# Extracting camera-based fingerprints for video forensics

Davide Cozzolino Giovanni Poggi Luisa Verdoliva
University Federico II of Naples
{davide.cozzolino, poggi, verdoliv}@unina.it

## Abstract

*Video source attribution is an important operation in forensics applications. Identifying which specific device or camera model took a video can help in authorship verification, but can be also a precious source of information for detecting a possible manipulation. The key observation is that any physical device leaves peculiar traces in the acquired content, a sort of fingerprint that can be exploited to establish data provenance. Moreover, absence or modification of such traces may reveal a possible manipulation. In this paper, inspired by recent work on images, we train a neural network that enhances the model-related traces hidden in a video, extracting a sort of camera fingerprint, called video noiseprint. The net is trained on pristine videos with a Siamese strategy, minimizing distances between same-model patches, and maximizing distances between unrelated patches. Experiments show that methods based on video noiseprints perform well in major forensic tasks, such as camera model identification and video forgery localization, with no need of prior knowledge on the specific manipulation or any form of fine-tuning.*

## 1. Introduction

Creating false images and videos has never been easier, thanks to advances in computer graphics and deep learning, and to the diffusion of powerful media editing tools. This is fun, most of the times, but may also be dangerous. Visual data can be manipulated for a number of malicious or even criminal purposes, like discrediting people, falsifying news, or fabricating false evidence. Such attacks are becoming more and more frequent and sophisticated, escaping easily visual scrutiny. Fig.1 shows a very recent example. On April fools day, 2019, a video featuring a giant Amazon blimp sending out a swarm of delivery drones circulated widely on Twitter. A large number of users were freaked out by the video, labelling it as "terrifying" or "dystopian". The video was a fake, created by a digital artist in Japan. Still it was realistic enough to fool a large number of viewers. Examples like this one show how easy it is to create realistic fakes, and raises a serious alarm over the general trustfulness of multimedia assets, especially videos. The level of alarm has grown exponentially with the advent of deep learning tools. A number of publicly available packages already exist (e.g. [1]), which allow even non-expert users to generate fake videos with a high level of realism, given only the availability of large amounts of data. In this context, establishing the integrity of visual resources has become of primary importance.

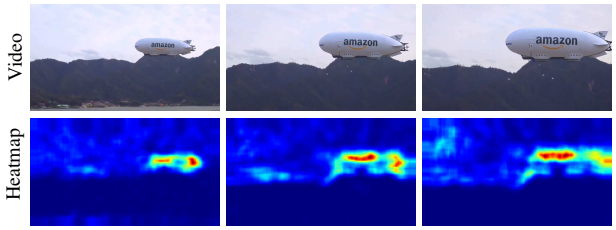


Figure 1: A video that recently appeared on the internet showing an Amazon blimp delivering drones[1]. Top: sample frames. Bottom: heatmaps obtained using our approach in a blind scenario as described in Section 5.2.

The research community has been working for several years on detecting fake content in videos [31]. Early papers focused mostly on frame-level manipulations, consisting in the deletion, insertion or replication of entire groups of frames [17, 15]. Pixel-level manipulations, targeting compact video objects and allowing for more flexible and subtle content modifications, were still difficult to perform, and hence raised a more limited attention [13, 12]. However, things are changing. With the rapid progresses in computer graphics and deep learning, removing, inserting, and copying video objects, both natural and computer-generated, is becoming easier and easier. Hence, this is by now the foremost form of video manipulation, and the object of this work.

To discover and localize manipulated material, we follow an anomaly detection strategy. Indeed, each video patch has a peculiar digital history which points to its origin, that is,

1. https://www.youtube.com/watch?v=92AMWbamo6s

the specific hardware and software involved in its generation. Of course, all patches of a video are expected to have the same source. Therefore, inconsistencies in the histories of different patches of the same video suggest that alien material was inserted, and allow one to tell it apart from pristine material.

In this work, we will rely on the traces left by the camera used for acquisition. In particular, we will leverage the image noiseprint [11], a sort of camera model fingerprint extracted from acquired images, and extend it to work on videos for applications like source identification and forgery detection. It is worth underlining that, although this approach is learning-based by definition, the training phase involves only pristine videos. Therefore, noiseprint-based video forensics is by no means limited to cameras seen in the training phase, nor to a limited set of manipulations known in advance, thereby sharing many valuable properties of blind methods.

In the rest of the paper, we will analyze related work (Section II), provide background information (Section III), give details on video noiseprint extraction (Section IV), and finally describe experimental results on source identification and forgery detection (Section V).

## 2. Related work

In the literature several different clues have been exploited to perform video forensics, often extending image-oriented methods. However, it is much more challenging working on videos than on images due to much stronger compression level. As a matter of fact, the first approaches proposed in the literature rely on compression artifacts [36, 24], with the evidence of tampering originated by double MPEG compression. Another popular approach is to detect editing based-artifacts [4, 12], which arise when a video object is copy-moved in the target video, possibly rotated or rescaled to better fit into the background.

