This WACV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

VideoFACT: Detecting Video Forgeries Using Attention, Scene Context, and Forensic Traces

Tai D. Nguyen Shengbang Fang Matthew C. Stamm
Drexel University
Philadelphia, PA, USA
{tdn47, sf683, mcs382}@drexel.edu

Abstract

Fake videos represent an important misinformation threat. While existing forensic networks have demonstrated strong performance on image forgeries, recent results reported on the Adobe VideoSham dataset show that these networks fail to identify fake content in videos. In response, we propose VideoFACT - a new network that is able to detect and localize a wide variety of video forgeries and manipulations. To overcome challenges that existing networks face when analyzing videos, our network utilizes both forensic embeddings to capture traces left by manipulation, context embeddings to control for variation in forensic traces introduced by video coding, and a deep self-attention mechanism to estimate the quality and relative importance of local forensic embeddings. We create several new video forgery datasets and use these, along with publicly available data, to experimentally evaluate our network's performance. These results show that our proposed network is able to identify a diverse set of video forgeries, including those not encountered during training. Furthermore, we show that our network can be fine-tuned to achieve even stronger performance on challenging AI-based manipulations. (Code is available at: https://github.com/ductai199x/videofact-wacv-2024)

1. Introduction

Detecting fake and manipulated content in videos is critical in the fight against misinformation, online fraud, and many other threats. Traditional editing software enables users to convincingly add, remove, and alter virtually any object in a video. Furthermore, recent advances in AI-based video editing have caused dramatic advancements in how videos can be falsified. For example, AI-based video inpainting makes it possible to seamlessly remove an object from a video and replace it with a visually convincing background.

To combat forgeries, researchers have developed a wide variety of general purpose forgery detection and localization techniques. These networks operate by directly learning to detect several known forgery types [1, 2, 7, 29, 45, 46, 73, 76, 77, 81], or by searching for localized anomalies in forensic traces [12, 16, 31, 49, 72]. This research has focused nearly exclusively on images.

Figure 1. Sample video forgery localization results obtained using our proposed VideoFACT network on videos modified by deepfaking (Top), splicing in an object (Middle), and removing an object with inpainting (Bottom).

Most existing video-specific forensic techniques are aimed at detecting manipulations such as frame deletion or addition [22, 23, 60, 61, 70], speed manipulation [26], source camera model identification [25, 48], etc. [51, 64] Other research makes specific assumptions about the video's content and forgery type, e.g. detecting deepfake videos of a human speaker's face [24, 28, 30, 32, 43, 68, 74, 78], authenticating video with a still background [58], etc. Currently, there are no deep learning approaches designed to detect general content forgeries in modern video [58]. This research is further limited by the lack public datasets of general video forgeries. To the best of our knowledge, the first such dataset is Adobe's VideoSham dataset, which was published in 2023 [52] .

Suprisingly, VideoSham's benchmarking results have revealed that existing general forgery detection and localization networks all fail when applied to video [52]. This finding, which is further reinforced by results presented in this paper, can be attributed to the effects of video compression. Modern video codecs, such as H.264, utilize different coding parameters and compression strengths for different macroblocks within a single frame [56]. This introduces local variations into the forensic traces in a frame. Forensic networks misinterpret these variations as anomalous traces, causing them to false alarm. Furthermore, stronger coding in some frame regions reduces the quality of local traces, which can cause forensic networks to make incorrect decisions.

In this paper, we propose a new general video forensic network that is able to detect and localize a wide variety of fake content in video. We name our network VideoFACT: Video Forensics using Attention, Context, and Traces. To overcome the negative effects of video coding, our network contains several critical and novel components. We learn generic forensic feature embeddings for video that can capture traces left by a variety of manipulations. We observe that local video coding parameters are dependent upon contextual information, including scene content and several other factors. We exploit this by introducing the concept of context embeddings to forensics. Our network uses these embeddings to control for local variation in forensic traces introduced by video coding. We also observe that to correctly identify anomalous forensic traces left by forgery, it is better to analyze local forensic embeddings with respect to one another rather than independently. Our network does this by utilizing a deep self-attention mechanism to estimate the quality and relative importance of local forensic embeddings.

The main contributions of our paper are as follows:

(1) We propose a new forensic network that is able to perform general purpose forgery detection and localization on video. We overcome the negative effects of video coding by introducing novel network components, including: context embeddings to control for variation in forensic traces, and a deep self-attention mechanism to estimate the quality and relative importance of forensic embeddings. We show that our network can detect a wide variety of manipulations, including those that it was not explicitly trained on.

(2) We develop multiple new video forgery datasets, composed of both standard video manipulations such as splicing, and advanced manipulations such as AI-based inpainting. Currently, there are no general video forgery datasets made for network training, and only VideoSham for benchmarking. Our datasets can be used to both train and benchmark video forensic algorithms, thus helping to enable further research.

(3) We provide an extensive set of experiments to evaluate both our proposed network and state-of-the-art forensic networks. We show that VideoFACT achieves the best reported video forgery detection and localization performance, while existing approaches fail due to video coding effects. Furthermore, we show that our network can be fine-tuned to achieve even stronger performance on AI-based manipulations. We perform an ablation study showing the importance of each component of VideoFACT.

2. Background and Related Work

Image and Video Forensics. Researchers have developed multiple forensic algorithms to fight against fake content in images and videos. Early image forensic algorithms utilized hand-crafted mathematical models of editing operations like JPEG compression [21, 33, 36, 53, 63], resampling [37, 55], median filtering [9, 38], contrast enhancement [8, 41], etc. As image editing became more sophisticated, machine learning and deep learning were used to create stronger detection algorithms [**?**,7,10,45,46,69,73,81]. However, these techniques typically encounter difficulties when detecting new manipulations outside of their respective training dataset [67].

Similarly, existing deep-learning based algorithms built for videos can only detect or localize specific types of falsified content. A recent survey suggested that there exists no algorithm that can detect general video forgeries [58]. Despite the fact that an extensive body of works have dedicated to detecting splicing [11, 34, 35], inpainting [71, 75, 82], and deepfakes [30, 43, 59, 74, 78–80], these algorithms cannot be used on forgeries that they are not trained for because: 1) They have inherent input restrictions (e.g. deepfake detection algorithms need to process a cropped out face), and 2) Their designs are purposefully made for a particular manipulation type (e.g. inpainting localization techniques extract specifically inpainting traces). Therefore, to fulfill the urgent need for a general video forgery detector, we propose a new anomaly-based detection network which can analyze a wide variety of manipulations.

Anomaly-based Detection Methods. Recently, researchers are able to train general image forgery detectors by looking for anomalous traces in the residual domain of images. Notably, Forensic Similarity Graph [49], EXIF-Net [31], and Noiseprint [16] do this by divided the full-size image into small patches, then compute patch-wise difference of forensic features and cluster patches into real and fake groups. ManTraNet [72] computes the anomaly score by measuring the consistencies of the features within a kernel. MVSS-Net [12] scores the consistencies of the features in an analysis window with a multi-resolution approach. Inspired by the anomaly detection concept, our method works by: 1) Extracting not only forensic, but also context traces to account for inconsistent compression effects in modern video, and 2) Using a deep self-attention mechanism to weight and discover the manipulated region with anomalous traces.

