Improving Deepfake Detection by Mixing Top Solutions of the DFDC

Anis Trabelsi
Digital Security Department
EURECOM
Biot, France
anis.trabelsi@eurecom.fr

Marc Michel Pic
Digital Labs
SURYS
France
m.pic@surys.com

Jean-Luc Dugelay
Digital Security Department
EURECOM
Biot, France
jean-luc.dugelay@eurecom.fr

***Abstract*—The falsification of faces in videos is a growing phenomenon over the years. One of the most popular ways to tamper a face in a video is known as "deepfake". Today, many tools exist to allow anyone to create a deepfake to discredit an individual or usurp an identity. Fortunately, the detection of deepfakes is an increasing topic of interest for the scientific community. As a result, many efforts have been made to develop mechanisms to automatically identify deepfake videos. In addition, several public deepfakes datasets have been built to help researchers to develop more effective detection methods. The most recent and also the most complete of these datasets is the one built by Facebook as part of the international DeepFake Detection Challenge (DFDC). Thousands of different frameworks, mainly based on deep learning, have been proposed during this challenge. The best solution that has been proposed obtains the accuracy of 82% on the DFDC dataset. However, the accuracy of this method is only 65% on unseen videos from the Internet. In this paper we analyse the five best methods of the DFDC and their complementarity. In addition, we experimented different assembly strategies (boosting, bagging and stacking) among these solutions. We show that we can achieve a large improvement (+41% on log loss and +2.26% on accuracy) when we carefully choose the models to be assembled with the most appropriate right merging method to use.**

***Index Terms*—deepfake detection, deepfake detection challenge, ensembling**

I. Introduction

The manipulation of faces in a digital image is not a new problem. There are many efficient methods to detect all types of falsification, such as morphing and face swapping. However, new methods of tampering have emerged over the last few years. These new methods can be applied to videos and the detection of these forgeries is much more difficult. The most popular technology is called deepfakes.

A deepfake is a method that allows exchanging in a fast, automatic and realistic way a face in a video. Since the appearance of the first deepfake video at the end of 2017, this technology has become more and more realistic and it is now very difficult for a human to identify a deepfake video [13].

As a result, deepfakes can be used as a tool to spread fake news by falsifying videos and sharing them over the internet. Figure 1 shows an example of a deepfake. The face in the original video (left) has been replaced by another face (right).

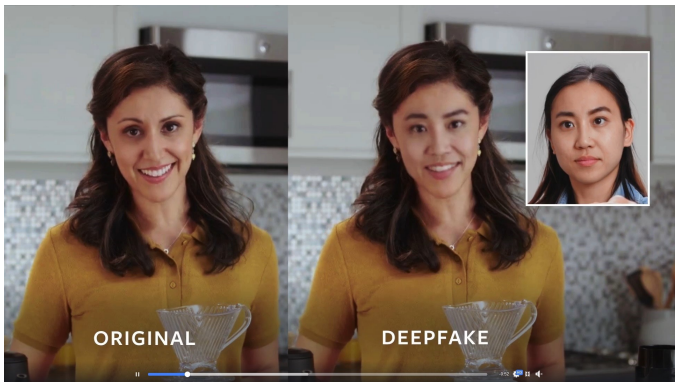

Fig. 1. An example of deepfake video.

Deepfakes are also a threat in the field of biometrics. Indeed, it is nowadays accepted to use a biometric feature as a mean of authentication and the face is the most widely used biometric feature. Many facial recognition authentication systems require a video of the user's face as biometric evidence. The deepfake technology can then be used as an attack in order to fool a system. Apart from the high degree of realism of a deepfake video, the greatest danger is that no special technical skills are required to produce this kind of forgery. It is not necessary to master complicated software either. Today, anyone can make a deepfake. For all these reasons, it is essential to combat deepfakes by developing powerful detection methods that automatically evaluate the authenticity of a video and spot deepfake.

Between late 2019 and early 2020, a worldwide competition called DeepFake Detection Challenge (DFDC) was held. The objective of this competition was to obtain generalizable methods to detect deepfakes. Thousands of participants proposed their own methods. Despite promising results, no method has been able to detect all types of deepfakes in the wild.

In this paper we propose an analysis of the winning solutions of the deepfakes detection challenge. In particular, we study the assembling of these solutions and the complementarity between them. We test different ensembles with various strategies to merge the scores and we show that a well-chosen assembly can significantly improve the results.