Recently, CNN-based methods have shown great potential in video forensics, especially for advanced forms of manipulation, such as DeepFake, Face2Face and FaceSwap. However, they usually require large amount of forged/pristine data for training [33, 2, 19], or at least a form of fine tuning to newer manipulations [10].

Some other methods, addressing face manipulations, exploit visual artifacts such as eye blinking, inconsistent head poses or other facial asymmetries [25, 37, 29]. Though interesting, these approaches exploit weaknesses of current deep learning-based generation methods, that are likely to disappear with new advances in the generation phase.

On the contrary, methods based on the PRNU pattern [26] are very general and stable, and do not depend on specific artifacts. The PRNU pattern is caused by imperfections in the device manufacturing process. Because of its uniqueness and stability in time, it can be regarded as a device fingerprint, and used to perform many forensic tasks. These include source identification and forgery localization in images [6, 7] and videos [32]. On the down side, PRNU-based methods need a large number of frames coming from the target device in order to obtain reliable results. Moreover, they do not exploit all camera artifacts (actually, they suppress most of them). This latter drawback is especially relevant. In fact, digital image acquisition involves a large number of processing steps, both in-camera (e.g., interpolation, gamma correction) and out-camera (e.g., compression, enhancement), which leave many subtle traces in the final image. Different camera models are characterized by different sets of traces, due to proprietary algorithms and specific settings. Obviously, exploiting all these traces rather than just one can only provide more reliable video forensics. This is the reason for which several approaches are based on noise residuals, obtained by high-pass filtering, in order to to discover different types of video manipulations [20, 13].

The approach proposed in [11] follows this direction, and improves the residual extraction process by leveraging the power of data-driven methods. The aim is to remove effectively the semantic image content, and to enhance all camera model-related artifacts, not just some specific ones. Training is carried out using siamese networks on pristine images coming from different camera models in order to extract a camera-related fingerprint, that is, the noiseprint. Siamese networks have been used in forensics also in [30, 21]. Unlike in [11], however, the training process of [21] relies on metadata information (e.g. DigitalZoomRatio, GainControl, LensModel). A weak point is that metadata can be easily deleted or manipulated. In [30], instead, high-level camera model features are first extracted through a CNN, and then fed to a similarity network which compares them to accomplish the camera model identification task.

## 3. Background

In this section we will briefly describe the work proposed in [11]. The aim is to extract a residual image from the original one by removing the semantic scene content and highlighting camera-model artifacts. This is accomplished by training a suitable neural network. Considering that artifacts arise from a multiplicity of processes, different from camera to camera and only partially known, no mathematical model is known in the literature to describe and reproduce them. Therefore, training cannot rely on simulated examples of the desired output, and requires a more elaborate procedure.

The core idea is to use two CNNs in Siamese configuration, namely, two nets with identical architecture and weights working in parallel. The training is carried out by feeding in parallel the two branches with pairs of patches

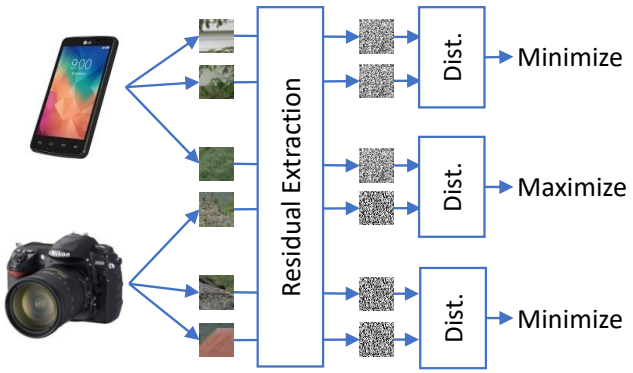


Figure 2: Building a noiseprint extraction. Patches coming from different cameras are used to train a siamese network so as to minimize the distance among patches coming from the same camera and maximing the distance among patches coming from different cameras.

extracted from the same (label +1) or different (label -1) camera model and spatial position (see Fig.2). When input patches are aligned, they can be expected to contain the same artifacts. Therefore, the output of each branch can be used as reference for the input of the other one, by-passing the need for the unavailable clean examples. We underline explicitly that, due to the spatially-varying nature of camera-model artifacts, a positive label is associated only with pairs that come from the same model *and* the same spatial position.

At the end of the training process, the CNN can be used to extract from each input image the corresponding noiseprint, displaying enhanced camera model artifacts. Of course, noiseprints will also exhibit random disturbances, including traces of the high-level scene. Nonetheless, the enhanced artifacts appear to be strong enough to provide a satisfactory basis for forensic operations. Indeed, noiseprints have already proven useful for several applications [11, 3]. In Fig.3 we show some examples of noiseprints extracted from different fake images, with the corresponding heatmaps obtained by feature clustering. In the first case the noiseprint clearly shows the $8 \times 8$ grid of the JPEG format.

