef/dfdc_deepfake_challenge

[69] Y.-J. Heo, Y.-J. Choi, Y.-W. Lee, and B.-G. Kim, "Deepfake detection scheme based on vision transformer and distillation," 2021, *arXiv:2104.01353*.

[70] B. Dolhansky, R. Howes, B. Pflaum, N. Baram, and C. Canton Ferrer, "The deepfake detection challenge (DFDC) preview dataset," 2019, *arXiv:1910.08854*.

[71] M. Caron et al., "Emerging properties in self-supervised vision transformers," in *Proc. ICCV*, Oct. 2021, pp. 9650–9660.

[72] Y. Li and S. Lyu, "Exposing deepfake videos by detecting face warping artifacts," in *Proc. CVPR Workshops*, 2019, pp. 1–7.

[73] H. H. Nguyen, F. Fang, J. Yamagishi, and I. Echizen, "Multi-task learning for detecting and segmenting manipulated facial images and videos," in *Proc. IEEE 10th Int. Conf. Biometrics Theory, Appl. Syst. (BTAS)*, Sep. 2019, pp. 1–8.

[74] E. Sabir, J. Cheng, A. Jaiswal, W. AbdAlmageed, I. Masi, and P. Natarajan, "Recurrent convolutional strategies for face manipulation detection in videos," in *Proc. CVPR Workshops*, 2019, pp. 1–8.

Open Access funding provided by 'Consiglio Nazionale delle Ricerche-CARI-CARE-ITALY'
within the CRUI CARE Agreement