3. Effect of Video Coding on Forensic Traces

Although state-of-the-art forensic networks are able to correctly identify fake content in images, these networks all fail when used to analyze video. This surprising behavior is due to effects that video coding has on forensic traces.

One important effect is that video coding introduces localized variations into the forensic traces within a frame. This is because modern video codecs, such as H.264, do not use the same coding parameters or compression strength for every macroblock in a frame [56]. Instead, these vary depending on several factors, including the content of a macroblock, the bit budget, the level of prediction error, etc. Because

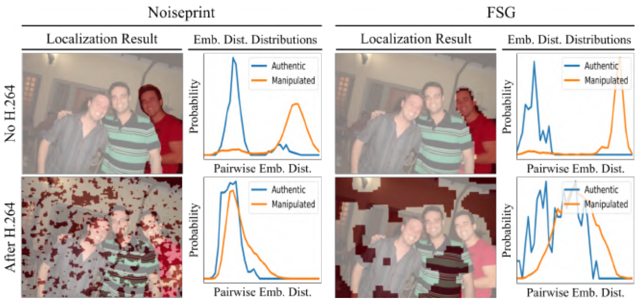

Figure 2. Example from the Carvalho Dataset [17] illustrating the influence of video compression on forensic embeddings. The distribution of pairwise forensic embedding distances across real and manipulated content as well as localization results are shown both before and after H.264 compression. We can see that video compression makes manipulated regions become indistinguishable for anomaly-based forgery detection networks like Noiseprint and FSG.

compression is well-known to significantly alter forensic traces [3,38,55], this causes traces in some authentic regions to differ significantly from those in others. These traces will then appear anomalous to forensic algorithms, which will cause them to false alarm.

Another important effect is that video coding reduces the quality of forensic traces in some regions. This is because some macroblocks are subjected to stronger compression, which is well-known to reduce the quality of forensic traces [33,44,62,63]. Existing forensic networks weight all local features equally and do not account for local variations in quality. As a result, low-quality local forensic traces can cause forensic algorithms to make incorrect decision.

We can see the effects of video compression in Fig. 2. This figure shows the distribution of pairwise differences between forensic embeddings in authentic and falsified regions obtained using Noiseprint [16] and FSG [49] on the widely benchmarked Carvalho dataset [17]. The top row shows these distributions before H.264 compression, while the bottom row shows them after compression. We observe that video coding alters these distributions such that manipulated regions become indistinguishable from authentic ones. In our experiments below, this adverse effect is proven to be detrimental for existing anomaly-based forensic networks.

4. Proposed Approach

An overview of VideoFACT's architecture is shown in Fig. 3. VideoFACT is designed to overcome challenges posed by video compression, and to accurately detect and localize a variety of fake content in video. To accomplish this, our network includes several important and novel aspects.

We utilize forensic feature embeddings specifically designed to capture traces in video. These embeddings are generic so as to detect a wide variety of manipulations.

We introduce the novel use of context embeddings to control for variation in forensic traces caused by video coding. These context embeddings capture relevant information, such as scene content, texture, lighting conditions, etc. in order to approximate local variations in compression strength. They are used to condition forensic embeddings and build a better forensic model of a local region, so that our network can accurately distinguish between authentic content and anomalies caused by forgery.

Additionally, we use a novel deep self-attention mechanism to estimate the quality and relative importance of local embeddings. This mechanism de-emphasizes embeddings from regions with low-quality traces, such as those strongly effected by compression. Similarly, it emphasizes embeddings from regions of high importance, such as those with high-quality traces or potentially anomalous traces. While other works use attention to weight the relative importance of different forensic feature subsets within an embedding, we use self-attention, which captures how relevant forensic embeddings are with respect to one another.

4.1. Low-Level Feature Extraction

VideoFACT consists of two low-level feature extractors working in tandem: the Forensic Feature Extractor (FFE) and the Context Feature Extractor (CFE). Our method first divides a frame into non-overlapping analysis blocks of size 128×128 pixels, then passes each analysis block b_k through both extractors to produce a forensic feature embedding f_k and a context feature embedding c_k.

Forensic Feature Embeddings. We use Bayar and Stamm's forensic network $g(\cdot)$ with learned high-pass filters [4] to produce dedicated forensic feature embeddings. Importantly, g is first pre-trained with a cross-entropy loss to discriminate between a video's source camera using a separate camera model identification dataset [27]. After pre-training, the final classification layer is discarded. This pre-training approach has been shown to be important for learning transferrable forensic embeddings [15,16,31,47,49,50]. We note that this branch is fixed during Training Stage 1, but is allowed to evolve during subsequent fine-tuning stages.

Context Feature Embeddings. The context feature extractor $h(\cdot)$ produces embeddings c that approximate local variations in compression strength and contextualize their forensic counterparts. To extract context information, we implement h using an Xception [13] network backbone that is modified to use only a single middle flow module, followed by a 1×1 layer to reduce the feature embedding dimension. Importantly, h is not trained until after g's pre-training is complete, so that the resulting context embeddings can provide conditional information about the distribution of f_k's. Denote θ as other layers in the network, this is equivalent to:

$$\min_{h,\theta} \mathcal{L}_T(h, g, \theta; b_1, ..., b_k, ..., b_N) \tag{1}$$

where $\mathcal{L}_T$ is VideoFACT's total loss defined in (6).

Joint Feature Extraction. After obtaining both embeddings, we produce the joint feature embeddings x by concatenating f and c, i.e.,

$$x_k = \text{concat}(f_k, c_k) \tag{2}$$

This process is repeated for every analysis block in the video

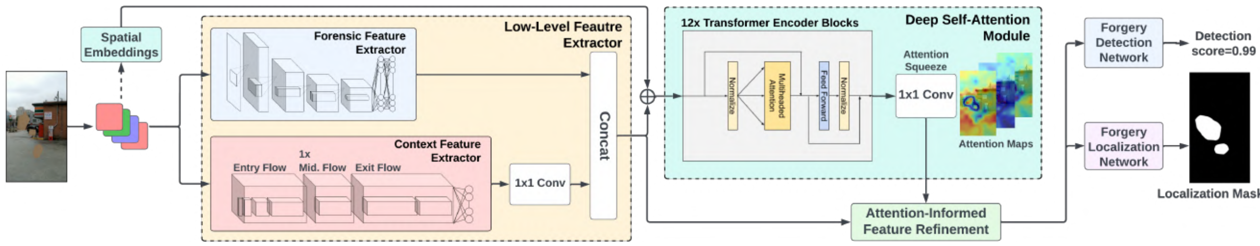

Figure 3. Overview of our proposed VideoFACT network for video forgery detection and localization. Our network extracts both forensic feature embeddings (FFE) and context feature embeddings (CFE) from local analysis blocks. These embeddings are concatenated, then weighted by attention maps produced by a Deep Self-Attention Module (DSAM). The attention-refined joint feature embeddings are passed to two different subnetworks that produce frame-level detection decisions and pixel-level forgery masks.

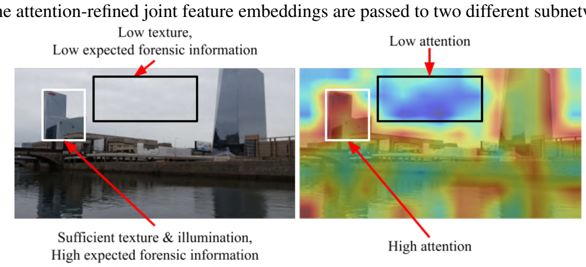

Figure 4. Example showing the effect of our attention module. The attention map produced for this frame gives large weights (shown in red) to regions with high quality forensic information - i.e. it has sufficient texture, illumination, no blurring, etc. Regions with low-quality forensic information are given small weight (shown in blue).

frame to produce N arrays of joint feature embedding with a dimension of 768.