Training is performed on minibatches of 200 $48 \times 48$ pixel patches. Each minibatch is formed by 50 groups of 4 homogeneous (same model, same position) patches. To boost the information conveyed by each minibatch, all available pairs of patches are used for training. Therefore, a single minibatch provides 300 positive and 19600 negative training pairs, allowing a relatively fast convergence. For a more detailed description, the reader is referred to the original paper [11].

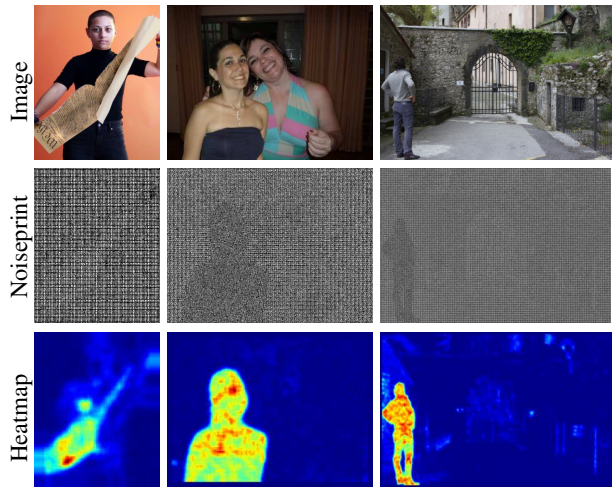


Figure 3: Examples of manipulated images (top) with extracted noiseprints (middle) and corresponding heatmaps (bottom). The first image is a real example coming from the web[2].

## 4. Training video noiseprint extractors

The image noiseprint extractor used in [11] was trained on a large dataset including 125 different cameras. It proved quite effective also on images acquired by models never seen in training. However, the statistics of videos depart significantly from those of images, and the image-oriented network does not work well on them. Therefore, in this work we performed dedicated training on suitable video datasets.

Unfortunately, video forensics is not as mature a field as image forensics, and there is a limited number of video datasets suitable for noiseprint training. In this work we rely on the VISION dataset [34], which includes videos acquired with 35 devices of 28 models. On the average, 18 videos per device are available, with duration going from 26 to 92 seconds. All videos are H.264 compressed at good quality. Following developments in the related problem of video PRNU estimation [**?**], we consider two alternative modalities to train the noiseprint extractor, *i)* using only the intra-coded frames (I frames) of the videos or *ii)* using both intra-coded and predicted frames. Given the video duration and the coding parameters, each video contributes only 26 to 142 I-frames. Therefore, although interframe prediction and low-rate coding may disrupt the weak traces we are interested in, it is possible that the information provided by additional frames proves valuable anyway.

In the training phase, all frames are treated like independent images, following the same protocol adopted in [11] to create minibatches, with the only difference that patches,

2. https://www.cnn.com/2018/03/26/us/emma-gonzalez-photo-doctored-trnd/index.html

here, have size 64×64 rather than 48×48, to compensate for the lower quality of the source. Compared with the dataset used for image noiseprints, we have a much lower variety of models and substantially less data. This may have a non-negligible impact on the overall performance, and using a richer dataset is one of the first priorities for future work.

When dealing with videos, a number of new problems arise which are not present or less relevant in the image case. In this initial analysis we skip over most of them. Most notably, we leave for future work an analysis of the impact of video recompression and resizing on the effectiveness of extracted noiseprint. However, we feel mandatory to consider the peculiar problem of video stabilization. Indeed, most videocameras use automatic stabilization of videos to compensate for unwanted user movements. As a consequence, in a stabilized video, frames are often shifted and/or rotated with respect to one another, causing a spatial misalignment of the model-related artifacts the noiseprint extractor works on. This phenomenon may impact heavily on performance, as is well known in the PRNU literature, where suitable methods have been proposed [35, 22, 28] to deal with stabilized videos. To account for this problem, we consider two different datasets, one including only 18 cameras with non-stabilized videos, corresponding to the favorable case of perfect alignment, and a second one including 26 available cameras with all their videos, irrespective of stabilization (we discarded two cameras from VISION which have a very low resolution).

Eventually, taking the combinations of interests, we consider the following three settings for training:

**a)** only I-frames, only from non stabilized videos;

**b)** only I-frames, from all videos;

**c)** all frames from all videos.

For the first setting (non stabilized) we use 18 cameras of different models (12 for training and 6 for validation), while for the other settings we use 26 cameras of different models (20 for training and 6 for validation). As for computational complexity, extracting the noiseprint of a 720×1280-pixel frame costs about 0.5 seconds using a NVIDIA Tesla P100 16GB GPU.

## 5. Experimental Analysis

We use video noiseprints for two major forensic tasks: source identification and forgery manipulation detection.

