MesoNet: a Compact Facial Video Forgery Detection Network

Darius Afchar
École des Ponts Paristech
Marne-la-Vallée, France
darius.afchar@live.fr

Vincent Nozick
JFLI, CNRS, UMI 3527, Japan
LIGM, UMR 8049,
UPEM, France
vincent.nozick@u-pem.fr

Junichi Yamagishi, Isao Echizen
National Institute of Informatics
Tokyo, Japan
jyamagis@nii.ac.jp,
iechizen@nii.ac.jp

Abstract

This paper presents a method to automatically and efficiently detect face tampering in videos, and particularly focuses on two recent techniques used to generate hyper-realistic forged videos: Deepfake and Face2Face. Traditional image forensics techniques are usually not well suited to videos due to the compression that strongly degrades the data. Thus, this paper follows a deep learning approach and presents two networks, both with a low number of layers to focus on the mesoscopic properties of images. We evaluate those fast networks on both an existing dataset and a dataset we have constituted from online videos. The tests demonstrate a very successful detection rate with more than 98% for Deepfake and 95% for Face2Face.

1. Introduction

Over the last decades, the popularization of smartphones and the growth of social networks have made digital images and videos very common digital objects. According to several reports, almost two billion pictures are uploaded everyday on the internet. This tremendous use of digital images has been followed by a rise of techniques to alter image contents, using editing software like Photoshop for instance. The field of digital image forensics research is dedicated to the detection of image forgeries in order to regulate the circulation of such falsified contents. There have been several approaches to detect image forgeries [8, 19], most of them either analyze inconsistencies relatively to what a normal camera pipeline would be or rely on the extraction of specific image alterations in the resulting image. Among others, image noise [11] has been shown to be a good indicator to detect splicing (copy-past from an image to another). The detection of image compression artifacts [2] also presents some precious hints about image manipulation.

Today, the danger of *fake news* is widely acknowledged and in a context where more than 100 million hours of video content are watched daily on social networks, the spread of falsified video raises more and more concerns. While significant improvements have been made for image forgery detection, digital video falsification detection still remains a difficult task. Indeed, most methods used with images can not be directly extended to videos, which is mainly due to the strong degradation of the frames after video compression. Current video forensic studies [16] mainly focus on the video re-encoding [28] and video recapture [29, 15], however video edition is still challenging to detect.

For the last years, deep learning methods has been successfully employed for digital image forensics. Amongst others, Barni et al. [2] use deep learning to locally detect double JPEG compression on images. Rao and Ni [18] propose a network to detect image splicing. Bayar and Stamm [3] target any image general falsification. Rahmouni et al. [17] distinguish computer graphics from photographic images. It clearly appears that deep learning performs very well in digital forensics, and disrupts traditional signal processing approaches.

In the other hand, deep learning can also be used to falsify videos. Recently, a powerful tool called *Deepfake* has been designed for face capture and reenactment. This methods, initially devoted to the creation of adult content, has not been presented in any academic publication. *Deepfake* follows *Face2Face* [26], a non deep learning method introduced by Thies et al. that targets similar goal, using more conventional real-time computer vision techniques.

This paper addresses the problem of detecting these two video editing processes, and is organized as follows: Sections 1.1 and 1.2 present more details on *Deepfake* and *Face2Face*, with a special attention for the first one that has not been published. In Section 2, we propose several deep learning networks to successfully overcome these two falsification methods. Section 3 presents a detailed evaluation of those networks, as well as the datasets we assembled for training and testing. Up to our knowledge, there is no other method dedicated to the detection of the *Deepfake* video falsification technique.

1.1. Deepfake

Deepfake is a technique which aims to replace the face of a targeted person by the face of someone else in a video. It first appeared in autumn 2017 as a script used to generate face-swapped adult contents. Afterwards, this technique was improved by a small community to notably create a user-friendly application called *FakeApp*.

The core idea lies in the parallel training of two auto-encoders. Their architecture can vary according to the output size, the desired training time, the expected quality and the available resources. Traditionally, an auto-encoder designates the chaining of an encoder network and a decoder network. The purpose of the encoder is to perform a dimension reduction by encoding the data from the input layer into a reduced number of variables. The goal of the decoder is then to use those variables to output an approximation of the original input. The optimization phase is done by comparing the input and its generated approximation and penalizing the difference between the two, typically using a L^2 distance. In the case of the *Deepfake* technique, the original auto-encoder is fed with images of resolution $64 \times 64 \times 3 = 12,288$ variables, encodes those images on 1024 variables and then generates images with the same size as the input.