4.2. Deep Self-Attention Module

The sequence of joint embeddings is passed to the Deep Self-Attention Module (DSAM), which is designed to produce a series of $L = 5$ different spatial attention maps that estimate the quality and relative importance of local embeddings. A vector of 1-D learnable positional embeddings is added to the joint embeddings before being passed through twelve Transformer encoder blocks stacked on top of one another. The output of these blocks is passed to an "Attention Squeeze" layer that consists of L, 1×1 convolutional kernels. This "squeezes" down the high dimensional output to a series of L different $M \times N$ spatial attention maps m_l. Each entry of the L maps m_l is the network's attention score for the corresponding joint feature embedding at that position.

Fig. 4 shows an example of a spatial attention map produced by our DSAM, as well as the corresponding the unaltered video frame. We can see that VideoFACT attends to regions containing high-quality forensic traces, i.e. regions with lower compression strength, sufficient texture and illumination, etc. By contrast, regions known to contain low-quality forensic information such as the sky are de-emphasized. Often, these regions will cause false alarms because their low-quality forensic traces will appear anomalous with respect to other high-quality forensic regions.

4.3. Attention-Informed Feature Refinement

This module enables the network to attend to feature embeddings that contain higher-quality forensic information. This is achieved by using L attention maps, produced by DSAM, to weight the low-level joint feature embeddings. The joint feature embedding x_k at the k^{th} spatial location is weighted by the corresponding k^{th} entry in the l^{th} attention map. This process is repeated for all L spatial attention maps. Then the resulting features are element-wise summed to produce the attention-refined features y, such that:

$$y_k = \sum_{\ell=1}^{L} x_k m_{k,l} \tag{3}$$

4.4. Detection and Localization

VideoFACT produces two outputs: a frame-level detection score and a pixel-level localization mask. The localization mask is disregarded if the detection score indicates no forged content exists. Our network separately analyzes the attention-refined feature embeddings using two different subnetworks for localization and detection.

Detection Network. The detection network $p(\cdot)$ consists of two 1×1 Conv + ReLU layers for dimension reduction with 200, and 2 kernels, respectively, followed by a fully connected and a softmax layer to output two neurons, one corresponds to pristine and the other corresponds to being falsified. The detection loss for this subnetwork is the cross-entropy loss between the label and the predicted output:

$$\mathcal{L}_D = -\sum_{n=1}^{2} w_n log(p_n) \tag{4}$$

where p_n is the output of $p(\cdot)$ and the w_n is the one-hot vector indicate whether the video is manipulated.

Localization Network. The localization network $q(\cdot)$ is composed of four 1×1 Convolutional layers with 192, 96, 12, and 1 kernel, respectively. We use ReLU activation except in the last layer, which uses sigmoid activation to output the block-wise probability of being manipulated.

The localization loss is built on top of this subnetwork which we define as

$$\mathcal{L}_L = -\sum_k \Big[\Big(\sum_{i,j \in \mathbb{P}_k} \frac{M_{i,j}}{|\mathbb{P}_k|}\Big) log(q_k) + \Big(1 - \sum_{i,j \in \mathbb{P}_k} \frac{M_{i,j}}{|\mathbb{P}_k|}\Big) log(1 - q_k)\Big] \tag{5}$$

where q_k is the prediction of $q(\cdot)$ for the corresponding block b_k. Because a block could contain a partial of manipulated region, $\mathbb{P}_k$ is the set of pixel coordinates that belong to block k, and M is the ground-truth binary mask.

Localization Mask Generation. During inference, we achieve the pixel-level predicted mask from the block-wise prediction. We first threshold the block-level prediction probabilities. Typically, there are two peaks in this histogram of block-level probabilities, one due to unaltered blocks and the other due to manipulated blocks. The threshold is chosen as the location of the first minima to the right of the first histogram peak (i.e. the one induced by unaltered blocks). After thresholding, we use the flood-fill morphological algorithm to remove holes from the localization mask. A final pixel-level mask is produced by scaling the block-level mask to the full video frame size using bilinear interpolation.

Total Loss. To train VideoFACT, we define the total loss as a linear combination of the detection & localization losses:

$$\mathcal{L}_T = \alpha \mathcal{L}_D + (1-\alpha)\mathcal{L}_L \tag{6}$$

where $\alpha \in (0,1)$ is the weight to balance the frame-level detection loss and the block-level localization loss.

4.5. Multi-Stage Training Protocol

Our proposed network training protocol consists of five stages. In the first three stages, the training datasets consist of videos falsified with manipulations that are successively more difficult to detect. This enables the network to progressively learn better features by refining those learned in the previous stage. In Stage 1, we use the Video Camera Model Splicing dataset (VCMS), which contains spliced content from other videos. In Stages 2 and 3, we train the network with the Video Perceptually Visible Manipulation dataset (VPVM) and Video Perceptually Invisible Manipulation datasets (VPIM), respectively. As described in Section 5, these videos contains basic manipulations with different strengths. In Stage 4, we fine-tune the network using all three previous datasets simultaneously (VCMS, VPVM and VPIM). In Stage 5, we further fine-tune the model by incorporating three auxiliary datasets made from images in the Camera Model Identification Database [4, 5] to diversify our content distribution. These datasets are made using the same process used to create the three video datasets, resulting in one dataset with spliced content (ICMS), visible manipulations (IPVM), and invisible manipulations (IPIM).

5. Video Forgery Datasets

Currently, there are no publicly available datasets of manipulated video large enough to train general content forgery detection and localization networks. Similarly, there are almost no datasets suitable to evaluate such networks, with the notable exception of the Adobe VideoSham dataset, which was released in 2023 [52]. To address this issue, we created a series of new video manipulation datasets for training and evaluating our network. These are divided into two subsets. Set A contains videos modified using standard manipulations, e.g. splicing and local editing, etc. Set B contains "In-the-Wild" videos made using sophisticated editing operations such as inpainting, deepfakes, etc. All datasets will be made publicly available upon publication of this paper.

Stage	Dataset	Optimizer	Epochs	Initial Lr	Decay rate	Decay step
1	A	SGD	6	$1.0e{-}4$	0.75	2
2	B	SGD	6	$8.5e{-}5$	0.85	2
3	C	SGD	23	$8.5e{-}5$	0.85	2
4	A, B, C	SGD	10	$8.5e{-}5$	0.85	2
5	A, B, C, D, E, F	SGD	9	$5.0e{-}5$	0.85	2

Table 1. Training parameters for different training stages of our model. We denote: A=VCMS, B=VPVM, C=VPIM, D=ICMS, E=IPVM, F=IPIM.

5.1. Set A: Standard Video Manipulations Datasets

We made three datasets by applying different sets of standard manipulations to videos from the Video-ACID [27] dataset. All three datasets were made using a common procedure. First, we created binary ground-truth masks specifying the tamper regions for each video. These tamper regions correspond to multiple randomly chosen shapes with random sizes, orientations, and placements within a frame. Fake videos were created by choosing a mask, then manipulating content within the tamper region. Original videos were retained to form the set of authentic videos. All video frames of both sets were re-encoded as H.264 videos using FFmpeg [66] with 30 FPS and constant rate factor of 23.