### 5.1. Source identification

In this section, we evaluate the ability of the proposed method to identify the video provenance, in particular whether two videos come from the same camera model or not. In this scenario we follow the standard pipeline used also for PRNU: first, for each camera model a reference fingerprint is extracted by averaging a certain number of frames; then, a new fingerprint is extracted from the video under test and compared with these references (see Fig. 4). To evaluate the similarity between two noiseprints (NPs) we use two different measures: the Normalized Correlation Coefficient (NCC) and the Mean Squared Error (MSE). Assuming Gaussian distributed scene-related noise, the MSE arises as the solution of a generalized likelihood ratio test, provided it is scaled by the factor $N_1 N_2/(N_1 + N_2)$ with $N_1, N_2$ the number of frames taken from the two videos to perform the estimate. In all cases, we work on the central region of $720 \times 1280$ pixels, which is the minimum resolution present in the dataset.

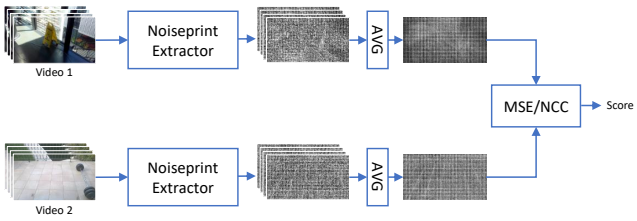


Figure 4: Block scheme of the reference-based pipeline. Noiseprint is extracted from several frames belonging to a reference video and the test video. All the noiseprints are then averaged and compared using a similarity measure.

As baseline, we consider the classic PRNU-based procedure with Peak-to-Correlation Energy (PCE) as a similarity measure [18]. In this baseline, residuals are obtained by applying a wavelet-based denoiser on the luminance component of the frames. Then, the PRNU is estimated by a maximum likelihood approach. Finally, the averages of each row and column are subtracted and a Wiener filter in the Fourier domain is applied [5]. To ensure a fair comparison, for each model we take only one device, so that camera model identification becomes equivalent to device identification. The test set is composed by 60 videos, 10 from each of 6 devices coming from the Socrates dataset [16] completely unrelated with VISION. We consider two separate cases, with non stabilized videos (Set 1) and stabilized ones (Set 2).

Results are reported in Table 1 in terms of Area Under ROC curve (AUC). On non stabilized videos (Set 1), the PRNU-based method achieves the best performance (0.907) but only if all the frames of the video are used, otherwise the AUC drops to 0.810. This is opposite to what has been shown in [8], but more in-line with recent findings [23]. The best NP-based value is 0.832, obtained using only I frames. As expected, all performance figures reduce on stabilized videos (Set 2). For the PRNU baseline the loss is dramatic, with AUC down to about 0.64, while it is much more graceful for NP-based methods, and more robust to misalignments caused by stabilization, with a top value of

| | only I-frames | | all frames | |
|---|---|---|---|---|
| | Set 1 | Set 2 | Set 1 | Set 2 |
| PRNU-based | 0.810 | 0.642 | 0.907 | 0.641 |
| NP, setting a, MSE | 0.830 | 0.752 | 0.689 | 0.679 |
| NP, setting a, NCC | 0.832 | 0.686 | 0.707 | 0.694 |
| NP, setting b, MSE | 0.832 | 0.806 | 0.686 | 0.676 |
| NP, setting b, NCC | 0.792 | 0.729 | 0.703 | 0.701 |
| NP, setting c, MSE | 0.759 | 0.827 | 0.678 | 0.720 |
| NP, setting c, NCC | 0.768 | 0.761 | 0.694 | 0.746 |

Table 1: Source identification results (AUC).

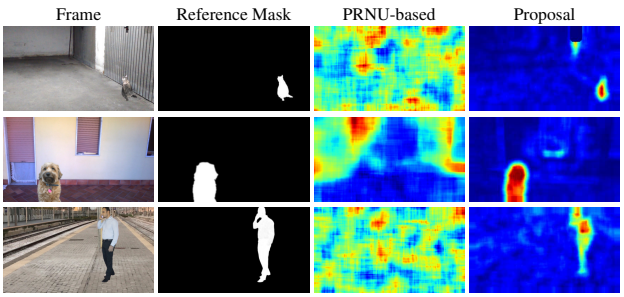


Figure 5: Some results using reference-based methods. The first two videos come from a non stabilized camera (Huawei P7mini, Nokia Lumia 520), the last one from an iPhone 7 with video stabilization.

0.827. In the next section we will apply PRNU estimation on small patches for forgery localization and compare it with the noiseprint-based approach.

### 5.2. Forgery detection and localization

In this experiment, we use video noiseprints to localize manipulations. Again, we consider various scenarios:

1. we build a reference pattern by extracting the noiseprint from a set of pristine frames coming from the same video. That is, we suppose to know *a priori* that some frames are unaltered and use the fingerprint extracted from them as a reference;

2. we restrict tests only to a suspect specific area of the video (region of interest). This is also a realistic scenario that happens either when we have two sources and want to establish which one is authentic, or when we care only for a specific area (e.g. the blimp shown in Fig. 1);

3. we do not have any type of prior information on the video and analyze it in a blind fashion.

