Deepfake Video Detection with Facial Features and Long-Short Term Memory Deep Networks

Dan-Cristian Stanciu, Bogdan Ionescu
AI Multimedia Lab, Politehnica University of Bucharest, Romania
Email: {dan.stanciu1203, bogdan.ionescu}@upb.ro

***Abstract*— Generative models have evolved immensely in the last few years. GAN-based video and image generation has become very accessible due to open source software available to anyone, and that may pose a threat to society. Deepfakes can be used to intimidate, blackmail certain public figures or to mislead the public. At the same time, with the rising popularity of deepfakes, detection algorithms have also evolved significantly. The majority of those algorithms focus on images rather than to explore the temporal evolution in the video. In this paper, we explore whether the temporal information of the video can be used to increase the performance of state-of-the-art deepfake detection algorithms. We also investigate whether certain facial regions contain more information about the authenticity of the video by using the entire aligned face as input for our model and by only selecting certain facial regions. We use late fusion to combine those results for increased performance. To validate our solution, we experiment on 2 state-of-the-art datasets, namely FaceForensics++ [1] and CelebDF [2]. The results show that using the temporal dimension can greatly enhance the performance of a deep learning model.**

Index terms: **deepfake, deep learning, digital video forensics, face manipulation, facial regions, LSTM**

I. Introduction

Fake videos generated by deep learning, or deepfakes, emerged in the years following the introduction of Generative Adversarial Networks (GAN) [3], as generated images, and therefore videos became more realistic and in some cases, almost indistinguishable from real ones. Nowadays, anyone can use software found online, such as FaceApp [4], to create falsified videos. This may pose a threat to society, due to the fact that deepfakes can be used to mislead individuals, impersonate or blackmail people or to defame celebrities.

The methods used to create deepfakes are very diverse and range from changing expressions, editing features such as hair color, age, ethnicity, to more dangerous attacks: face swapping, expression reenactment or mouth movements "dubbing" using a target audio clip. GAN models are trained on very large datasets and use a discriminator to ensure that the generated images are hard to distinguish from benign samples.

With the rise in popularity of deepfakes came an increased interest of the machine learning community to expose them. Several datasets such as FaceForensics++ [1], CelebDF [2], DFDC [5] were created by the community with the purpose of helping researchers find the best approaches to detect deepfakes. The latter was proposed by Facebook in December, 2019 in the Deepfake Detection Challenge [6], a competition with the goal to aid deepfake detection research.

The reminder of the article is organized as following: Section II overviews the existing literature and highlights our contribution beyond the state of the art. Section III contains information about out implementation: the used datasets, the preprocessing algorithm and out model's architecture. Finally, Section IV presents our experimental results and a comparison with the state of the art.

II. Related work

Multimedia Manipulation. There are multiple ways to manipulate multimedia content. One of the most simple ways is using an image editing software package, which can enable users to delete elements from pictures, insert objects by copying them from a different image or to manipulate certain image characteristics like brightness, color distribution or size. These manipulations, sometimes called "cheap fakes" do not require deep learning methods but can be very effective in some cases. Fortunately, there are many methods to detect content that has been manipulated by using software editing tools. As an example, Error Level Analysis [7] is an algorithm that can detect inserted objects in images by using lossy compression algorithms like JPEG to find compression artefacts in images.

Multimedia content generation has been a challenge for the deep learning community for a long time. In recent years, Generative Adversarial Networks have been the most used and most accurate way to generate data. As an example, the website thispersondoesnotexist.com [8] has popularized the deep learning image generation systems and has shown that their performance is already incredible.

Deepfakes are defined to be videos of people that have been digitally altered using deep learning techniques. Many of those techniques originate from the idea of Generative Adversarial Networks [3], first introduced in 2014 by Goodfellow *et al*. Starting from architectures like CycleGAN [9], Face2Face [10], or StarGAN [11] which use style transfer to reenact a video using another person's face, software like Faceapp [4], DeepFaceLab [12], FaceSwap-GAN [13] and many others made deepfake generation a fairly simple task.

