Deepfake Video Detection Using Recurrent Neural Networks

David Güera Edward J. Delp

Video and Image Processing Laboratory (VIPER), Purdue University

Abstract

In recent months a machine learning based free software tool has made it easy to create believable face swaps in videos that leaves few traces of manipulation, in what are known as "deepfake" videos. Scenarios where these realistic fake videos are used to create political distress, blackmail someone or fake terrorism events are easily envisioned. This paper proposes a temporal-aware pipeline to automatically detect deepfake videos. Our system uses a convolutional neural network (CNN) to extract frame-level features. These features are then used to train a recurrent neural network (RNN) that learns to classify if a video has been subject to manipulation or not. We evaluate our method against a large set of deepfake videos collected from multiple video websites. We show how our system can achieve competitive results in this task while using a simple architecture.

Figure 1. Face swapping is not new. Examples such as the swap of U.S. President Lincoln's head with politician John Calhoun's body were produced in mid-19th century (left). Modern tools like FakeApp [2] have made it easy for anyone to produce "deepfakes", such as the one swapping the heads of late-night TV hosts Jimmy Fallon and John Oliver (right).

1. Introduction

The first known attempt at trying to swap someone's face, circa 1865, can be found in one of the iconic portraits of U.S. President Abraham Lincoln. The lithography, as seen in Figure 1, mixes Lincoln's head with the body of Southern politician John Calhoun. After Lincoln's assassination, demand for lithographies of him was so great that engravings of his head on other bodies appeared almost overnight [27].

Recent advances [21, 42] have radically changed the playing field of image and video manipulation. The democratization of modern tools such as Tensorflow [6] or Keras [12] coupled with the open accessibility of the recent technical literature and cheap access to compute infrastructure have propelled this paradigm shift. Convolutional autoencoders [38, 37] and generative adversarial network (GAN) [17, 7] models have made tampering images and videos, which used to be reserved to highly-trained professionals, a broadly accessible operation within reach of almost any individual with a computer. Smartphone and desktop applications like FaceApp [1] and FakeApp [2] are built upon this progress.

FaceApp automatically generates highly realistic transformations of faces in photographs. It allows one to change face hair style, gender, age and other attributes using a smartphone. FakeApp is a desktop application that allows one to create what are now known as "deepfakes" videos. Deepfake videos are manipulated videoclips which were first created by a Reddit user, deepfake, who used TensorFlow, image search engines, social media websites and public video footage to insert someone else's face onto pre-existing videos frame by frame.

Although some benign deepfake videos exist, they remain a minority. So far, the released tools [2] that generate deepfake videos have been broadly used to create fake celebrity pornographic videos or revenge porn [5]. This kind of pornography has already been banned by sites including Reddit, Twitter, and Pornhub. The realistic nature of deepfake videos also makes them a target for generation of pedopornographic material, fake news, fake surveillance videos, and malicious hoaxes. These fake videos have already been used to create political tensions and they are being taken into account by governmental entities [4].

As presented in the Malicious AI report [11], researchers in artificial intelligence should always reflect on the dual-use nature of their work, allowing misuse considerations to influence research priorities and norms. Given the severity of the malicious attack vectors that deepfakes have caused, in this paper we present a novel solution for the detection of this kind of video.

The main contributions of this work are summarized as follows. First, we propose a two-stage analysis composed of a CNN to extract features at the frame level followed by a temporally-aware RNN network to capture temporal inconsistencies between frames introduced by the face-swapping process. Second, we have used a collection of 600 videos to evaluate the proposed method, with half of the videos being deepfakes collected from multiple video hosting websites. Third, we show experimentally the effectiveness of the described approach, which allows use to detect if a suspect video is a deepfake manipulation with 94% more accuracy than a random detector baseline in a balanced setting.

2. Related Work

Digital Media Forensics. The field of digital media forensics aims to develop technologies for the automated assessment of the integrity of an image or video. Both feature-based [35, 16] and CNN-based [18, 19] integrity analysis methods have been explored in the literature. For video-based digital forensics, the majority of the proposed solutions try to detect computationally cheap manipulations, such as dropped or duplicated frames [40] or copy-move manipulations [9]. Techniques that detect face-based manipulations include methods that distinguish computer generated faces from natural ones such as Conotter et al. [13] or Rahmouni et al. [33]. In biometry, Raghavendra et al. [32] recently proposed to detect morphed faces with two pre-trained deep CNNs and Zhou et al. [41] proposed detection of two different face swapping manipulations using a two-stream network. Of special interest to practitioners is a new dataset by Rössler et al. [34], which has about half a million edited images that have been generated with feature-based face editing [38].