The process to generate *Deepfake* images is to gather aligned faces of two different people $\mathbf{A}$ and $\mathbf{B}$, then to train an auto-encoder $\mathbf{E_A}$ to reconstruct the faces of $\mathbf{A}$ from the dataset of facial images of $\mathbf{A}$, and an auto-encoder $\mathbf{E_B}$ to reconstruct the faces of $\mathbf{B}$ from the dataset of facial images of $\mathbf{B}$. The trick consists in sharing the weights of the encoding part of the two auto-encoders $\mathbf{E_A}$ and $\mathbf{E_B}$, but keeping their respective decoder separated. Once the optimization is done, any image containing a face of $\mathbf{A}$ can be encoded through this shared encoder but decoded with decoder of $\mathbf{E_B}$. This principle is illustrated in Figure 1 and 2.

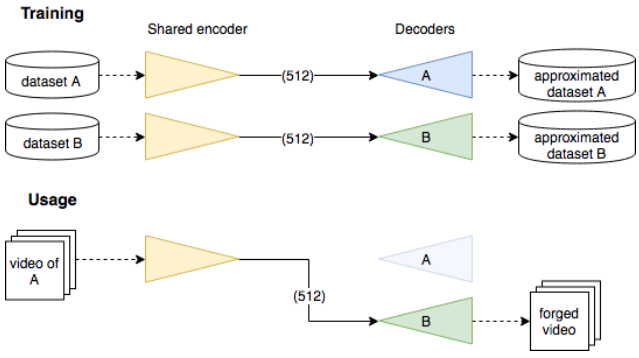

Figure 1. *Deepfake* principle. Top: the training parts with the shared encoder in yellow. Bottom: the usage part where images of $\mathbf{A}$ are decoded with the decoder of $\mathbf{B}$.

The intuition behind this approach is to have an encoder that privileges to encode general information of illumination, position and expression of the face and a dedicated decoder for each person to reconstitute constant characteristic shapes and details of the person face. This may thus separate the contextual information on one side and the morphological information on the other.

In practice, the results are impressive, which explains the popularity of the technique. The last step is to take the target video, extract and align the target face from each frame, use the modified auto-encoder to generate another face with the same illumination and expression, and then merge it back in the video.

Figure 2. Example of an image (left) being forged (right) using the *Deepfake* technique. Note that the forged face lacks the expressiveness of the original.

Fortunately, this technique is far from flawless. Basically, the extraction of faces and their reintegration can fail, especially in the case of face occlusions: some frames can end up with no facial reenactment or with a large blurred area or a doubled facial contour. However, those technical errors can easily be avoided with more advanced networks. More deeply, and this is true for other applications, auto-encoders tend to poorly reconstruct fine details because of the compression of the input data on a limited encoding space, the result thus often appears a bit blurry. A larger encoding space does not work properly since while the fine details are certainly better approximated, on the other hand, the resulting face loses realism as it tends to resemble the input face, i.e. morphological data are passed to the decoder, which is a undesired effect.

1.2. Face2Face

Reenactment methods, like [9], are designed to transfer image facial expression from a source to a target person. *Face2Face* [26], introduced by Thies et al., is its most advanced form. It performs a photorealistic and markerless facial reenactment in real-time from a simple RGB-camera, see Figure 3. The program first requires few minutes of pre-recorded videos of the target person for a training sequence to reconstruct its facial model. Then, at runtime, the pro-

gram tracks both the expressions of the source and target actors video. The final image synthesis is rendered by overlaying the target face with a morphed facial blendshape to fit the source facial expression.

Figure 3. Example of a *Face2Face* reenactment result from the demo video of the Face2Face paper [26].

2. Proposed method

This section presents several effective approaches to deal with either *Deepfake* or *Face2Face*. It turned out that these two problems can not be efficiently solved with a unique network. However, thanks to the similar nature of the falsifications, identical network structures for both problems can yield good results.