Dataset	Year	# Fake videos	# Real videos	# Identities	# Methods	# Augmentation
UADFV [23]	2018	49	49	49	1	-
DeepfakeTIMIT [12]	2018	640	320	43	2	-
FaceForensics++ [21]	2019	4000	1000	?	4	2
Google DFD [18]	2019	3000	363	28	**5**	-
Celeb-DFD [16]	2019	5639	890	59	1	-
DeeperForensics-1.0 [9]	2020	1000	**59000**	100	1	7
DFDC [6]	2020	**104500**	23654	**960**	**5**	**19**

TABLE I
LIST OF ALL THE DIFFERENT EXISTING DEEPFAKES DATASETS

II. PREVIOUS WORKS

The deepfake could be considered as a research topic in itself. It is a topic that is widely studied in the literature. Each year, the number of articles dealing with deepfakes is increasing. This ranges from the general study of deepfakes and the issues raised, to the development of deepfakes detection methods and the publication of databases dedicated to deepfakes. Numerous articles also propose new methods for making deepfakes.

A. Methods for creating deepfake

Most of the methods to generate a deepfake are based on two types of neural networks : auto-encoders and Generative Adversarial Networks (GANs).

A deepfake based on auto encoders consists in using two auto encoders and crossing the decoders. An auto-encoder is a type of neural network used to reconstruct an image from compressed information (called latent space) of the same image.

A deepfake based on GANs is made of two distinct parts, a generator G and a discriminator D. In the case of deepfake generation, the role of the generator is to synthesize a video capable of deceiving the discriminator and the role of the discriminator is to determine whether the content proposed by the generator is authentic or not. Many variants of deepfake generation based on GANs have been developed: FSGAN [19], StyleGAN [11], PGGAN [10].

In the beginning, it was very resource-intensive to produce a realistic deepfake. Today this is no longer the case. The general public can create deepfakes with limited effort thanks to easy-to-use applications. The most popular of this application is FaceApp [2], but more and more other applications are being released every year (DeepFaceLab [20], ZAO [4], Faceswap web [3], etc.).

B. Deepfake detection methods

Considering the many threats involved by deepfakes, many detection methods have been proposed. In the literature, there are mainly three categories of deepfake detection methods: based on physiological analysis, based on images texture analysis and based on automatic detection with artificial intelligence. As part of the physiological analysis, Li et al. [14] observed some inconsistencies in the eye blinking in a deepfake video. Using a Long-term Recurrent Convolutional Network (LCRN) they successfully detected deepfake videos. In [23], the authors determine whether a video is deepfake by analyzing inconsistencies in head position. For detection methods based on image or texture analysis, the authors mainly look for inconsistencies in optical flow [5] or the presence of artifacts [15]. Finally, approaches purely based on a detection using artificial intelligence passes the frames of a video through neural networks. The neural networks can be recurrent neural networks [7], 3D convolutional networks [22], or ensemble of them.

Unfortunately, and because of the significant diversity of the different ways to generate a deepfake, it is very difficult to develop a method suitable to detect all deepfake videos. It is also important to consider the models must be robust to adversarial attacks. Indeed, in [17], it has been shown that it is possible to easily deceive a detector by injecting an adverse noise into a video of them. To face all these problems, more and more diverse and rich database of deepfakes are being made available.

C. Existing deepfake datasets

To the best of our knowledge, we count seven large datasets of deepfakes (Table I). We can evaluate the "quality" of a dataset according to the number of deepfake videos, the number of original videos, the number of distinct identities, the number of methods used to create a deepfake and the number of augmentations that are applied.

The dataset that most closely matches these criteria is the dataset made by Facebook for the DeepFake Detection Challenge.

III. DEEPFAKE DETECTION CHALLENGE

The Deepfake Detection Challenge (DFDC) [6] is an international competition, launched in December 2019 by Facebook, in collaboration with Microsoft, Amazon Web Services (AWS), and some academic partners as well. A database of over 100,000 videos of real and deepfake video has been made available to participants. Facebook used different techniques to modify the face of the actor presented in the videos. The final results were published on 12 June 2020. A total of 2114 teams from all over the world participated.

The best solution was proposed by Selim Seferbekov. He extracts 32 frames of a video, detects the face and crops it and then feeds the faces present in these frames in an architecture composed of an ensemble of seven EfficientNet-B7. During the learning step, he uses an augmentation strategy by removing semantic parts of the face.