Each dataset in Set A corresponds to a different manipulation type. The **Video Camera Model Splicing (VCMS)** dataset contains videos with content spliced in from other videos. The **Video Perceptually Visible Manipulation (VPVM)** dataset contains content modified using common editing operations, e.g. contrast enhancement, smoothing, sharpening, blurring, etc. applied with strengths that can be visually detected. The **Video Perceptually Invisible Manipulation (VPIM)** dataset was made in a similar fashion to VPVM, but with much smaller manipulation strengths to create challenging forgeries. For each dataset, we made 3200 videos (96000 frames) for training, 520 videos (15600 frames) for validation, 280 videos (8400 frames) for testing.

5.2. Set B: In-the-Wild Manipulated Datasets

We use both publicly available datasets and datasets created by us to evaluate our network. These datasets contain advanced, challenging video forgeries with scene content that significantly differs from our training datasets.

Public Datasets. We evaluate on three publicly available datasets: VideoSham [52], DeepfakeDetectionDataset (DFD) [19], and FaceForensics++ (FF++) [57]. VideoSham contains high-quality videos manipulated by professional editors using multiple techniques. Because this work focuses exclusively on identifying fake content, we excluded videos with audio track or temporal manipulations. Both DFD and FF++ are popular deepfake benchmarking datasets that contain original videos and videos which were deepfaked using different algorithms (Face2Face [65], FaceSwap [39], etc.).

Datasets Created By Us. We also created three additional datasets to evaluate our proposed approach more comprehensively, including: E2FGVI Inpainted Videos, FuseFormer Inpainted Videos, and DeepFaceLab Deepfake

Method	VCMS				VPVM				VPIM			
	Det. mAP	*Det. ACC*	*Loc. MCC*	*Loc. F1*	*Det. mAP*	*Det. ACC*	*Loc. MCC*	*Loc. F1*	*Det. mAP*	*Det. ACC*	*Loc. MCC*	*Loc. F1*
FSG [49]	0.445	0.497	0.001	0.064	0.431	0.480	0.004	0.067	0.485	0.494	0.011	0.065
EXIFnet [31]	0.610	0.502	0.208	0.230	0.568	0.501	0.213	0.236	0.509	0.500	0.026	0.124
Noiseprint [16]	0.521	0.500	0.041	0.030	0.495	0.500	0.012	0.013	0.511	0.500	0.010	0.010
ManTra-Net [72]	0.451	0.500	0.079	0.114	0.526	0.500	0.110	0.145	0.513	0.500	0.025	0.064
MVSS-Net [12]	0.883	0.602	**0.545**	**0.557**	0.644	0.529	0.267	0.279	0.482	0.492	0.018	0.042
E.ViT [14]	0.491	0.500	N/A	N/A	0.507	0.507	N/A	N/A	0.503	0.504	N/A	N/A
CCE.ViT [14]	0.472	0.461	N/A	N/A	0.503	0.500	N/A	N/A	0.509	0.507	N/A	N/A
CNN Ensemble [6]	0.506	0.521	N/A	N/A	0.495	0.493	N/A	N/A	0.486	0.487	N/A	N/A
VideoFACT	**0.995**	**0.987**	0.530	0.526	**0.980**	**0.950**	**0.676**	**0.697**	**0.869**	**0.797**	**0.515**	**0.547**

Table 2. Frame-level detection and pixel-level localization performance on Set A Standard Video Manipulations datasets - VCMS, VPVM, VPIM.

Method	E2FGVI Inpainted Videos				FuseFormer Inpainted Videos				VideoSham [52]				DeepFaceLab Deepfake Videos				DFD [19]				FF++ [57]			
	Det. mAP	*Det. ACC*	*Loc. MCC*	*Loc. F1*	*Det. mAP*	*Det. ACC*	*Loc. MCC*	*Loc. F1*	*Det. mAP*	*Det. ACC*	*Loc. MCC*	*Loc. F1*	*Det. mAP*	*Det. ACC*	*Loc. MCC*	*Loc. F1*	*Det. mAP*	*Det. ACC*	*Loc. MCC*	*Loc. F1*	*Det. mAP*	*Det. ACC*	*Loc. MCC*	*Loc. F1*
FSG [49]	0.386	0.452	0.208	0.302	0.351	0.484	**0.241**	**0.290**	0.596	0.538	0.162	0.246	0.450	0.515	0.204	0.137	0.449	0.325	0.097	0.043	0.509	**0.519**	0.144	0.113
EXIFnet [31]	0.635	0.501	0.160	0.244	0.506	0.507	0.146	0.225	0.584	0.555	0.148	0.246	0.447	0.492	0.180	0.133	0.489	0.258	0.095	0.051	0.487	**0.519**	0.141	0.073
Noiseprint [16]	0.601	0.500	0.091	0.232	0.471	0.500	0.001	0.200	0.422	0.447	0.034	0.206	0.591	0.500	0.010	0.062	0.489	0.252	0.000	0.021	0.486	0.518	0.000	0.066
ManTra-Net [72]	0.499	0.500	0.009	0.055	0.613	0.500	0.031	0.204	0.551	0.553	0.009	0.058	0.450	0.500	0.004	0.042	0.476	0.253	0.017	0.025	0.504	0.514	0.070	0.091
MVSS-Net [12]	0.341	0.435	0.058	0.227	0.230	0.359	0.029	0.206	0.595	0.449	0.142	0.096	0.464	0.498	0.199	0.189	**0.513**	**0.532**	**0.152**	**0.108**	0.499	0.487	0.133	0.164
VideoFACT	**0.782**	**0.687**	**0.225**	**0.309**	**0.652**	**0.527**	0.118	0.237	**0.691**	**0.656**	**0.193**	**0.312**	**0.666**	**0.648**	**0.415**	**0.410**	0.468	0.444	0.081	0.077	**0.529**	**0.519**	**0.160**	**0.167**
VideoFACT-FT	**0.908**	**0.820**	**0.411**	**0.445**	**0.948**	**0.846**	**0.361**	**0.411**	N/A	N/A	N/A	N/A	**0.988**	**0.922**	**0.745**	**0.732**	**0.937**	**0.804**	**0.536**	**0.490**	**0.916**	**0.837**	**0.661**	**0.645**
E.ViT [14]	0.557	0.528	N/A	N/A	0.535	0.509	N/A	N/A	0.497	0.499	N/A	N/A	0.896	0.805	N/A	N/A	0.811	0.737	N/A	N/A	0.764	0.676	N/A	N/A
CCE.ViT [14]	0.564	0.550	N/A	N/A	0.653	0.586	N/A	N/A	0.489	0.493	N/A	N/A	0.962	0.837	N/A	N/A	0.816	0.761	N/A	N/A	0.796	0.719	N/A	N/A
CNN Ensemble [6]	0.595	0.556	N/A	N/A	0.579	0.543	N/A	N/A	0.551	0.552	N/A	N/A	0.936	0.857	N/A	N/A	0.829	0.745	N/A	N/A	0.713	0.672	N/A	N/A

Table 3. Frame-level detection and pixel-level localization performance on Set B's "In-the-Wild" datasets - E2FGVI Inpainted Videos, FuseFormer Inpainted Videos, VideoSham [52], DeepFaceLab DF Videos, DFD [19], and FF++ [57]. Note that E.ViT [14], CCE.ViT [14] and CNN Ensemble [6] are trained on 100% of the deepfake training data, which represent 75% of the total data from both FF++ [57] and DFDC [18] datasets. Meanwhile, VideoFACT-FT denotes a version of VideoFACT finetuned using only 10% of each relevant dataset.