We propose to detect forged videos of faces by placing our method at a mesoscopic level of analysis. Indeed, microscopic analyses based on image noise cannot be applied in a compressed video context where the image noise is strongly degraded. Similarly, at a higher semantic level, human eye struggles to distinguish forged images [21], especially when the image depicts a human face [1, 7]. That is why we propose to adopt an intermediate approach using a deep neural network with a small number of layers.

The two following architectures have achieved the best classification scores among all our tests, with a low level of representation and a surprisingly low number of parameters. They are based on well-performing networks for image classification [14, 23] that alternate layers of convolutions and pooling for feature extraction and a dense network for classification. Their source code is available online[1].

2.1. Meso-4

We have started our experiments with rather complex architectures and have gradually simplified them, up to the following one that produces the same results but more efficiently.

This network begins with a sequence of four layers of successive convolutions and pooling, and is followed by a dense network with one hidden layer. To improve generalization, the convolutional layers use ReLU activation functions that introduce non-linearities and Batch Normalization [10] to regularize their output and prevent the *vanishing gradient* effect, and the fully-connected layers use Dropout [24] to regularize and improve their robustness.

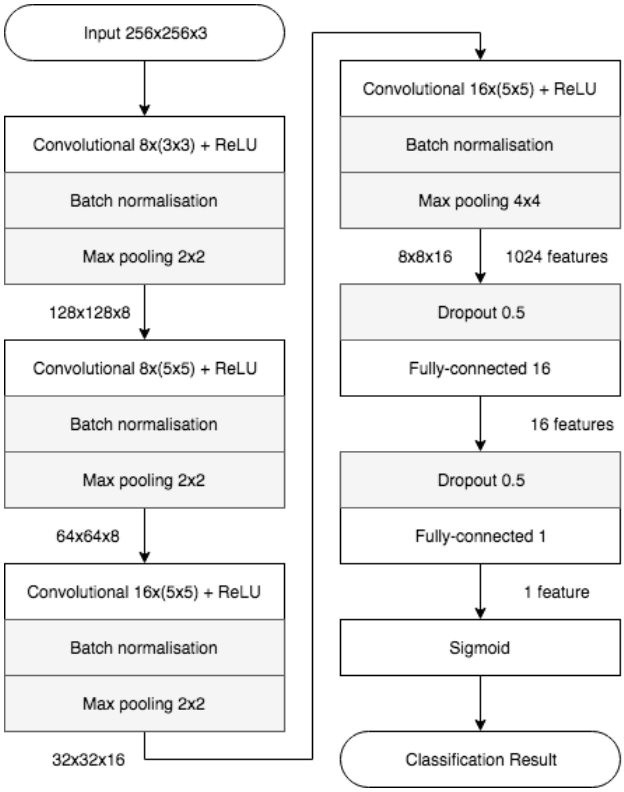

Figure 4. The network architecture of Meso-4. Layers and parameters are displayed in the boxes, output sizes next to the arrows.

In total, there are 27,977 trainable parameters for this network. Further details can be found on Figure 4.

2.2. MesoInception-4

An alternative structure consists in replacing the first two convolutional layers of Meso4 by a variant of the *inception module* introduced by Szegedy et al [25].

The idea of the module is to stack the output of several convolutional layers with different kernel shapes and thus increase the function space in which the model is optimized. Instead of the 5×5 convolutions of the original module, we propose to use 3×3 dilated convolutions [30] in order to avoid high semantic. This idea of using dilated convolutions with the inception module can be found in [22] as a mean to deal with multi-scale information, but we have added 1×1 convolutions before dilated convolutions for dimension reduction and an extra 1×1 convolution in parallel that acts as skip-connection between successive modules. Further details can be found in Figure 5.