Multimedia Forensics and Datasets. With the evolution of deepfakes and their increasing accessibility online, came a need to detect them, in order to protect the public against misinformation or certain people against blackmail or defamation. Multimedia forensics aims to protect against any kind of manipulation of digital content and ensure its integrity. For detection of cheapfakes, state-of-the-art methods proved

to be very effective. On the other side, deepfakes were introduced recently and while there are some good results in their detection, there are still some problems to be solved like: dealing with low resolution videos, dealing with compression or generalization. Due to interest in deepfakes detection rising, a few datasets were created for training and evaluating forensics algorithms. There are a few datasets containing generated images, but the most important ones contain videos created using identity swap/expression reenactment algorithms. Since the creation of the first publicly available datasets, there have been several improvements in the video quality and realism of the fake videos. The majority of datasets contain short length ($<$15 sec) videos of people who are usually speaking while being fairly stationary. In literature, there are 2 generations of deepfake datasets currently available.

The first generation of datasets includes DeepfakeTIMIT (2018) [14] which contains 620 deepfakes generated using FaceSwap-GAN [13] and FaceForensics++ (2019) [1], containing 1,000 real Youtube videos and 2 sets of 1,000 images each created with a non-GAN computer graphics software [15] and a public DeepFake FaceSwap software [16]. The DeepFake FaceSwap software [16] is based on two autoencoders with a shared encoder and recreates the facial expression of an input image with the looks of a target image.

The second generation of datasets brought some improvements to the techniques used to generate videos and increased the number of videos per dataset. Celeb-DF (2019) [2] is a datset containing 5,639 deepfakes and 890 real videos. The deepfakes are obtained by "faceswapping" videos of celebrities starting from the real videos. The biggest and most recent dataset is the DeepFake Detection Challenge Dataset (2020) [5] from Facebook. It was used in a competition [6] and contains over 124,000 videos of consenting individuals in indoor, outdoor settings and different light levels.

Deepfake Detection Approaches. There are several ideas presented in the state of the art, with many focusing on image-level solutions. A few important methods for deepfake classifiers are the ones proposed by the authors in [1], [17], which are based on CNN classifiers, an algorithm using Capsule Networks [18], an approach based on detecting face warping artefacts [19] and some papers which use RNN-based architectures to take advantage of the temporal dimension in the video [20], [21], [22]. An interesting approach was presented by the authors of [23], who proposed selecting facial regions like mouth, eyes, nose and the rest of the face from the original image and using them to train CNNs.

In this paper we use a facial landmark detection algorithm to detect and crop different facial regions. We use the evolution in time of those facial regions to train a CNN-LSTM architecture with the purpose of binary deepfake detection. We will compare the results from different face regions with the results obtained from feeding the whole face as an input. We use late fusion to combine the probabilities resulted from each face region.

Fig. 1. Preprocessing steps: (1) Face detection, (2) extraction of facial landmarks with OpenFace2 [26], (3) extraction and alignment of face and (4) facial regions of interest (eyes, nose, mouth).

III. Proposed Method

We present the implementation details of our algorithm, from the datasets used, the facial landmark detection and how we select the needed facial regions, and including the architecture for our model using Xception [24], Long-Short Term Memory (LSTM) [25] and the late fusion approach for combining outputs generated from facial regions.

Proprocessing. For cropping the face region and identifying facial landmark points, we used the open source software OpenFace2 [26]. The algorithm extracts 68 facial landmarks and saves their coordinates. Using the landmark points, we extracted 3 different regions of the face: mouth, eyes and nose. We also used the whole face for comparison and to determine if the extraction of certain facial regions improves performance. For every 5^{th} frame, facial regions were extracted, the face was aligned and cropped using a mask and the image was resized to 299×299. The diagram of the proposed approach is presented in Fig. 1. It consists of the following processing blocks: (1) face detection, (2) facial landmark mapping, (3) face extraction and alignment, (4) facial regions extraction for eyes, nose and mouth, using the landmark points.