Results [6]	Final rank [6]	Re-implementation (Log-Loss)	Re-implementation (Accuracy %)	New rank
0.1983	**1st**	0.1957	92.68	4th
0.1787	2nd	**0.1790**	93.36	**1st**
0.1703	3rd	0.1821	**93.90**	2nd
0.1882	4th	0.1863	92.58	3rd
0.2157	5th	0.2158	90.86	5th

TABLE II
SCORE OF THE WINNING SOLUTIONS ON THE FACEBOOK PUBLIC TEST SET

	1st	2nd	3rd	4th	5th
1st	100%	48%	41%	48%	26%
2nd		100%	53%	47%	**22%**
3rd			100%	49%	27%
4th				100%	27%
5th					100%

TABLE III
PERCENTAGE OF FALSE POSITIVES AND FALSE NEGATIVES IN COMMON BETWEEN WINNING SOLUTIONS

The team that finished in second position, proposed a set of two XCeption and one EfficientNet-B3. They also used a special data augmentation strategy called WS-DAN [8].

The third best model proposed an ensemble of three EfficientNet-B7. During the training stage, it uses the mixup data augmentation strategy. The fourth team proposed a large ensemble of different CNNs (EfficientNet-B0, EfficientNet-B1, EfficientNet-B3, ResNet-34, Xception and SlowFast).

Finally, the fifth solution proposes an architecture composed of an ensemble of 2D and 3D CNNs. They apply the cutmix data augmentation strategy during the training stage.

All the methods we have just described use a simple fusion strategy by applying weights on the predictions of each of the models they use in their ensemble. For evaluating the submissions of each participant during this competition, the organizers used the log-loss function (1).

$$LogLoss = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)] \quad (1)$$

Where n is the number of videos to predict, $\hat{y}_i$ represents the probability of the video being fake, and y_i is the label of the video, 0 for a real video and 1 for a fake video. This metric is used to evaluate the predictions returned by the submitted models. A wrong prediction with a high confidence will be highly penalised.

According to Facebook [6], the log-loss of the best model is 0.4279 (which corresponds here to an accuracy of 82%) on the test set made available by Facebook. Unfortunately, when evaluating the same model with video taken from the internet, the accuracy dropped by more than 15%, and the same model achieved an accuracy of 65%.

IV. EXPERIMENTS

In this section, we will perform several experiments. First, we will determine if the top-five solutions can complement each other by determining the number of false positives and false negatives they have in common. In order to do so, we have reimplemented the 5 best solutions of the challenge. These five methods were made open source and the trained models were shared at the end of the competition. We have rerun each solution on the public test set that Facebook has made available online composed of 5000 videos (Table II).

This test set represents only 50% of the total test set that has been used to determine the final leaderboard during the DFDC. In fact, the complete test set consists of 10,000 videos. Half of the videos are deepfakes made by Facebook and original footage, the other half is composed of deepfakes and original videos retrieved over the Internet. The videos retrieved from the internet were not made public by the organizers. Considering this point, we only focus on the 5,000 public videos built by Facebook. This is why the scores presented in the table II do not correspond exactly to the final DFDC ranking. In the rest of the paper we refer to the new ranking we obtain after re-implementation.

A. False positives and false negatives in common

The different solutions have an error rate between 7% and 10% (Table II), which corresponds to false positive and false negative. In order to determine the level of similarity between each solution, we calculate the percentage of false positives and false negatives in common between two solutions (Table III). We can observe that there are far fewest false positives and false negatives in common between the fifth solution and the others. From this observation, it can be presumed that it is not the same elements that make the decision of method #5 compared to the other methods. This can be explained by the particular architecture of the fifth solution composed mainly of 3D-CNN contrary to the other methods which have mainly used 2D-CNN. In view of this observation, we have tested all different ensembles of winning solutions to determine if the fifth method can be complementary when used with one of the other methods and thus improve the results.

B. Strategies for merging scores

In the literature, making ensemble is a concept that can significantly improve results if the models do not converge to the same predictions. The downside is that by making ensemble we add complexity and it becomes more difficult to understand the decisions.