Replacing more than two layers with *inception modules*

[1] https://github.com/DariusAf/MesoNet

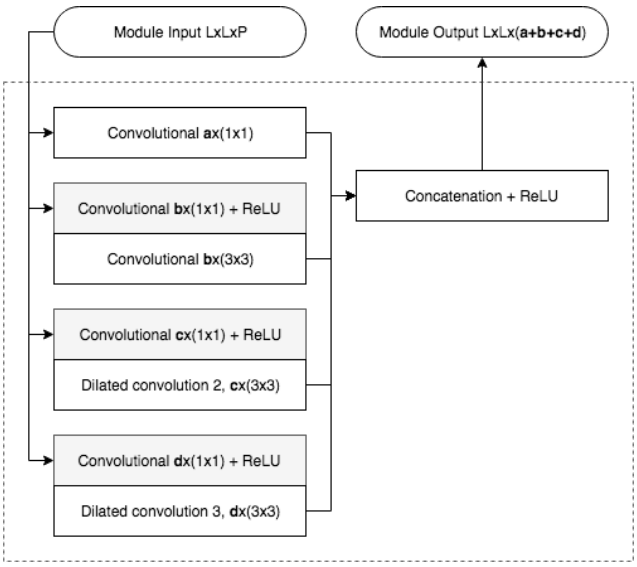

Figure 5. Architecture of the inception modules used in MesoInception-4. The module is parameterized using $a, b, c, d \in \mathbb{N}$. The dilated convolutions are computed without stride.

did not offer better results for the classification. The chosen parameters (a_i, b_i, c_i, d_i) for the module at layer i can be found in Table 1. With those hyper-parameters, this network has 28,615 trainable parameters overall.

Layer	a	b	c	d
1	1	4	4	1
2	1	4	4	2

Table 1. Hyper-parameters for the modified Inception modules used in MesoInception-4

3. Experiments

In this section we expose the results of the implementation of the two proposed architectures to detect the studied digital forgeries. In order to extend our work to the real case of online videos, we also discuss the robustness of our method to video compression.

3.1. Datasets

3.1.1 Deepfake dataset

To our knowledge, no dataset gathers videos generated by the *Deepfake* technique, so we have created our own.

Training auto-encoders for the forgery task requires several days of training with conventional processors to achieve realistic results and can only be done for two specific faces at a time. To have a sufficient variety of faces, we have rather chosen to download the profusion of videos available to the general public on the internet.

Thus, 175 rushes of forged videos have been collected from different platforms. Their duration ranges from two seconds to three minutes and have a minimum resolution of 854×480 pixels. All videos are compressed using the H.264 codec but with different compression levels, which puts us in real conditions of analysis. An accurate study on the effect of compression levels is conducted on another dataset introduced in Section 3.1.2.

All the faces have been extracted using the Viola-Jones detector [27] and aligned using a trained neural network for facial landmark detection [12]. In order to balance the distribution of faces, the number of selected frames for extraction per video is proportional to the number of camera angle and illumination changes on the target face. As a reference, approximately 50 faces were extracted per scene.

The dataset has then been doubled with real face images, also extracted from various internet sources and with the same resolutions. Finally, it has been manually reviewed to remove misalignment and wrong face detection. As much as possible, the same distribution of good resolution and poor resolution images were used in both classes to avoid bias in the classification task.

Precise numbers of the image count in each classes as long as the separation into a set used for training and for model evaluation can be found in Table 2.

3.1.2 Face2Face dataset

Additionally to the *Deepfake* dataset, we have examined whether the proposed architecture could be used to detect other face forgeries. As a good candidate, the FaceForensics dataset [20] contains over a thousand forged videos and their original using the *Face2Face* approach. This dataset is already split into a training, validation and testing set.

More than extending the use of the proposed architecture to another classification task, one advantage of the FaceForensics set is to provide losslessly compressed videos, which has enabled us to evaluate the robustness of our model with different compression levels. To be able to compare our results with those from the FaceForensics paper [20], we have chosen the same compression rate with H.264: lossless compression, 23 (light compression), 40 (strong compression).

Only 300 videos were used for training out of more than a thousand. For the model evaluation, the 150 forged video and their original of the testing set were used. Details about the number of extracted face images for each class can be found in Table 2.

3.2. Classification Setup

We denote $\mathcal{X}$ the input set and $\mathcal{Y}$ the output set, the random variable pair (X, Y) taking values in $\mathcal{X} \times \mathcal{Y}$, and f the prediction function of the chosen classifier that takes

Set	**forged** class	**real** class
Deepfake training	5111	7250
Deepfake testing	2889	4259
Face2Face training	4500	4500
Face2Face testing	3000	3000

Table 2. Cardinality of each class in the studied datasets. Note that for both datasets, 10% of the training set was used during the optimization for model validation.

values in $\mathcal{X}$ to the action set $\mathcal{A}$. The chosen classification task is to minimize the error $\mathcal{E}(f) = \mathbb{E}[l(f(X), Y)]$, with $l(a, y) = \frac{1}{2}(a - y)^2$.