**Scenario 1: reference pattern**. This scenario is quite similar to the pipeline shown in Fig. 4, but the techniques work in a sliding window modality on patches of dimension $128 \times 128$, computing features based on co-occurrences that are used to obtain the final heatmaps as specified in [9]. The reference pattern is estimated on 50 frames. For training the video noiseprint extractor we use setting c) and apply it on videos (stabilized and non stabilized) which include some frames known to be pristine. Noiseprints are averaged on 20 consecutive frames. Again, we compare results with the PRNU-based reference, which now performs much worse than NP-based methods, as clearly appears from the examples of Fig. 5. The main reason is that PRNU relies on a less reliable reference (only 50 frames) and is estimated on small patches.

**Scenario 2: region of interest**. We consider this scenario for the datasets created in [33], where faces have been manipulated using Face2Face (F2F), DeepFake (DF) and FaceSwap (FS). We use a face detector based on the HOG features to automatically select a bounding box including the face, while the rest of the frame is used as a reference. We adopt a strategy similar to [9] for obtaining the final heat map. In particular, for each pixel of a regular sampling grid, a feature vector is built from the spatial co-occurrences of the extracted noiseprint. Feature vectors of the reference area are used to estimate reference statistics. Then, for each feature vector of the suspect area, we compute the Mahalanobis distance w.r.t. the reference. The 99-th percentile of all these Mahalanobis distances is used as the manipulation score. This procedure is summarized in Fig. 6.

Experiments are carried out on the test set of FaceForensic++ dataset [33] which includes 140 genuine videos, 140 Face2Face manipulated videos, 140 FaceSwap manipulated videos and 140 Deepfake manipulated videos. Note that we do not use these videos to fine-tune the noiseprint extractor, which is fixed, as already said. Using the described strategy, a single score is computed for each video. To improve the estimation of noiseprint, we average it over some consecutive frames. In Fig. 7 we show the resulting ROCs. Results are always quite good, and improve significantly when the number of averaged frames grows to 20.

The average accuracy on this dataset is $92.14\%$, which is worse than than the $99.41\%$ achieved by the best CNN-based method used in [33]. However our net has never seen these (nor any other) fake videos during the training phase, and not even the pristine ones. Looking at Table 2, the best results are obtained using setting **c)** and on the FaceSwap dataset, with a Recall equal to $92.14\%$, while the most difficult manipulation to detect is Face2Face with $82.14\%$.

Finally, in Fig. 8 we show sample results from this dataset. We applied this very same method also to the video mentioned in the introduction, where the fake blimp is cor-

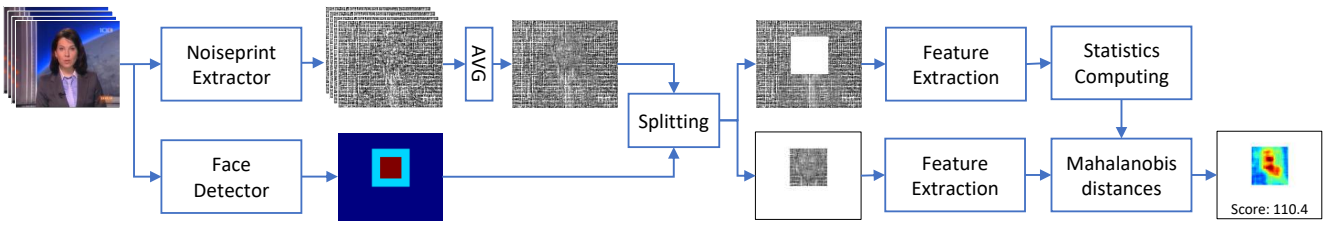


Figure 6: Block scheme for the detection of manipulations in a limited region of interest (face). Noiseprint is extracted on a certain number of frames and then averaged. Features are extracted from the background and the region of interest and the Mahalanobis distance between them is computed in order to obtain the final heat map.

| | Recall | | | | |
|---|---|---|---|---|---|
| | DF | F2F | FS | Pristine | Accuracy |
| setting b | 85.00 | 81.43 | 91.43 | 95.00 | 90.48 |
| setting c | 87.14 | 82.14 | 92.14 | 97.14 | 92.14 |

Table 2: Recall on DeepFakes (DF), Face2Face (F2F), FaceSwap (FS) and on pristine frames and accuracy on all the dataset by averaging noiseprint on 20 frames.

rectly detected (see Fig. 9) and to another video with a real blimp, which is by no means highlighted in the heatmap.

**Scenario 3: completely blind.** Here we show some results obtained in a blind scenario, where no prior information is available. In this case, noiseprint-based methods work on whole frames, and the heatmap is obtained using the same post-processing as described in [11]. Some sample results are shown in Fig. 10, compared with the results of several blind approaches proposed for images [14, 27, 21, 9, 11] and applied on individual frames. We selected the best methods, according to results in [11], among those that do not rely on JPEG artifacts. Results show that applying image-based methods to individual frames of a video does not provide satisfactory results, while a suitable video-oriented procedure, like the one proposed in this paper, can give better results.