Videos. The two inpainting datasets were made by using SOTA AI-aided video inpainting algorithms, E2FGVI [40] and FuseFormer [42], to remove objects specified by segmentation masks from videos in the Densely Annotated Video Segmentation (DAVIS) dataset [54]. Each datasets have authentic and manipulated subsets with each contain of 6208 frames from 90 videos. Additionally, The DeepFaceLab Deepfake Videos dataset was made by applying DeepFaceLab [20] - a popular, high quality deepfake algorithm - on a set of publicly available videos of celebrities downloaded from YouTube. This dataset consists of authentic and manipulated subsets with each having 300 frames from 10 videos.

6. Experiments

Training Implementation. We implemented our network using PyTorch and trained it using an NVIDIA RTX 3090. The network input resolution is 1080×1920 pixels. We first pre-trained the FFE on the Video-ACID dataset using the Stochastic Gradient Descent (SGD) with an initial learning rate of $1.0\text{e}{-3}$, momentum of 0.95, and an exponential decay-rate of 0.5 for every 2 epochs. After pretraining the FFE, we trained VideoFACT following the five stages described in Section 4.5 with different training parameters shown in Table 1. Throughout the stages, we set $\alpha = 0.4$.

Evaluation Datasets. We evaluated the frame-level performance of our proposed network and competing networks on the nine datasets described in Section 5.

Evaluation Metrics. For frame-level manipulation detection, we report the mean average precision (mAP) for each dataset. Also, we provide the average accuracy (ACC) per datasets using a unified threshold of 0.5 to reflect real-world's performance. For forgery localization (i.e. pixel-level manipulation detection) we use F1 and MCC scores to evaluate the correlation between the ground-truth and predicted masks, which are binarized with a threshold of 0.5.

6.1. Detection and Localization Performance

We compared the performance of VideoFACT to several state-of-the-art (SOTA) image forensic networks including Forensic Similarity Graphs (FSG) [49], EXIFnet [31], Noiseprint [16], ManTra-Net [72], and MVSS-Net [12], representing a broad spectrum of successful techniques for performing general content forgery detection and localization in images. Frame-level detection scores are calculated for Noiseprint and ManTra-Net by computing the average normalized per-pixel detection probability. Additionally, we benchmarked against three SOTA deepfake detectors: Efficient ViT (E.ViT) [14], Convolutional Cross Efficient ViT (CCE.ViT) [14], and CNN Ensemble [6]. All three algorithms are trained on 100% of the deepfake training data, representing 75% of the total data from both the FaceForensics++ [57] and DFDC [18] datasets. Since these only perform detection, no localization results are presented.

Set A: Standard Video Manipulations. Table 2 shows the performance of both our proposed network and competing networks on the three Standard Video Manipulations datasets in Set A. These results show that VideoFACT achieves the best performance by a large margin on these datasets. The only exception is for the VCMS dataset, where MVSS-Net's localization performance is slightly better than ours, though still comparable. Except for this case, existing networks largely do no better than a random guess (i.e. mAP $=$ 0.5 and MCC $=$ 0). This phenomenon can be clearly seen in both Table 2 and the qualitative results presented in Fig 5. These results reinforce similar findings reported in the VideoSham paper [52], i.e. existing forensic networks fail when analyzing video forgeries, and deepfake detectors cannot transfer to forgeries other than deepfakes.

Set B: "In-the-Wild" Datasets. Table 3 shows the performance of both VideoFACT and competing networks on the six "In-the-Wild" datasets, which contains complex and

Figure 5. This figure shows localization results from our proposed network as well as FSG [49], EXIFnet [31], Noiseprint [16], ManTra-Net [72], and MVSS-Net [12] on six different datasets, VCMS, VPVM, VPIM, Deepfake Video, Inpainted Video and VideoSham [52]. Our proposed network correctly identifies the manipulated area in videos falsified using a wide variety of forgery operations. We note that we do not provide localization results for deepfake detectors because these algorithms only perform detection.

challenging forgeries. From these results, we see that VideoFACT outperforms existing general forgery detectors and localizers. Additionally, VideoFACT can transfer to advanced manipulations by finetuning on a small amount of data.

Specifically, on the two Inpainted Videos datasets, VideoFACT outperforms existing forensic networks and deepfake detectors by a large margin. For example, we achieved mAP $=0.782$ on E2FGVI and 0.652 on FuseFormer Inpainted Videos. Additionally, we obtain strong performance and outperform competing networks on VideoSham, which contains four difficult forgery types. Here, we achieved mAP $=0.691$ for detection and F1 $=0.312$ for localization. On the three deepfake datasets, we did not outperform deepfake detectors. This is expected because they leverage significant problem-specific information, while VideoFACT does not. Furthermore, the traces left by deepfakes and the content of these datasets differs significantly from our training data. However, through finetuning, we will show that VideoFACT can achieve strong performance on deepfakes.

Next, our experiments show that VideoFACT can transfer to advanced manipulations by finetuning using only a small portion of data. Here, VideoFACT-FT denotes a version of VideoFACT finetuned using only 10% of each relevant dataset. By finetuning on the Inpainted Videos datasets, we achieved very strong performance: mAP $=0.908$ on E2FGVI and mAP $=0.948$ on FuseFormer. Finetuning for deepfake detection using DFD and FF++ data, we were able to achieve strong performance comparable with SOTA deepfake detectors. Notably, we achieved mAP $=0.937$ on DFD, 0.916 on FF++, and 0.988 on our DeepFaceLab dataset. These results show that by only seeing very little deepfake data, our network can generalize well to this type of manipulation, achieving mAPs than dedicated deepfake detectors whose training dataset was nine times larger than ours. Here, we report frame-level deepfake detection results for fair comparison between algorithms. We note that no finetuning experiments were performed on VideoSham because it has only an evaluation set and no training set.

6.2. Discussion

From the results presented in Tables 2 and 3, we can see that existing approaches, which reported strong performance on image forgeries, largely fail on video. There are multiple reasons why this may occur.

The first reason, as mentioned in Section 3, is the effects of video compression. Video coding parameters vary for each macroblock, which induces localized inconsistencies in forensic traces. Another important reason is the false alarms effects in authentic regions. An example of this can be seen in Fig. 6. Here, existing networks produce false alarms in different ways depending on each network's design. One set of existing approaches, such as EXIFnet, FSG, and

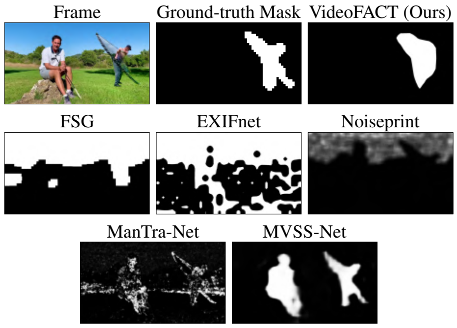

Figure 6. Example showing scene conditions that typically cause competing approaches to false alarm. Systems like FSG, EXIFnet, and Noiseprint will mistake traces in smooth regions such as the sky for anomalous traces due to editing. Networks like ManTra-Net and MVSS-Net mistake naturally occurring differences in noise statistics between foreground and background objects as caused by editing. Our network is able to use scene context and attention to control for these effects.