Both networks have been implemented with Python 3.5 using the Keras 2.1.5 module [5]. Weights optimization of the network is achieved with successive batches of 75 images of size $256 \times 256 \times 3$ using ADAM [13] with default parameters ($\beta_1 = 0.9$ and $\beta_2 = 0.999$). The initial learning rate of 10^{-3} is divided by 10 every 1000 iterations down to 10^{-6}. To improve generalization and robustness, input batches underwent several slight random transformations including zoom, rotation, horizontal flips, brightness and hue changes.

As both network have a relatively small amount of parameters, few hours of optimization on a standard consumer grade computer were enough to obtain good scores.

3.3. Image classification results

Classification scores of both trained network are shown in Table 3 for the *Deepfake* dataset. Both networks have reached fairly similar score around 90% considering each frame independently. We do not expect a higher score since the dataset contains some facial images extracted with a very low resolution.

Network	Deepfake classification score		
Class	forged	real	total
Meso-4	0.882	0.901	0.891
MesoInception-4	0.934	0.900	0.917

Table 3. Classification scores of several networks on the *Deepfake* dataset, considering each frame independently.

Table 4 presents results for the *Face2Face* forgery detection. We observed a notable deterioration of scores at the strong video compression level. The paper that introduces the FaceForensics dataset used in our tests [20] presents better classification results using the state-of-the-art network for image classification Xception [4]. However, with the configuration given by the latter paper, we only managed to fine-tune Xception up to obtain a 96.1% score at the compression level 0 and 93.5% score at level 23. It is therefore unclear how to interpret the results.

Network	Face2Face classification score		
Compression level	0	23 (light)	40 (strong)
Meso-4	0.946	0.924	0.832
MesoInception-4	0.968	0.934	0.813

Table 4. Classification scores of several networks on the *FaceForensics* dataset, considering each frame independently.

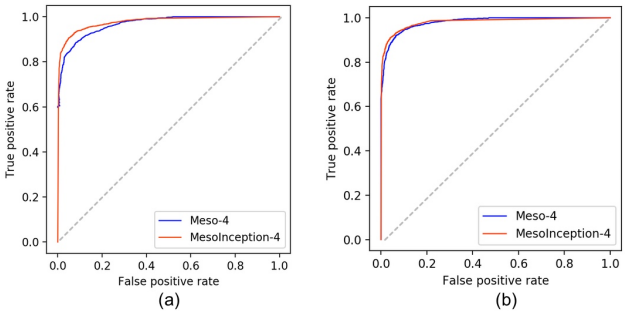

Figure 6. ROC curves of the evaluated classifiers on the *Deepfake* dataset (a) and the *Face2Face* dataset compressed at rate 23 (b).

3.4. Image aggregation

One downside of video analysis, especially online videos, is the compression which causes a huge loss of information. But on the other hand, having a succession of frames of the same face makes it possible to multiply experiences and might help obtain a more accurate overall score on the video. A natural way of doing so is to average the network prediction over the video. Theoretically speaking, there is no justification for a gain in scores or a confidence interval indicator as frames of a same video are strongly correlated to one another. In practice, for the viewer comfort, most filmed face contain a majority of stable clear frames. The effect of punctual movement blur, face occlusion and random misprediction can thus be outweighted by a majority of good predictions on a sample of frames taken from the video.

Experimental results can be found in Table 5. The image aggregation significantly improved both detection rate. It even soared higher than **98%** with the MesoInception-4 network on the *Deepfake* dataset. Note that on the *Face2Face* dataset, the same score is reached for both networks but the misclassified videos are different.

Network	Aggregation score	
Dataset	Deepfake	Face2Face (23)
Meso-4	**0.969**	**0.953**
MesoInception-4	**0.984**	**0.953**

Table 5. Video classification scores on the two dataset using image aggregation, with the *Face2Face* dataset compressed at rate 23.

3.5. Aggregation on intra-frames

To extend our study of the effect of video compression on forgery detection, we have conducted the same image aggregation but only with intra-frames of compressed videos, i.e. frames that are not interpolated over time, to see if the reduced amount of compression artifacts would help increase the classification score. The flip side is that videos only lasting a few second may contain as little as three I-frames, which cancels out the expected smoothing effect of the aggregation.