CNN-LSTM Architecture. The model used in this paper combines a Convolutional Neural Network which handles the spatial dimension and extracts feature vectors, and a LSTM [25] which receives the feature vectors as input and handles the temporal dimension. LSTM architectures are known for handling sequences well and have been used in computer vision tasks like activity recognition, image and video description or person re-identification.

The LSTM model needs an "encoder" prior to handling image sequences. For this task we used an Xception [24] network, due to the fact that it is one of the go-to architectures for image classification tasks. Xception is inspired by the Inception V3 [27] architecture, but Inception modules are exchanged for depthwise separable convolutions. Xception has proven to be successful for deepfake detection tasks in recent papers like [1], [5], [23], [28]. The Xception model is pretrained on the ImageNet dataset [29]. We froze the parameters from start to the 4^{th} block and we fine-tuned the

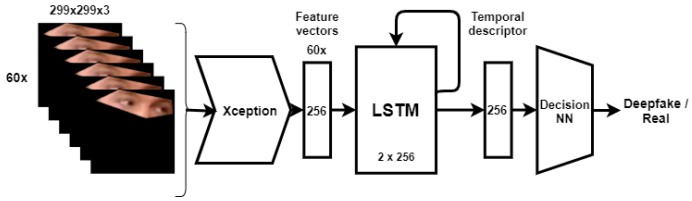

Fig. 2. CNN-LSTM architecture. Xception network will output 60 feature vectors for a temporal sequence of 60 images. The 2 layer LSTM [25] will output a temporal descriptor which is fed to the decision fully connected layers to obtain the deepfake probability.

rest of the model. The last fully connected layer has been replaced by another two layers which will also be fine-tuned during training. The resulting output feature vector has a length of 256 and is an input for the LSTM.

We use a 2-layer LSTM network, with 256 hidden units per layer, which will get a feature vector resulted from the Xception as an input, for every 5$^{\text{th}}$ frame of the video. The LSTM will receive a total of 60 frames from the video, which is equivalent to a 10 second sequence at 6 frames per second. Zero padding was used on videos shorter than 10 seconds. Only 60 frames were used because [30] showed that the accuracy difference between 20 and 80 frames is less that 1%. The LSTM output will go into a fully connected layer with a dropout probability of 0.7. We use a final layer with a Sigmoid activation function to output the probability for the video being deepfake or pristine. Binary crossentropy is used as a loss function.

Fig. 2 illustrates the model architecture presented in this section and it consists of the following blocks: (1) input block for extracted region (in this case, eyes): a sequence of 60 RGB images of 299×299 resolution, (2) the Xception network, used to extract a feature vector for every image in the sequence, (3) the LSTM block: a 2 layer LSTM of size 256 which outputs a temporal descriptor for the sequence, and (4) the decision fully connected layers.

After training the models, we save the final outputs for the training datasets and use them to train a late fusion algorithm. It combines the resulted deepfake probabilities for all 3 face regions (mouth, eyes, nose) using a simple deep neural network, predicting a single number which is the fusion deepfake probability.

IV. Experimental Results

Datasets and evaluation metrics. We used the CelebDF dataset [2] - 6,529 videos and the FaceForensics++ dataset [1] - 2,000 videos, for training and validation. We use 80% of the data for training purposes and 20% for evaluation. We decided to run our experiments on both datasets due to the fact that the videos are created using different methods: FaceForensics is a 1^{st} generation deepfake dataset, while CelebDF is 2^{nd} generation. The facial region image sequences are obtained using the preprocessing techniques described in Section III. Because the CelebDF dataset has 7 times more deepfakes compared to real images, we used benign samples 3 times more often in training to balance the dataset.

TABLE I

Analysis of the influence of different face regions.

Face Region	Study	CelebDF *AUC[%]*	FF++ *AUC[%]*
Full Face	Proposed	97.06	99.95
	Tolosana *et al [23]*	83.60	99.40
Mouth	Proposed	84.29	98.15
	Tolosana *et al [23]*	65.10	93.90
Nose	Proposed	75.60	95.35
	Tolosana *et al [23]*	64.90	86.30
Eyes	Proposed	85.81	98.64
	Tolosana *et al [23]*	77.30	92.70
Late Fusion	Proposed	86.09	98.46

The main evaluation metric used for this paper is the AUC-ROC score, which will be referred to as AUC (Area Under Curve) for the rest of this paper. Because this is a binary classification problem, accuracy should not be used due to the fact that the results are dependant on a chosen threshold. The AUC-ROC curve plots True Positive vs False Positive at different thresholds, thus measuring a classifier's ability to distinguish between classes without needing a fixed threshold.