Face-based Video Manipulation Methods. Multiple approaches that target face manipulations in video sequences have been proposed since the 1990s [10, 14]. Thies et al. demonstrated the first real-time expression transfer for faces and later proposed Face2Face [38], a real-time facial reenactment system, capable of altering facial movements in different types of video streams. Alternatives to Face2Face have also been proposed [8].

Several face image synthesis techniques using deep learning have also been explored as surveyed by Lu et al. [29]. Generative adversarial networks (GANs) are used for aging alterations to faces [7], or to alter face attributes such as skin color [28]. Deep feature interpolation [39] shows remarkable results in altering face attributes such as age, facial hair or mouth expressions. Similar results of attribute interpolations are achieved by Lample et al. [24]. Most of these deep learning based image synthesis techniques suffer from low image resolution. Karras et al. [22] show high-quality synthesis of faces, improving the image quality using progressive GANs.

Recurrent Neural Networks. – Long Short Term Memory (LSTM) networks are a particular type of Recurrent Neural Network (RNN), first introduced by Hochreiter and Schmidhuber [20] to learn long-term dependencies in data sequences. When a deep learning architecture is equipped with a LSTM combined with a CNN, it is typically considered as "deep in space" and "deep in time" respectively, which can be seen as two distinct system modalities. CNNs have achieved massive success in visual recognition tasks, while LSTMs are widely used for long sequence processing problems. Because of the inherent properties (rich visual description, long-term temporal memory and end-to-end training) of a convolutional LSTM architecture, it has been thoroughly studied for other computer vision tasks involving sequences (e.g. activity recognition [15] or human re-identification in videos [30]) and has lead to significant improvements.

3. Deepfake Videos Exposed

Due to the way that FakeApp [2] generates the manipulated deepfake video, intra-frame inconsistencies and temporal inconsistencies between frames are created. These video anomalies can be exploited to detect if a video under analysis is a deepfake manipulation or not. Let us briefly explain how a deepfake video is generated to understand why these anomalies are introduced in the videos and how we can exploit them.

3.1. Creating Deepfake Videos

It is well known that deep learning techniques have been successfully used to enhance the performance of image compression. Especially, the autoencoder has been applied for dimensionality reduction, compact representations of images, and generative models learning [26]. Thus, autoencoders are able to extract more compressed representations of images with a minimized loss function and are expected to achieve better compression performance than existing image compression standards. The compressed representations or latent vectors that current convolutional autoencoders learn are the first cornerstone behind the faceswapping capabilities of [2]. The second insight is the use of two sets of encoder-decoders with shared weights for the encoder networks. Figure 2 shows how these ideas are used in the training and generation phases that happen during the creation of a deepfake video.

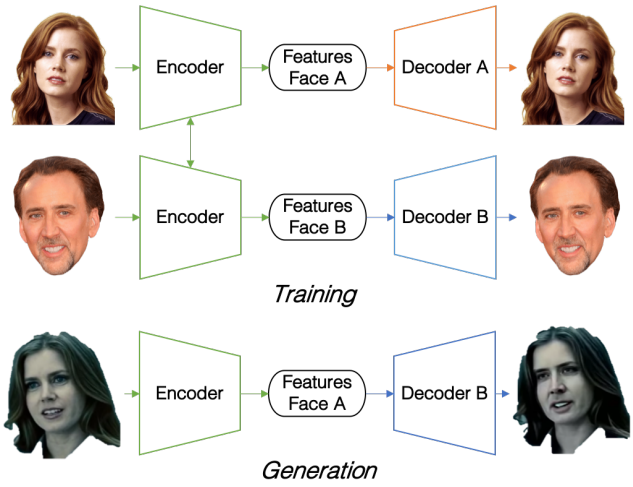

Figure 2. What makes deepfakes possible is finding a way to force both latent faces to be encoded on the same features. This is solved by having two networks sharing the same encoder, yet using two different decoders (top). When we want to do a new faceswapp, we encode the input face and decode it using the target face decoder (bottom).