We found out that it rather had a bad effect on the classification, however the different is slight, as shown in Table 6. It might be used as a quick aggregation since the resulting scores are higher than a single image classification.

Network	I-Aggregation score	Difference
Meso-4	0.932	-0.037
MesoInception-4	0.959	-0.025

Table 6. Classification score variation on the *Deepfake* dataset using only I-frames.

3.6. Intuition behind the network

We have tried to understand how those networks solve the classification problem. This can be done by interpreting weights of the different convolutional kernel and neurons as image descriptors. For instance, a sequence of a positive weight, a negative one, then a positive one again, can be interpreted as a discrete second order derivation. However, this is only relevant for the first layer and does not tell much in the case of faces.

Another way is to generate an input image that maximizes the activation of a specific filter [6] to see what kind of signal it is reacting to. Concisely, if we note f_{ij} the output of filter j of layer i and x a random image, and we add a regularization on the input to reduce noise, that idea boils down to the maximization of $E(x) = f_{ij}(x) - \lambda\|x\|_p$.

Figure 7 shows such maximum activation for several neurons of the last hidden layer of Meso4. We can separate those neurons according to the sign of the weight applied to their output for the final classification decision, thus accounting for whether their activation pushes toward a negative score, which corresponds to the forged class, or a positive one for the real class. Strikingly, positive-weighted neurons activation display images with highly detailed eyes, nose and mouth areas while negative-weighted ones display strong details on the background part, leaving a smooth face area. That's understandable as *Deepfake*-generated faces tend to be blurry, or at least to lack details, compared to the rest of the image that was left untouched.

We can also take the mean output of a layer for batches of real and forged images, observe the differences of activation and interpret the parts of the input images that play a key role in the classification. If we study the trained MesoInception-4 network on the *deepfake* dataset, as it can be seen in Figure 8, eyes are strongly activated for real images but not on *deepfake* images for which the background shows the highest peaks. We can surmise that it is again a question of blurriness: the eyes being the most detailed part of real images while it's the background in forged images because of the dimension reduction underwent by the face.

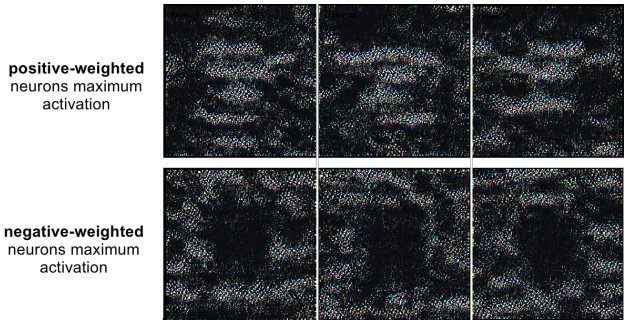

Figure 7. Maximum activation of some neurons of the hidden layer of Meso4. The optimization was done with $\lambda = 10$ and $p = 6$.

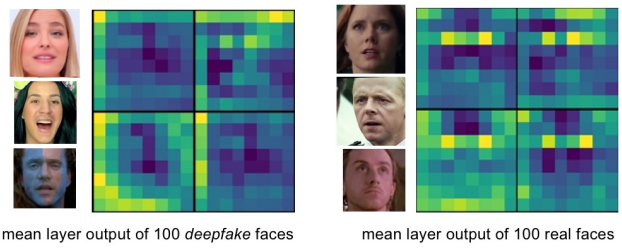

Figure 8. Mean outputs of the *Deepfake* dataset for the some filters of the last convolutional layer of MesoInception-4.

4. Conclusion

These days, the dangers of face tampering in video are widely recognized. We provide two possible network architectures to detect such forgeries efficiently and with a low computational cost. In addition, we give access to a dataset devoted to the *Deepfake* approach, a very popular yet underdocumented topic to our knowledge. Our experiments show that our method has an average detection rate of 98% for *Deepfake* videos and 95% for *Face2Face* videos under real conditions of diffusion on the internet.

One fundamental aspect of deep learning is to be able to generate a solution to a given problem without the need of a prior theoretical study. However, it is vital to be able to understand the origin of this solution in order to evaluate its qualities and limitations, which is why we spent a significant time visualizing the layers and filters of our networks.