Training and parameters. For each dataset, we have trained 4 different models, one for every selected facial region and one for the entire face. The models were trained for 20 epochs at most, using early stopping when needed. Finally, we used late fusion to try to improve the performance, based on our model's outputs for the 3 face regions. For both datasets, we used 5×10^{-4} as the value for the learning rate, with a 0.92 learning rate decay factor and a value of 10^{-5} for weight decay. We used the Adam optimizer and Binary Cross-Entropy as a loss function.

Results. Table 1 contains a comparison between our method and the approach of Tolosana *et al* [23], which used extracted facial regions in a similar fashion. The table presents the results of the Xception Network applied in [23], compared to our approach that uses an Xception network combined with LSTM. As we can see from the results, individual facial regions do not yield better performance versus the full face image. What is more, although the Late Fusion of the 3 facial region could provide slightly better AUC, it is still not comparable to using the full face region. In spite of that, we can see that they still yield great performance and can even be used separately to detect deepfakes, especially in cases where we can suspect that only certain parts of the face were attacked. Finally, we can conclude that using the temporal dimension with LSTM benefits the performance greatly, as the performance of our Xception-LSTM model is significantly better than the performance of an Xception network alone.

In Table 2 we can observe a comparison between the some of the state-of-the-art methods used in the deepfake detection task. The FaceForensics++ dataset is a fairly easy one, as we can see by the fact that all results have an AUC close to 100%.

TABLE II

COMPARISON WITH DIFFERENT STATE-OF-THE-ART APPROACHES.

Study	Method	CelebDF *AUC[%]*	FF++ *AUC[%]*
Li *et al. (2019) [19]*	Face Warping Features + CNN	64.60	93.00
Rössler *et al. (2019) [1]*	Mesoscopic Features + Steganalysis Features + CNN	-	99.26
Nguyen *et al. (2019) [18]*	Capsule Networks	57.50	96.60
Sabir *et al. (2019) [20]*	CNN + RNN	-	96.9
Dang *et al (2019) [28]*	CNN + Attention Map	71.2	-
Tolosana *et al (2020) [23]*	Facial Regions Features CNN	83.6	99.4
Proposed	CNN + LSTM	**97.06**	**99.95**

For this dataset, out method added a 0.55% improvement in AUC.

On the other hand, the 2nd generation CelebDF can bring some difficulties. In spite of that, our method still yields a 97.06% AUC (13.46% increase over similar state-of-the-art approaches), proving that using temporal features with LSTM can greatly improve performance.

V. CONCLUSION

In this paper we proposed using a spatio-temporal CNN-LSTM approach for deepfake detection using 3 selected facial regions. We compared the performance of the model run on those regions separately to a late fusion combination and to the model using entire face, on 2 state-of-the-art datasets. We can conclude that using a temporal network helps improve performance in deepfake detection, as our method yields a 13.46% increase in AUC for the CelebDF dataset (from 83.6% to 97.06%), and an almost perfect 99.95% AUC for the FaceForensics++ dataset. For future improvements, we plan to add an attention mechanism, test this method on other datasets and to bring improvements to generalization and dealing with compression.

ACKNOWLEDGMENT

This work was supported under project AI4Media, A European Excellence Centre for Media, Society and Democracy, H2020 ICT-48-2020, grant #951911.

REFERENCES

[1] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, Matthias Nießner, "FaceForensics++: Learning to Detect Manipulated Facial Images," in 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), 2019 pp. 1-11.

[2] Y. Li, X. Yang, P. Sun, H. Qi and S. Lyu, "Celeb-DF: A Large-Scale Challenging Dataset for DeepFake Forensics," in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020 pp. 3204-3213.