### 5.3. Conclusions

This work represents a first attempt to extend the noiseprint approach to videos. Experiments have been conducted on two forensic applications: video camera identification and forgery detection/localization, considering various scenarios. Methods based on video noiseprints show very promising results, especially considering that the network has been trained once and for all, only on pristine videos, and never fine-tuned on data belonging to the test set. Future work will focus on improving the extraction of video noiseprints by exploiting the temporal direction both in the extraction phase and in the denoising process. We will also work on compressed videos with low quality factor to account for situations typically encountered on social networks. Advances in media editing capabilities are bound to make video forensics a more and more relevant problem, and a major research issue for the research community. Video noiseprint represents a promising new tool in the hand of the forensic analyst.

## 6. Acknowledgement

We gratefully acknowledge the support of this research by a Google Faculty Award. In addition, this material is based on research sponsored by the Air Force Research Laboratory and the Defense Advanced Research Projects Agency under agreement number FA8750-16-2-0204. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory and the Defense Advanced Research Projects Agency or the U.S. Government.

## References

[1] Deepfakes Github. https://github.com/deepfakes/faceswap. 1

[2] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen. Mesonet: a compact facial video forgery detection network. In *IEEE WIFS*, 2018. 2

[3] C. Artaud, N. Sidere, A. Doucet, J.-M. Ogier, and V. P. D. Yooz. Find it! fraud detection contest report. In *IEEE ICPR*, Aug. 2018. 3

[4] P. Bestagini, S. Milani, M. Tagliasacchi, and S. Tubaro. Local tampering detection in video sequences. In *IEEE International Workshop on Multimedia Signal Processing*, pages 488–493, October 2013. 2

[5] M. Chen, J. Fridrich, M. Goljan, and J. Lukás. Source digital camcorder identification using sensor photo response non-uniformity. In *Proc. of SPIE Conference on Security, Steganography and Watermarking of Multimedia*, 2007. 4

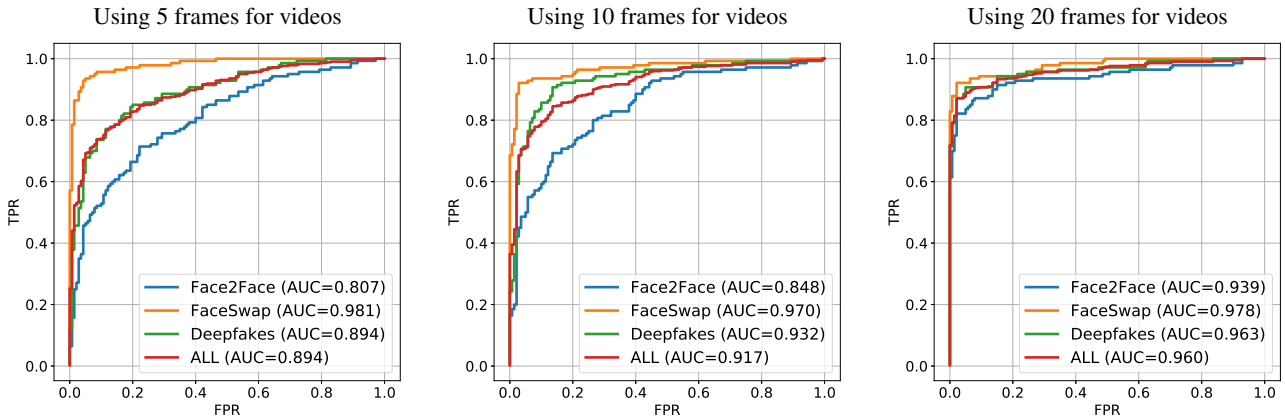


Figure 7: ROCs obtained using 5, 10 and 20 frames for video. The noiseprint extractor is trained with setting c).

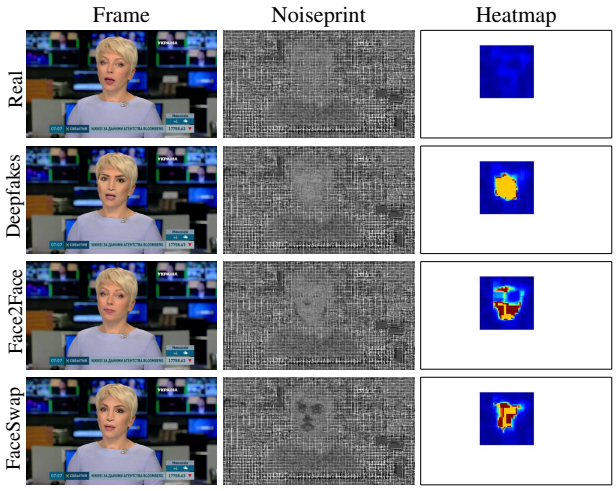


Figure 8: An example from the dataset proposed in [33]. Original frames (left), noiseprints (center) and heatmaps for the region of interest (right).