We have notably understood that the eyes and mouth play a paramount role in the detection of faces forged with *Deepfake*. We believe that more tools will emerge in the future toward an even better understanding of deep networks to create more effective and efficient ones.

References

[1] B. Balas and C. Tonsager. Face animacy is not all in the eyes: Evidence from contrast chimeras. *Perception*, 43(5):355–367, 2014. 3

[2] M. Barni, L. Bondi, N. Bonettini, P. Bestagini, A. Costanzo, M. Maggini, B. Tondi, and S. Tubaro. Aligned and non-aligned double jpeg detection using convolutional neural networks. *Journal of Visual Communication and Image Representation*, 49:153–163, 2017. 1

[3] B. Bayar and M. 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. ACM, 2016. 1

[4] F. Chollet. Xception: Deep learning with depthwise separable convolutions. *arXiv preprint*, pages 1610–02357, 2017. 5

[5] F. Chollet et al. Keras. https://keras.io, 2015. 5

[6] D. Erhan, Y. Bengio, A. Courville, and P. Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1, 2009. 6

[7] S. Fan, R. Wang, T.-T. Ng, C. Y.-C. Tan, J. S. Herberg, and B. L. Koenig. Human perception of visual realism for photo and computer-generated face images. *ACM Transactions on Applied Perception (TAP)*, 11(2):7, 2014. 3

[8] H. Farid. A Survey Of Image Forgery Detection. *IEEE Signal Processing Magazine*, 26(2):26–25, 2009. 1

[9] P. Garrido, L. Valgaerts, O. Rehmsen, T. Thormahlen, P. Perez, and C. Theobalt. Automatic face reenactment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4217–4224, 2014. 2

[10] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 3

[11] T. Julliand, V. Nozick, and H. Talbot. Image noise and digital image forensics. In Y.-Q. Shi, J. H. Kim, F. Pérez-González, and I. Echizen, editors, *Digital-Forensics and Watermarking: 14th International Workshop (IWDW 2015)*, volume 9569, pages 3–17, Tokyo, Japan, October 2015. 1

[12] D. E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009. 4

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

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 3

[15] J.-W. Lee, M.-J. Lee, T.-W. Oh, S.-J. Ryu, and H.-K. Lee. Screenshot identification using combing artifact from interlaced video. In *Proceedings of the 12th ACM workshop on Multimedia and security*, pages 49–54. ACM, 2010. 1

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

[17] N. Rahmouni, V. Nozick, J. Yamagishi, and I. Echizen. Distinguishing computer graphics from natural images using convolution neural networks. In *IEEE Workshop on Information Forensics and Security, WIFS 2017*, Rennes, France, December 2017. 1

[18] Y. Rao and J. Ni. A deep learning approach to detection of splicing and copy-move forgeries in images. In *Information Forensics and Security (WIFS), 2016 IEEE International Workshop on*, pages 1–6. IEEE, 2016. 1

[19] J. A. Redi, W. Taktak, and J.-L. Dugelay. Digital image forensics: a booklet for beginners. *Multimedia Tools and Applications*, 51(1):133–162, 2011. 1

[20] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner. Faceforensics: A large-scale video dataset for forgery detection in human faces. *arXiv preprint arXiv:1803.09179*, 2018. 4, 5

[21] V. Schetinger, M. M. Oliveira, R. da Silva, and T. J. Carvalho. Humans are easily fooled by digital images. *arXiv preprint arXiv:1509.05301*, 2015. 3

[22] W. Shi, F. Jiang, and D. Zhao. Single image super-resolution with dilated convolution based multi-scale information learning inception module. *arXiv preprint arXiv:1707.07128*, 2017. 3

[23] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014. 3

[25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, et al. Going deeper with convolutions. Cvpr, 2015. 3

[26] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. 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. 1, 2, 3

[27] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2001. 4

[28] W. Wang and H. Farid. Exposing digital forgeries in video by detecting double mpeg compression. In *Proceedings of the 8th workshop on Multimedia and security*, pages 37–47. ACM, 2006. 1

[29] W. Wang and H. Farid. Detecting re-projected video. In *International Workshop on Information Hiding*, pages 72–86. Springer, 2008. 1

[30] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 3