3.1.1 Training

Two sets of training images are required. The first set only has samples of the original face that will be replaced, which can be extracted from the target video that will be manipulated. This first set of images can be further extended with images from other sources for more realistic results. The second set of images contains the desired face that will be swapped in the target video. To ease the training process of the autoencoders, the easiest face swap would have both the original face and target face under similar viewing and illumination conditions. However, this is usually not the case. Multiple camera views, differences in lightning conditions or simply the use of different video codecs makes it difficult for autencoders to produce realistic faces under all conditions. This usually leads to swapped faces that are visually inconsistent with the rest of the scene. This frame-level scene inconsistency will be the first feature that we will exploit with our approach.

It is also important to note that if we train two autoencoders separately, they will be incompatible with each other. If two autoencoders are trained separately on different sets of faces, their latent spaces and representations will be different. This means that each decoder is only able to decode a single kind of latent representations which it has learnt during the training phase. This can be overcome by forcing the two set of autoencoders to share the weights for the encoder networks, yet using two different decoders. In this fashion, during the training phase these two networks are treated separately and each decoder is only trained with faces from one of the subjects. However, all latent faces are produced by the same encoder which forces the encoder itself to identify common features in both faces. This can be easily accomplished due to the natural set of shared traits of all human faces (e.g. number and position of eyes, nose, ...).

3.1.2 Video Generation

When the training process is complete, we can pass a latent representation of a face generated from the original subject present in the video to the decoder network trained on faces of the subject we want to insert in the video. As shown in Figure 2, the decoder will try to reconstruct a face from the new subject, from the information relative to the original subject face present in the video. This process is repeated for every frame in the video where we want to do a faceswapping operation. It is important to point out that for doing this frame-level operation, first a face detector is used to extract only the face region that will be passed to the trained autoencoder. This is usually a second source of scene inconsistency between the swapped face and the reset of the scene. Because the encoder is not aware of the skin or other scene information it is very common to have boundary effects due to a seamed fusion between the new face and the rest of the frame.

The third major weakness that we exploit is inherent to the generation process of the final video itself. Because the autoencoder is used frame-by-frame, it is completely unaware of any previous generated face that it may have created. This lack of temporal awareness is the source of multiple anomalies. The most prominent is an inconsistent choice of illuminants between scenes with frames, with leads to a flickering phenomenon in the face region common to the majority of fake videos. Although this phenomenon can be hard to appreciate to the naked eye in the best manually-tuned deepfake manipulations, it is easily captured by a pixel-level CNN feature extractor. The phenomenon of incorrect color constancy in CNN-generated videos is a well known and still open research problem in the computer vision field [31]. Hence, it is not surprising that an autoencoder trained with very constrained data fails to render illuminants correctly.

4. Recurrent Network for Deepfake Detection

In this section, we present our end-to-end trainable recurrent deepfake video detection system (Figure 3). The proposed system is composed by a convolutional LSTM structure for processing frame sequences. There are two essential components in a convolutional LSTM:

1. CNN for frame feature extraction.

2. LSTM for temporal sequence analysis.

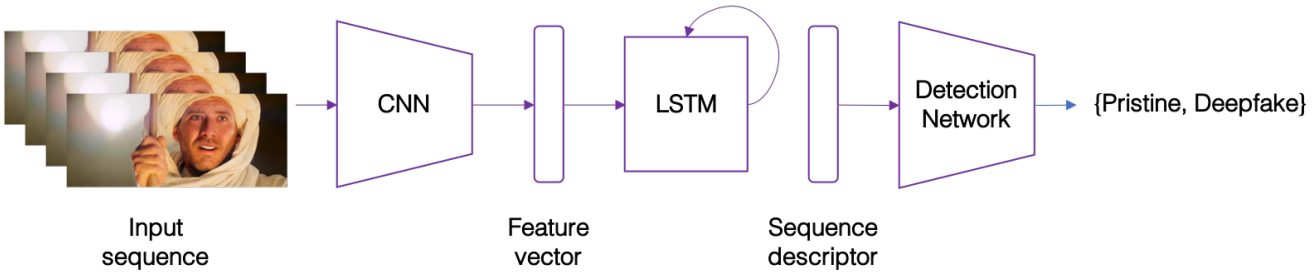

Figure 3. Overview of our detection system. The system learns and infers in an end-to-end manner and, given a video sequence, outputs a probability of it being a deepfake or a pristine video. It has a convolutional LSTM subnetwork, for processing the input temporal sequence.