Setup	Component					VideoSham			
	FFE	CFE	Trans-former	Attn. maps	Data comb.	Det. ACC	Det. mAP	Loc. F1	Loc. MCC
Proposed	+	+	+	3	Add	**0.656**	**0.691**	**0.258**	**0.168**
No FFE	—	+	+	3	Add	0.610	0.646	0.209	0.118
No CFE	+	—	+	3	Add	0.586	0.635	0.163	0.043
No DSAM	+	+	—	—	—	0.601	0.626	0.144	0.000
No Transformer	+	+	—	3	Add	0.533	0.538	0.140	0.048
No Attention Squeeze	+	+	+	—	—	0.622	0.656	0.254	0.120
1 Attention Map	+	+	+	1	Add	0.610	0.655	0.175	0.121
10 Attention Maps	+	+	+	10	Add	0.622	0.676	0.212	0.127
Diff. Feat. Refine	+	+	+	3	Concat	0.614	0.684	0.162	0.091

Table 4. Ablation study of the components in our proposed network and their performance evaluations.

Noiseprint, identify fake content by searching for anomalous forensic traces. Hence, they produce false alarms in scene regions that contain low-quality forensic information such as the sky. Other approaches, such as ManTra-Net and MVSS-Net, learn forgery features by analyzing noise residuals and edge information. These techniques can false alarm when highly salient foreground objects, like the man on the rock in Fig. 6, naturally exhibit different statistics, i.e. sharper edges, from background objects.

How Our Network Overcomes These Effects. Our use of context embeddings and self-attention enables VideoFACT to be much more resilient to the effects described above. Furthermore, because the Deep Self-Attention Module learns which scene regions contain high quality forensic information, VideoFACT can rely more heavily on these regions when making decisions and avoids many of the false alarm issues faced by other networks. Finally, local context information enables our network to control for local variations in forensic traces induced by video compression.

Failure Cases and Limitations. We identified several common failure cases experienced by our network. Our network often misses falsified regions that are very small, particularly smaller than our 128×128 pixels analysis window. Our network produces poor decisions when both the manipulated region and background have similar poor illumination. It also has difficulty detecting regions altered by color swaps. Additionally, our network's current implementation is limited to analyzing video resolution of 1080p.

7. Ablation Study

We conducted multiple ablation experiments to validate the importance of various components in VideoFACT's architecture and record them in Table 4. We trained each of the network variants using the same settings as the proposed method and assessed their performance on VideoSham.

The results show that each proposed components improve the performance of the model. By removing either FFE or CFE, every metric has a significant reduction. When removing the entire DSAM, the joint embeddings were fed directly into the detector and the localizer networks. This variant still resulted in a substantial performance drop across the board. The detection metrics suggest that this network only does slightly better than random guess.

We also measure the importance of the Transformer and Attention Squeeze in the DSAM. We first replace the Transformer encoders with six fully connected layers with ReLU as the activation function. In this scenario, the model performs at a level close to random guess. Therefore, this cements the necessity of the Transformer for our network. Besides, we also try removing Attention Squeeze and connect the output of the Transformer encoders directly to the detector and localizer. We see that this variant under-performs in both detection and localization. Finally, we try using 1, 3 (proposed) and 10 attention maps. Results show that using 3 attention maps yields the best performance.

For Feature Refinement approaches, we also try concatenating all three sets of weighted spatially contextualized forensic embeddings, instead of proposed approach. Results show that both detection and localization performance drop significantly. Therefore, the proposed method is optimal.

8. Conclusion

In this paper, we propose a new network, VideoFACT, to detect and localize a broad range of video forgeries and manipulations. Our network does this by utilizing both forensic embeddings to capture traces left by manipulation, context embeddings to control for variation in forensic traces caused by video coding, and a deep self-attention mechanism to estimate the local quality and relevance of forensic embeddings. We create several new video forgery datasets, which we used along with the Adobe VideoSham dataset to experimentally evaluate our network's performance. Our results show that our proposed network is able to identify a diverse set of video forgeries, including those not encountered during training. Furthermore, our results show that existing image forensic networks largely fail to identify fake content in video.

9. Acknowledgment

This work was supported in part by ARO under cooperative grant W911NF-20-2-0111 and by DARPA and AFRL under agreement number HR0011-20-C-0126.

References

[1] Jawadul H. Bappy, Amit K. Roy-Chowdhury, Jason Bunk, Lakshmanan Nataraj, and B. S. Manjunath. Exploiting spatial structure for localizing manipulated image regions. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 1

[2] Belhassen Bayar and Matthew C Stamm. A deep learning approach to universal image manipulation detection using a new convolutional layer. In *Proceedings of the 4th ACM workshop on information hiding and multimedia security*, pages 5–10, 2016. 1

[3] Belhassen Bayar and Matthew C. Stamm. On the robustness of constrained convolutional neural networks to jpeg post-compression for image resampling detection. In *ICASSP*, pages 2152–2156, 2017. 3

[4] Belhassen Bayar and Matthew C. Stamm. Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection. *IEEE Trans. Inform. Forensics and Security*, 13(11):2691–2706, 2018. 3, 5

[5] Belhassen Bayar and Matthew C. Stamm. Towards open set camera model identification using a deep learning framework. In *ICASSP*, pages 2007–2011, 2018. 5

[6] Nicolo Bonettini, Edoardo Daniele Cannas, Sara Mandelli, Luca Bondi, Paolo Bestagini, and Stefano Tubaro. Video face manipulation detection through ensemble of cnns. In *2020 25th international conference on pattern recognition (ICPR)*, pages 5012–5019. IEEE, 2021. 6

[7] Jason Bunk, Jawadul H. Bappy, Tajuddin Manhar Mohammed, Lakshmanan Nataraj, Arjuna Flenner, B.S. Manjunath, Shivkumar Chandrasekaran, Amit K. Roy-Chowdhury, and Lawrence Peterson. Detection and localization of image forgeries using resampling features and deep learning. In *CVPRW*, pages 1881–1889, 2017. 1, 2

[8] Gang Cao, Yao Zhao, Rongrong Ni, and Xuelong Li. Contrast enhancement-based forensics in digital images. *IEEE transactions on information forensics and security*, 9(3):515–525, 2014. 2

[9] Gang Cao, Yao Zhao, Rongrong Ni, Lifang Yu, and Huawei Tian. Forensic detection of median filtering in digital images. In *2010 IEEE international conference on multimedia and expo*, pages 89–94. IEEE, 2010. 2

[10] Keshigeyan Chandrasegaran, Ngoc-Trung Tran, Alexander Binder, and Ngai-Man Cheung. Discovering transferable forensic features for cnn-generated images detection. In *ECCV*, pages 671–689, Cham, 2022. Springer Nature Switzerland. 2

[11] Shengda Chen, Shunquan Tan, Bin Li, and Jiwu Huang. Automatic detection of object-based forgery in advanced video. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(11):2138–2151, 2015. 2