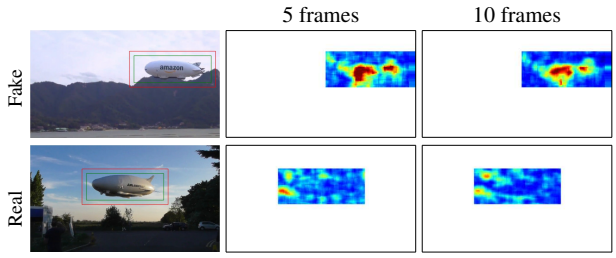


Figure 9: A single frame from two videos with a blimp and the corresponding noiseprint heatmaps over a ROI. Our method provides a higher score for the fake blimp (top) than for the real one (bottom).

[6] M. Chen, J. Fridrich, M. Goljan, and J. Lukás. Determining image origin and integrity using sensor noise. *IEEE Trans. Inf. Forensics Security*, 3(1):74–90, March 2008. 2

[7] G. Chierchia, G. Poggi, C. Sansone, and L. Verdoliva. A bayesian-MRF approach for PRNU-based image forgery detection. *IEEE Trans. Inf. Forensics Security*, 9(4):554–567, April 2014. 2

[8] W.-H. Chuang, H. Su, and M. Wu. Exploring compression effects for improved source camera identification using strongly compressed video. In *IEEE ICIP*, pages 1953–1956, 2011. 4

[9] D. Cozzolino, G. Poggi, and L. Verdoliva. Splicebuster: a new blind image splicing detector. In *IEEE WIFS*, pages 1–6, 2015. 5, 6, 8

[10] D. Cozzolino, J. Thies, A. Rössler, C. Riess, M. Nießner, and L. Verdoliva. ForensicTransfer: Weakly-supervised Domain Adaptation for Forgery Detection. *arXiv preprint arXiv:1812.02510*, 2018. 2

[11] D. Cozzolino and L. Verdoliva. Noiseprint: a CNN-based camera model fingerprint. *IEEE Trans. Inf. Forensics Security*, in press 2019. 2, 3, 6, 8

[12] L. D'Amiano, D. Cozzolino, G. Poggi, and L. Verdoliva. A PatchMatch-based dense-field algorithm for video copy-move detection and localization. *IEEE Trans. Circuits Syst. Video Technol.*, 29(3):669–682, March 2019. 1, 2

[13] D. D'Avino, D. Cozzolino, G. Poggi, and L. Verdoliva. Autoencoder with recurrent neural networks for video forgery detection. In *IS&T Electronic Imaging: Media Watermarking, Security, and Forensics*, Feb. 2017. 1, 2

[14] A. Dirik and N. Memon. Image tamper detection based on demosaicing artifacts. In *IEEE ICIP*. 6, 8

[15] C. Feng, Z. Xu, W. Zhang, and Y. Xu. Automatic location of frame deletion point for digital video forensics. In *ACM Information Hiding and Multimedia Security Workshop*, pages 171–179, 2014. 1

[16] C. Galdi, F. Hartung, and J.-L. Dugelay. Videos versus still images: asymmetric sensor pattern noise comparison on mobile phones. In *IS&T Electronic Imaging: Media Watermarking, Security and Forensics*, 2017. 4

[17] A. Gironi, M. Fontani, T. Bianchi, A. Piva, and M. Barni. A video forensic technique for detection frame deletion and insertion. In *IEEE ICASSP*, pages 6226–6230, 2014. 1