Given an unseen test sequence, we obtain a set of features for each frame that are generated by the CNN. Afterwards, we concatenate the features of multiple consecutive frames and pass them to the LSTM for analysis. We finally produce an estimate of the likelihood of the sequence being either a deepfake or a nonmaninpulated video.

4.1. Convolutional LSTM

Given an image sequence (see Figure 3), a convolutional LSTM is employed to produce a temporal sequence descriptor for image manipulation of the shot frame. Aiming at end-to-end learning, an integration of fully-connected layers is used to map the high-dimensional LSTM descriptor to a final detection probability. Specifically, our shallow network consists of two fully-connected layers and one dropout layer to minimize training over-fitting. The convolutional LSTM can be divided into a CNN and a LSTM, which we will describe separately in the following paragraphs.

CNN for Feature Extraction. Inspired by its success in the IEEE Signal Processing Society Camera Model Identification Challenge, we adopt the InceptionV3 [36] with the fully-connected layer at the top of the network removed to directly output a deep representation of each frame using the ImageNet pre-trained model. Following [3], we do not fine-tune the network. The 2048-dimensional feature vectors after the last pooling layers are then used as the sequential LSTM input.

LSTM for Sequence Processing. Let us assume a sequence of CNN feature vectors of input frames as input and a 2-node neural network with the probabilities of the sequence being part of a deepfake video or an untampered video. The key challenge that we need to address is the design of a model to recursively process a sequence in a meaningful manner. For this problem, we resort to the use of a 2048-wide LSTM unit with 0.5 chance of dropout, which is capable to do exactly what we need. More particularly, during training, our LSTM model takes a sequence of 2048-dimensional ImageNet feature vectors. The LSTM is followed by a 512 fully-connected layer with 0.5 chance of dropout. Finally, we use a softmax layer to compute the probabilities of the frame sequence being either pristine or deepfake. Note that the LSTM module is an intermediate unit in our pipeline, which is trained entirely end-to-end without the need of auxiliary loss functions.

5. Experiments

In this section we report the details about our experiments. First, we describe our dataset. Then, we provide details of the experimental settings to ensure reproducibility and end up by analyzing the reported results.

5.1. Dataset

For this work, we have collected 300 deepfake videos from multiple video-hosting websites. We further incorporate 300 more videos randomly selected from the HOHA dataset [25], which leads to a final dataset with 600 videos. We selected the HOHA dataset as our source of pristine videos since it contains a realistic set of sequence samples from famous movies with an emphasis on human actions. Given that a considerable number of the deepfake videos are generated using clips from major films, using videos from the HOHA dataset further ensures that the overall system learns to spot manipulation features present in the deepfake videos, instead of memorizing semantic content from the two classes of videos present in the final dataset.

5.2. Parameter Settings

First, we have used a random 70/15/15 split to generate three disjoints sets, used for training, validation and test respectively. We do a balanced splitting, i.e., we do the splitting first for the 300 deepfake videos and then we repeat the process for the 300 nonmanipulated videos. This guarantees that each final set has exactly 50% videos of each class, which allows use to report our results in terms of accuracy without having to take into account biases due to the appearance frequency of each class or the need of using regularizing terms during the training phase. In terms of data preprocessing of the video sequences, we do:

- Subtracting channel mean from each channel.
- Resizing of every frame to 299×299.
- Sub-sequence sampling of length N controlling the length of input sequence – $N = 20, 40, 80$ frames. This allows use to see how many frames are necessary per video to have an accurate detection.
- The optimizer is set to Adam [23] for end-to-end training of the complete model with a learning rate of $1e{-}5$ and decay of $1e{-}6$.