[12] Xinru Chen, Chengbo Dong, Jiaqi Ji, Juan Cao, and Xirong Li. Image manipulation detection by multi-view multi-scale supervision. In *ICCV*, pages 14185–14193, October 2021. 1, 2, 6, 7

[13] Francois Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, July 2017. 3

[14] Davide Alessandro Coccomini, Nicola Messina, Claudio Gennaro, and Fabrizio Falchi. Combining efficientnet and vision transformers for video deepfake detection. In *Image Analysis and Processing–ICIAP 2022: 21st International Conference, Lecce, Italy, May 23–27, 2022, Proceedings, Part III*, pages 219–229. Springer, 2022. 6

[15] Davide Cozzolino, Giovanni Poggi, and Luisa Verdoliva. Splicebuster: A new blind image splicing detector. In *2015 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6, 2015. 3

[16] Davide Cozzolino and Luisa Verdoliva. Noiseprint: A cnn-based camera model fingerprint. *IEEE Trans. Inform. Forensics and Security*, 15:144–159, 2020. 1, 2, 3, 6, 7

[17] Tiago José De Carvalho, Christian Riess, Elli Angelopoulou, Helio Pedrini, and Anderson de Rezende Rocha. Exposing digital image forgeries by illumination color classification. *IEEE Transactions on Information Forensics and Security*, 8(7):1182–1194, 2013. 3

[18] Brian Dolhansky, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes, Menglin Wang, and Cristian Canton Ferrer. The deepfake detection challenge (dfdc) dataset, 2020. 6

[19] Nick Dufour and Andrew Gully. Deepfakedetectiondataset - contributing data to deepfake detection research, Sep 2019. 5, 6

[20] Ivan Perov et. al. Deepfacelab: A simple, flexible and extensible face swapping framework. *CoRR*, abs/2005.05535, 2020. 6

[21] Jessica Fridrich and Jan Kodovsky. Rich models for steganalysis of digital images. *IEEE Trans. Inform. Forensics and Security*, 7(3):868–882, 2012. 2

[22] A. Gironi, M. Fontani, T. Bianchi, A. Piva, and M. Barni. A video forensic technique for detecting frame deletion and insertion. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6226–6230, 2014. 1

[23] Edgar González Fernández, Ana Lucila Sandoval Orozco, and Luis Javier García Villalba. Digital video manipulation detection technique based on compression algorithms. *IEEE Transactions on Intelligent Transportation Systems*, 23(3):2596–2605, 2022. 1

[24] David Güera and Edward J. Delp. Deepfake video detection using recurrent neural networks. In *IEEE Intl. Conf. Adv. Video and Signal Based Surveillance*, pages 1–6, 2018. 1

[25] B. Hosler, O. Mayer, B. Bayar, X. Zhao, C. Chen, J. A. Shackleford, and M. C. Stamm. A video camera model identification system using deep learning and fusion. In *ICASSP*, pages 8271–8275, 2019. 1

[26] Brian C Hosler and Matthew C Stamm. Detecting video speed manipulation. In *CVPRW*, pages 670–671, 2020. 1

[27] Brian C. Hosler, Xinwei Zhao, Owen Mayer, Chen Chen, James A. Shackleford, and Matthew C. Stamm. The video authentication and camera identification database: A new database for video forensics. *IEEE Access*, 7:76937–76948, 2019. 3, 5

[28] Chih-Chung Hsu, Yi-Xiu Zhuang, and Chia-Yen Lee. Deep fake image detection based on pairwise learning. *Journal on Applied Sciences*, 10(1), 2020. 1

[29] Xuefeng Hu, Zhihan Zhang, Zhenye Jiang, Syomantak Chaudhuri, Zhenheng Yang, and Ram Nevatia. Span: Spatial pyramid attention network for image manipulation localization. In *ECCV*, pages 312–328, Cham, 2020. Springer International Publishing. 1

[30] Yihao Huang, Felix Juefei-Xu, Qing Guo, Yang Liu, and Geguang Pu. Fakelocator: Robust localization of gan-based face manipulations. *IEEE Trans. Inform. Forensics and Security*, 17:2657–2672, 2022. 1, 2

[31] Minyoung Huh, Andrew Liu, Andrew Owens, and Alexei A. Efros. Fighting fake news: Image splice detection via learned self-consistency. In *ECCV*, September 2018. 1, 2, 3, 6, 7

[32] Liming Jiang, Ren Li, Wayne Wu, Chen Qian, and Chen Change Loy. Deeperforensics-1.0: A large-scale dataset for real-world face forgery detection. In *CVPR*, June 2020. 1

[33] Xinghao Jiang, Peisong He, Tanfeng Sun, Feng Xie, and Shilin Wang. Detection of double compression with the same coding parameters based on quality degradation mechanism analysis. *IEEE Transactions on Information Forensics and Security*, 13(1):170–185, 2018. 2, 3

[34] Xiao Jin, Zhen He, Yongwei Wang, Jiawei Yu, and Jing Xu. Towards general object-based video forgery detection via dual-stream networks and depth information embedding. *Multimedia Tools and Applications*, 81(25):35733–35749, 2022. 2

[35] Xiao Jin, Zhen He, Jing Xu, Yongwei Wang, and Yuting Su. Object-based video forgery detection via dual-stream networks. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2021. 2

[36] Lyndon Kennedy and Shih-Fu Chang. Internet image archaeology: Automatically tracing the manipulation history of photographs on the web. In *Proceedings of the 16th ACM international conference on Multimedia*, pages 349–358, 2008. 2

[37] Matthias Kirchner. Fast and reliable resampling detection by spectral analysis of fixed linear predictor residue. In *Proceedings of the 10th ACM workshop on Multimedia and security*, pages 11–20, 2008. 2

[38] Matthias Kirchner and Jessica Fridrich. On detection of median filtering in digital images. In Nasir D. Memon, Jana Dittmann, Adnan M. Alattar, and Edward J. Delp III, editors, *Media Forensics and Security II*, volume 7541, page 754110. International Society for Optics and Photonics, SPIE, 2010. 2, 3

[39] M Kowalski. Marekkowalski/faceswap: 3d face swapping implemented in python. *GitHub.[Online]. Available: https://github. com/MarekKowalski/FaceSwap*. 5

[40] Zhen Li, Cheng-Ze Lu, Jianhua Qin, Chun-Le Guo, and Ming-Ming Cheng. Towards an end-to-end framework for flow-guided video inpainting. In *CVPR*, pages 17562–17571, June 2022. 6

[41] Xufeng Lin, Chang-Tsun Li, and Yongjian Hu. Exposing image forgery through the detection of contrast enhancement. In *2013 IEEE international conference on image processing*, pages 4467–4471. IEEE, 2013. 2

[42] Rui Liu, Hanming Deng, Yangyi Huang, Xiaoyu Shi, Lewei Lu, Wenxiu Sun, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Fuseformer: Fusing fine-grained information in transformers for video inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14040–14049, 2021. 6

[43] Yuchen Luo, Yong Zhang, Junchi Yan, and Wei Liu. Generalizing face forgery detection with high-frequency features. In *CVPR*, pages 16317–16326, June 2021. 1, 2

[44] Sara Mandelli, Nicolò Bonettini, Paolo Bestagini, and Stefano Tubaro. Training cnns in presence of jpeg compression: Multimedia forensics vs computer vision. In *2020 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6, 2020. 3