[3] Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A. Bengio, "Generative Adversarial Networks", 2014.

[4] Faceapp, https://www.faceapp.com, (Accessed on 16/04/2021).

[5] Brian Dolhansky, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes, Menglin Wang, Cristian Canton Ferrer, "The DeepFake Detection Challenge (DFDC) Dataset", 2020.

[6] DeepFake Detection Challenge, https://www.kaggle.com/c/deepfake-detection-challenge, (Accessed on 16/04/2021).

[7] Wang, W.; Dong, J.; Tan, T., "Tampered Region Localization of Digital Color Images". Digital Watermarking: 9th International Workshop, IWDW 2010. Seoul, Korea: Springer. pp. 120–133.

[8] This Person Does Not Exist, https://thispersondoesnotexist.com, (Accessed on 16/04/2021).

[9] J. Zhu, T. Park, P. Isola, A. Efros, "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," in 2017 IEEE International Conference on Computer Vision, Venice, Italy, pp. 2242-2251.

[10] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt and M. Niessner, "Face2Face: Real-Time Face Capture and Reenactment of RGB Videos," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016 pp. 2387-2395.

[11] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, Jaegul Choo, "StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation", 2018.

[12] Ivan Perov, Daiheng Gao, Nikolay Chervoniy, Kunlin Liu, Sugasa Marangonda, Chris Umé, Mr. Dpfks, Carl Shift Facenheim, Luis RP, Jian Jiang, Sheng Zhang, Pingyu Wu, Bo Zhou, Weiming Zhang, "DeepFaceLab: A simple, flexible and extensible face swapping framework", 2020.

[13] faceswap-GAN github. https://github.com/shaoanlu/faceswap-GAN, (Accessed on 16/04/2021).

[14] DeepFake TIMIT Dataset, https://www.idiap.ch/dataset/deepfaketimit, (Accessed on 16/04/2021).

[15] FaceSwap , https://github.com/MarekKowalski/FaceSwap, (Accessed on 16/04/2021).

[16] DeepFake FaceSwap, https://github.com/deepfakes/faceswap, (Accessed on 16/04/2021).

[17] B. Dolhansky, R. Howes, B. Pflaum, N. Baram, and C. Ferrer, "The Deepfake Detection Challenge (DFDC) Preview Dataset", 2019.

[18] H.H. Nguyen, J. Yamagishi and I. Echizen, "Use of a Capsule Network to Detect Fake Images and Videos", 2019.

[19] Y. Li and S. Lyu, "Exposing DeepFake Videos By Detecting Face Warping Artifacts," in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2019.

[20] E. Sabir, J. Cheng, A. Jaiswal, W. AbdAlmageed, I. Masi, and P. Natarajan, "Recurrent Convolutional Strategies for Face Manipulation Detection in Videos," in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2019.

[21] D. Güera and E. J. Delp, "Deepfake Video Detection Using Recurrent Neural Networks," 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Auckland, New Zealand, 2018, pp. 1-6.

[22] Shahroz Tariq, Sangyup Lee, Simon S. Woo, "A Convolutional LSTM based Residual Network for Deepfake Video Detection", 2020.

[23] Ruben Tolosana, Sergio Romero-Tapiador, Julian Fierrez and Ruben Vera-Rodriguez, "DeepFakes Evolution: Analysis of Facial Regions and Fake Detection Performance", 2020.

[24] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017 pp. 1800-1807.

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

[26] T. Baltrusaitis, A. Zadeh, Y. Lim, and L. Morency, "OpenFace 2.0: Facial Behavior Analysis Toolkit," in Proc. International Conference on Automatic Face and Gesture Recognition, 2018.

[27] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016 pp. 2818-2826.

[28] H. Dang, F. Liu, J. Stehouwer, X. Liu, and A. Jain, "On the Detection of Digital Face Manipulation," in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020.

[29] J. Deng, W. Dong, R. Socher, L. Li, Kai Li and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 2009, pp. 248-255

[30] D. Guera and E. Delp, "Deepfake Video Detection Using Recurrent Neural Networks," in 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Auckland, New Zealand, pp. 1-6.