5.3. Results

It is not unusual to find deepfake videos where the manipulation is only present in a small portion of the video (i.e. the target face only appears briefly on the video, hence the deepfake manipulation is short in time). To account for this, for every video in the training, validation and test splits, we extract continuous subsequences of fixed frame length that serve as the input of our system.

In Table 1 we present the performance of our system in terms of detection accuracy using sub-sequences of length $N = 20, 40, 80$ frames. These frame sequences are extracted sequentially (without frame skips) from each video. The entire pipeline is trained end-to-end until we reach a 10-epoch loss plateau in the validation set.

Model	Training acc. (%)	Validation acc. (%)	Test acc. (%)
Conv-LSTM, 20 frames	99.5	96.9	96.7
Conv-LSTM, 40 frames	99.3	97.1	97.1
Conv-LSTM, 80 frames	99.7	97.2	97.1

Table 1. Classification results of our dataset splits using video subsequences with different lengths.

As we can observe in our results, with less than 2 seconds of video (40 frames for videos sampled at 24 frames per second) our system can accurately predict if the fragment being analyzed comes from a deepfake video or not with an accuracy greater than 97%.

6. Conclusion

In this paper we have presented a temporal-aware system to automatically detect deepfake videos. Our experimental results using a large collection of manipulated videos have shown that using a simple convolutional LSTM structure we can accurately predict if a video has been subject to manipulation or not with as few as 2 seconds of video data.

We believe that our work offers a powerful first line of defense to spot fake media created using the tools described in the paper. We show how our system can achieve competitive results in this task while using a simple pipeline architecture. In future work, we plan to explore how to increase the robustness of our system against manipulated videos using unseen techniques during training.

Acknowledgments: This material is based on research sponsored by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL) under agreement number FA8750-16-2-0173. 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 DARPA, AFRL or the U.S. Government.

References

[1] Faceapp. https://www.faceapp.com/. (Accessed on 05/29/2018). 1

[2] Fakeapp. https://www.fakeapp.org/. (Accessed on 05/29/2018). 1, 2

[3] IEEE's Signal Processing Society - Camera Model Identification — Kaggle. https://www.kaggle.com/c/sp-society-camera-model-identification/discussion/49299. (Accessed on 05/29/2018). 4

[4] The Outline: Experts fear face swapping tech could start an international showdown. https://theoutline.com/post/3179/deepfake-videos-are-freaking-experts-out?zd=1&zi=hbmf4svs. (Accessed on 05/29/2018). 1

[5] What are deepfakes & why the future of porn is terrifying. https://www.highsnobiety.com/p/what-are-deepfakes-ai-porn/. (Accessed on 05/29/2018). 1

[6] M. Abadi et al. Tensorflow: A system for large-scale machine learning. *Proceedings of the USENIX Conference on Operating Systems Design and Implementation*, 16:265–283, Nov. 2016. Savannah, GA. 1

[7] G. Antipov, M. Baccouche, and J.-L. Dugelay. Face aging with conditional generative adversarial networks. *arXiv:1702.01983*, Feb. 2017. 1, 2

[8] H. Averbuch-Elor et al. Bringing portraits to life. *ACM Transactions on Graphics*, 36(6):196:1–196:13, Nov. 2017. 2

[9] P. Bestagini et al. Local tampering detection in video sequences. *Proceedings of the IEEE International Workshop on Multimedia Signal Processing*, pages 488–493, Sept. 2013. Pula, Italy. 2

[10] C. Bregler, M. Covell, and M. Slaney. Video rewrite: Driving visual speech with audio. *Proceedings of the ACM Annual*

Conference on Computer Graphics And Interactive Techniques, pages 353–360, Aug. 1997. Los Angeles, CA. 2

[11] M. Brundage et al. The malicious use of artificial intelligence: Forecasting, prevention, and mitigation. *arXiv:1802.07228*, Feb. 2018. 2

[12] F. Chollet et al. Keras. https://keras.io, 2015. 1

[13] V. Conotter, E. Bodnari, G. Boato, and H. Farid. Physiologically-based detection of computer generated faces in video. *Proceedings of the IEEE International Conference on Image Processing*, pages 248–252, Oct. 2014. Paris, France. 2

[14] K. Dale, K. Sunkavalli, M. K. Johnson, D. Vlasic, W. Matusik, and H. Pfister. Video face replacement. *ACM Transactions on Graphics*, 30(6):1–130, Dec. 2011. 2