[45] Francesco Marra, Diego Gragnaniello, Luisa Verdoliva, and Giovanni Poggi. A full-image full-resolution end-to-end-trainable cnn framework for image forgery detection. *IEEE Access*, 8:133488–133502, 2020. 1, 2

[46] Falko Matern, Christian Riess, and Marc Stamminger. Depth map fingerprinting and splicing detection. In *ICASSP*, pages 2782–2786, 2020. 1, 2

[47] Owen Mayer, Belhassen Bayar, and Matthew C. Stamm. Learning unified deep-features for multiple forensic tasks. In *ACM Workshop on Information Hiding and Multimedia Security*, IHMMSec '18, New York, NY, USA, 2018. Association for Computing Machinery. 3

[48] Owen Mayer, Brian Hosler, and Matthew C Stamm. Open set video camera model verification. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2962–2966. IEEE, 2020. 1

[49] Owen Mayer and Matthew C. Stamm. Exposing fake images with forensic similarity graphs. *IEEE Journal of Selected Topics in Signal Processing*, 14(5):1049–1064, 2020. 1, 2, 3, 6, 7

[50] Owen Mayer and Matthew C. Stamm. Forensic similarity for digital images. *IEEE Trans. Inform. Forensics and Security*, 15:1331–1346, 2020. 3

[51] Simone Milani, Marco Fontani, Paolo Bestagini, Mauro Barni, Alessandro Piva, Marco Tagliasacchi, and Stefano Tubaro. An overview on video forensics. *APSIPA Transactions on Signal and Information Processing*, 1:e2, 2012. 1

[52] Trisha Mittal, Ritwik Sinha, Viswanathan Swaminathan, John Collomosse, and Dinesh Manocha. Video manipulations beyond faces: A dataset with human-machine analysis. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 643–652, 2023. 1, 5, 6, 7

[53] Tomas Pevny and Jessica Fridrich. Detection of double-compression in jpeg images for applications in steganography. *IEEE Trans. Inform. Forensics and Security*, 3(2):247–258, 2008. 2

[54] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 6

[55] A.C. Popescu and H. Farid. Exposing digital forgeries by detecting traces of resampling. *IEEE Transactions on Signal Processing*, 53(2):758–767, 2005. 2, 3

[56] Iain E. Richardson. *The H.264 Advanced Video Compression Standard*. Wiley Publishing, 2nd edition, 2010. 1, 2

[57] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1–11, 2019. 5, 6

[58] Nitin Arvind Shelke and Singara Singh Kasana. A comprehensive survey on passive techniques for digital video forgery detection. *Multimedia Tools and Applications*, 80:6247–6310, 2021. 1, 2

[59] Kaede Shiohara and Toshihiko Yamasaki. Detecting deepfakes with self-blended images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18720–18729, 2022. 2

[60] Matthew C Stamm, W Sabrina Lin, and KJ Ray Liu. Forensics vs. anti-forensics: A decision and game theoretic framework. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1749–1752. IEEE, 2012. 1

[61] Matthew C Stamm, W Sabrina Lin, and KJ Ray Liu. Temporal forensics and anti-forensics for motion compensated video. *IEEE Transactions on Information Forensics and Security*, 7(4):1315–1329, 2012. 1

[62] Matthew C. Stamm and K.J. Ray Liu. Anti-forensics of digital image compression. *IEEE Transactions on Information Forensics and Security*, 6(3):1050–1065, 2011. 3

[63] Matthew C. Stamm, Steven K. Tjoa, W. Sabrina Lin, and K.J. Ray Liu. Undetectable image tampering through jpeg compression anti-forensics. In *2010 IEEE International Conference on Image Processing*, pages 2109–2112, 2010. 2, 3

[64] Matthew C. Stamm, Min Wu, and K. J. Ray Liu. Information forensics: An overview of the first decade. *IEEE Access*, 1:167–200, 2013. 1

[65] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2387–2395, 2016. 5

[66] Suramya Tomar. Converting video formats with ffmpeg. *Linux Journal*, 2006(146):10, 2006. 5

[67] Luisa Verdoliva. Media forensics and deepfakes: An overview. *IEEE Journal of Selected Topics in Signal Processing*, 14(5):910–932, 2020. 2

[68] Sheng-Yu Wang, Oliver Wang, Andrew Owens, Richard Zhang, and Alexei A. Efros. Detecting photoshopped faces by scripting photoshop. In *ICCV*, October 2019. 1

[69] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A. Efros. Cnn-generated images are surprisingly easy to spot... for now. In *CVPR*, June 2020. 2

[70] Weihong Wang and Hany Farid. Exposing digital forgeries in video by detecting double mpeg compression. In *Proceedings of the 8th ACM workshop on information hiding and multimedia security*, pages 37–47, 2006. 1

[71] Shujin Wei, Haodong Li, and Jiwu Huang. Deep video inpainting localization using spatial and temporal traces. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8957–8961. IEEE, 2022. 2

[72] Yue Wu, Wael AbdAlmageed, and Premkumar Natarajan. Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In *CVPR*, pages 9535–9544, 2019. 1, 2, 6, 7

[73] Chao Yang, Huizhou Li, Fangting Lin, Bin Jiang, and Hao Zhao. Constrained r-cnn: A general image manipulation detection model. In *ICME*, pages 1–6, 2020. 1, 2

[74] Xin Yang, Yuezun Li, and Siwei Lyu. Exposing deep fakes using inconsistent head poses. In *ICASSP*, pages 8261–8265, 2019. 1, 2

[75] Bingyao Yu, Wanhua Li, Xiu Li, Jiwen Lu, and Jie Zhou. Frequency-aware spatiotemporal transformers for video inpainting detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8188–8197, 2021. 2

[76] Luo Yu, Yujin Zhang, Hua Han, Lijun Zhang, and Fei Wu. Robust median filtering forensics by cnn-based multiple residuals learning. *IEEE Access*, 7:120594–120602, 2019. 1

[77] Ru Zhang, Feng Zhu, Jianyi Liu, and Gongshen Liu. Depth-wise separable convolutions and multi-level pooling for an efficient spatial cnn-based steganalysis. *IEEE Trans. Inform. Forensics and Security*, 15:1138–1150, 2020. 1

[78] Hanqing Zhao, Wenbo Zhou, Dongdong Chen, Tianyi Wei, Weiming Zhang, and Nenghai Yu. Multi-attentional deepfake detection. In *CVPR*, pages 2185–2194, June 2021. 1, 2

[79] Tianchen Zhao, Xiang Xu, Mingze Xu, Hui Ding, Yuanjun Xiong, and Wei Xia. Learning self-consistency for deepfake detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 15023–15033, 2021. 2

[80] Yinglin Zheng, Jianmin Bao, Dong Chen, Ming Zeng, and Fang Wen. Exploring temporal coherence for more general video face forgery detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 15044–15054, 2021. 2

[81] Peng Zhou, Xintong Han, Vlad I. Morariu, and Larry S. Davis. Learning rich features for image manipulation detection. In *CVPR*, June 2018. 1, 2

[82] Peng Zhou, Ning Yu, Zuxuan Wu, Larry S Davis, Abhinav Shrivastava, and Ser-Nam Lim. Deep video inpainting detection. *arXiv preprint arXiv:2101.11080*, 2021. 2