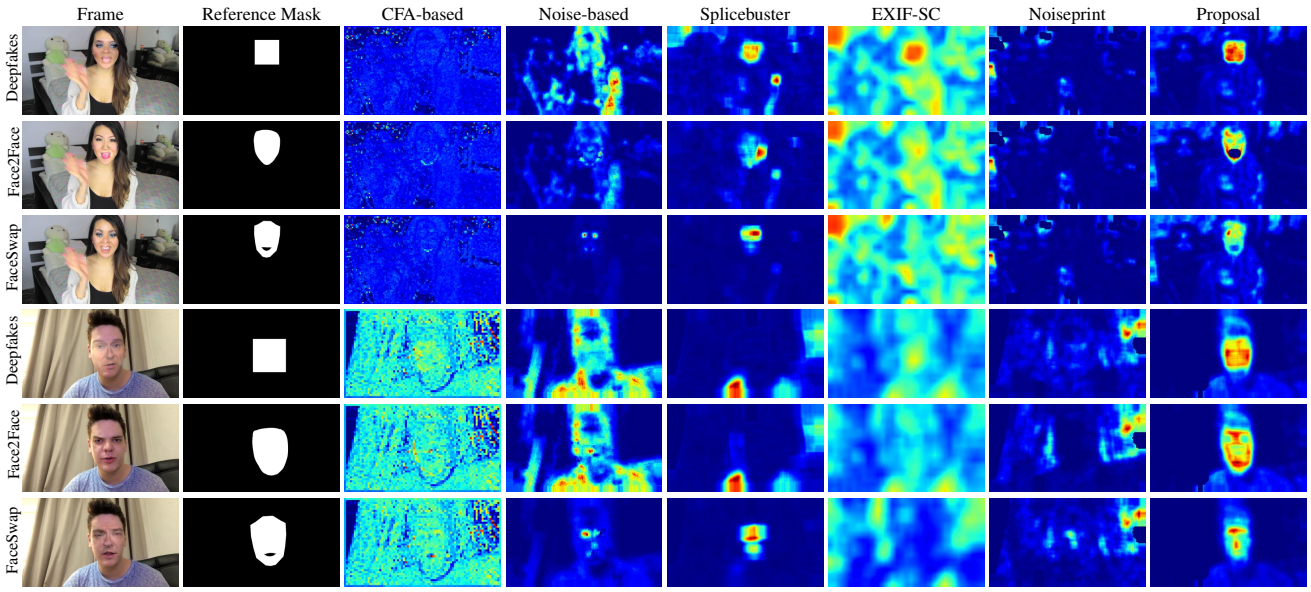


Figure 10: A comparison with state-of-the-art single-image approaches. From left to right: CFA-based [14], Noise-based [27], Splicebuster [9], EXIF-SC [21], noiseprint-image [11] and our video noiseprint approach.


[18] M. Goljan, J. Fridrich, and T. Filler. Large scale test of sensor fingerprint camera identification. In *IS&T Electronic Imaging: Media Forensics and Security*, volume 7254, 2009. 4

[19] D. Güera and E. Delp. Deepfake video detection using recurrent neural networks. In *IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2018. 2

[20] C.-C. Hsu, T.-Y. Hung, C.-W. Lin, and C.-T. Hsu. Video forgery detection using correlation of noise residue. In *IEEE MMSP Workshop*, pages 170–174, 2008. 2

[21] M. Huh, A. Liu, A. Owens, and A. Efros. Fighting fake news: Image splice detection via learned self-consistency. In *ECCV*, 2018. 2, 6, 8

[22] M. Iuliani, M. Fontani, D. Shullani, and A. Piva. Hybrid reference-based video source identification. *Sensors*, 19(3), 2019. 4

[23] E. Kouokam and A. Dirik. PRNU-based source device attribution for YouTube videos. *Digital Investigation*, 29:91–100, 2019. 4

[24] D. Labartino, T. Bianchi, A. D. Rosa, M. Fontani, D. Vazquez-Padin, A. Piva, and M. Barni. Localization of forgeries in MPEG-2 video through GOP size and DQ analysis. In *IEEE MMSP Workshop*, pages 494–499, 2013. 2

[25] Y. Li, M. Chang, and S. Lyu. In ictu oculi: Exposing AI created fake videos by detecting eye blinking. In *IEEE WIFS*, 2018. 2

[26] J. Lukas, J. Fridrich, and M. Goljan. Digital camera identification from sensor pattern noise. *IEEE Trans. Inf. Forensics Security*, 1(2):205–214, 2006. 2

[27] S. Lyu, X. Pan, and X. Zhang. Exposing region splicing forgeries with blind local noise estimation. *International Journal of Computer Vision*, 110(2):202–221, 2014. 6, 8

[28] S. Mandelli, P. Bestagini, L. Verdoliva, and S. Tubaro. Facing Device Attribution Problem for Stabilized Video Sequences. *arXiv preprint arXiv:1811.01820*, 2018. 4

[29] F. Matern, C. Riess, and M. Stamminger. Exploiting Visual Artifacts to Expose Deepfakes and Face Manipulations. In *IEEE WACV Workshops*, pages 83–92, 2019. 2

[30] O. Mayer and M. Stamm. Learned forensic source similarity for unknown camera models. In *IEEE ICASSP*, pages 2012–2016, April 2018. 2

[31] S. Milani, M. Fontani, P. Bestagini, M. Barni, A. Piva, M. Tagliasacchi, and S. Tubaro. An overview on video forensics. *APSIPA Trans. on Signal and Information Processing*, 1, December 2012. 1

[32] N. Mondaini, R. Caldelli, A. Piva, M. Barni, and V. Cappellini. Detection of malevolent changes in digital video for forensic applications. In *Proc. of SPIE Conference on Security, Steganography and Watermarking of Multimedia*, volume 6505, 2007. 2

[33] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner. Faceforensics++: Learning to detect manipulated facial images. *arXiv preprint arXiv:1901.08971*, 2019. 2, 5, 7

[34] D. Shullani, M. Fontani, M. Iuliani, O. A. Shaya, and A. Piva. Vision: a video and image dataset for source identification. *EURASIP Journal on Inf. Security*, 2017. 3

[35] S. Taspinar, M. Mohanty, and N. Memon. Source camera attribution using stabilized video. In *IEEE WIFS*, 2016. 4

[36] W. Wang and H. Farid. Exposing digital forgeries in video by detecting double MPEG compression. In *ACM Workshop on Multimedia and Security*, pages 37–47, 2006. 2

[37] X. Yang, Y. Li, and S. Lyu. Exposing deep fakes using inconsistent head poses. In *IEEE ICASSP*, 2019. 2