[15] J. Donahue et al. Long-term recurrent convolutional networks for visual recognition and description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):677–691, Apr. 2017. 2

[16] H. Farid. *Photo Forensics*. MIT Press Ltd, 2016. 2

[17] I. Goodfellow et al. Generative adversarial nets. *Advances in Neural Information Processing Systems*, pages 2672–2680, Dec. 2014. Montréal, Canada. 1

[18] D. Güera, Y. Wang, L. Bondi, P. Bestagini, S. Tubaro, and E. J. Delp. A counter-forensic method for CNN-based camera model identification. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1840–1847, July 2017. Honolulu, HI. 2

[19] D. Güera, S. K. Yarlagadda, P. Bestagini, F. Zhu, S. Tubaro, and E. J. Delp. Reliability map estimation for cnn-based camera model attribution. *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, Mar. 2018. Lake Tahoe, NV. 2

[20] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, Nov. 1997. 2

[21] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5967–5976, July 2017. Honolulu, HI. 1

[22] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv:1710.10196*, Oct. 2017. 2

[23] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, Dec. 2014. 5

[24] G. Lample et al. Fader networks: Manipulating images by sliding attributes. *Advances in Neural Information Processing Systems*, pages 5967–5976, Dec. 2017. Long Beach, CA. 2

[25] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008. Anchorage, AK. 4

[26] Y. Liao, Y. Wang, and Y. Liu. Graph regularized auto-encoders for image representation. *IEEE Transactions on Image Processing*, 26(6):2839–2852, June 2017. 2

[27] S. Lorant. *Lincoln; a picture story of his life*. Norton, 1969. 1

[28] Y. Lu, Y.-W. Tai, and C.-K. Tang. Conditional cyclegan for attribute guided face image generation. *arXiv:1705.09966*, May 2017. 2

[29] Z. Lu, Z. Li, J. Cao, R. He, and Z. Sun. Recent progress of face image synthesis. *arXiv:1706.04717*, June 2017. 2

[30] N. McLaughlin, J. M. d. Rincon, and P. Miller. Recurrent convolutional network for video-based person re-identification. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1325–1334, June 2016. Las Vegas, NV. 2

[31] Y. Qian et al. Recurrent color constancy. *Proceedings of the IEEE International Conference on Computer Vision*, pages 5459–5467, Oct. 2017. Venice, Italy. 3

[32] R. Raghavendra, K. B. Raja, S. Venkatesh, and C. Busch. Transferable deep-cnn features for detecting digital and print-scanned morphed face images. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1822–1830, July 2017. Honolulu, HI. 2

[33] N. Rahmouni, V. Nozick, J. Yamagishi, and I. Echizen. Distinguishing computer graphics from natural images using convolution neural networks. *Proceedings of the IEEE Workshop on Information Forensics and Security*, pages 1–6, Dec. 2017. Rennes, France. 2

[34] A. Rössler et al. Faceforensics: A large-scale video dataset for forgery detection in human faces. *arXiv:1803.09179*, Mar. 2018. 2

[35] H. T. Sencar and N. Memon, editors. *Digital Image Forensics*. Springer New York, 2013. 2

[36] C. Szegedy et al. Rethinking the inception architecture for computer vision. *Proeedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, June 2016. Las Vegas, NV. 4

[37] A. Tewari et al. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 1274–1283, Oct. 2017. Venice, Italy. 1

[38] J. Thies et al. Face2Face: Real-time face capture and reenactment of rgb videos. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2387–2395, June 2016. Las Vegas, NV. 1, 2

[39] P. Upchurch et al. Deep feature interpolation for image content changes. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6090–6099, July 2017. Honolulu, HI. 2

[40] W. Wang and H. Farid. Exposing digital forgeries in interlaced and deinterlaced video. *IEEE Transactions on Information Forensics and Security*, 2(3), 2007. 2

[41] P. Zhou et al. Two-stream neural networks for tampered face detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1831–1839, July 2017. Honolulu, HI. 2

[42] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *Proceedings of the IEEE International Conference on Computer Vision*, pages 2242–2251, Oct. 2017. Venice, Italy. 1