Assembly using a voting strategy is one of the simplest methods. Two types of voting classifier exist. **Majority voting**: it is a vote on class prediction. In our experiments we also managed the case of a tie between the models. In case of a tie, we take into consideration the prediction score. **Weight**

Type of ensemble	Best score (Log-loss)	Best score (Accuracy %)	Fusion strategies	Models
Single	0.1790	93.90	-	1st
Ensemble by 2	**0.1376**	95.24	Majority vote	1st + 5th
Ensemble by 3	0.1605	**95.82**	Majority vote	1st + 2nd + 5th
Ensemble by 4	0.1583	95.64	Majority vote	1st + 2nd + 4th + 5th
Ensemble by 5	0.1597	95.42	Majority vote	all

TABLE IV
SCORE OF THE WINNING SOLUTIONS ON THE FACEBOOK PUBLIC TEST SET

Ensemble	LR Best score	SVM Best score	RF Best score	AdaBoost Best score	MLP Best score
Ensemble by 2	0.1231 | 95.36	0.1268 | 94.96	0.1888 | 94.48	0.4680 | **95.52**	**0.1221** | 95.36
Ensemble by 3	**0.1063** | 95.52	0.1092 | 95.60	0.1160 | 95.60	0.4608 | **95.68**	0.1076 | 95.52
Ensemble by 4	**0.1056** | **96.16**	0.1080 | 95.92	0.1096 | **96.16**	0.4763 | 95.20	0.1065 | 96.00
Ensemble by 5	**0.1049** | 95.76	0.1072 | 95.52	0.1233 | **95.92**	0.5842 | 90.40	0.1056 | 95.84

TABLE V
RESULTS OF THE BEST ENSEMBLE ON EACH OF THE ASSEMBLY STRATEGIES ON THE FACEBOOK PUBLIC TEST SET (LOG LOSS | ACCURACY %)

voting: We define weights according to the importance of the rank of each model (0.3 for the 1st, 0.25 for the 2nd, 0.2 for the 3rd, 0.15 for the 4th, and 0.1 for the 5th).

More sophisticated methods are based on machine learning: **Bagging**: train several sub-models on random portions of data from the training dataset (*e.g.*random forest). **Boosting**: train several models one after the other, each model corrects the errors of its predecessor (*e.g.*adaboost). **Stacking**: train a model to predict a final score from the predictions of each of the models (*e.g.*voting ensemble model).

C. Deep ensemble experiments results

We have realized several ensembles composed of two, three, four and all models among the winning solutions using the strategies described in the last subsection. We started by using a voting strategy as these are the simplest methods. Our database is composed of the predictions of 5000 videos of each of the models. The results of the best ensemble for these experiments are presented in the table IV ("Ensemble by n means an assembly using n models among the models).

It is always the majority voting fusion strategy that gives the best results on different types of ensembles. By combining the first and fifth solution we manage to improve the log loss by 23% and the accuracy by 1.34% compared to the best single solution. This combination is composed of the best solution (1st) and the solution that had the least false positives and false negatives in common with the others (5th), which verifies our preliminary hypothesis. Adding more models to the ensembling will degrade the log-loss compared to the best ensemble of two models.

We then evaluated five machine learning algorithms as an assembly method on all possible combinations between the five models. These algorithms are : Logistic Regression (LR), Support Vector Machine (SVM), Random Forest (RF), Adaptive Boosting (AdaBoost) and Multi-layer Perceptron (MLP). These algorithms are implemented in the popular open source Python library called "scikit-learn". We have used this library to apply these algorithms with the default parameters.

We train each of these algorithms with the predictions of the top 5 solutions. We have split our dataset into 75% for training and 25% for testing. The results of each algorithm are reported in table V. From these results, we can make several observations.

Firstly, the results improve significantly when moving from a single solution to an ensemble of two models. A similar observation can be made when moving from an ensemble of two methods to an ensemble of three methods. Then, the improvement is limited.

Secondly, all these merger strategies are better than fusion by vote (except for RF). The best ensemble is obtained with the MLP strategy with a log-loss of 0.1221 (which corresponds to an improvement of 31.78% compared to the best single solution).

Thirdly, mixing all models can improve results by over 41% compared to the best single model. However, we observe that assembling only three models can already improve the log-loss by 40% and therefore adding more models is not relevant given the trade-off between increased complexity and performance gain. In terms of accuracy, we observe an improvement for each fusion strategy on each of the ensembles. With the LR fusion strategy, the accuracy is improved by 2.26% when assembling the 1st + 3rd + 4th + 5th models. Combining all the models is not necessary, as the accuracy for each of the fusion strategies decreases compared to assembling four models.

Finally, AdaBoost is the only strategy that degrades performance for every type of ensemble. With this strategy the best ensemble is the one composed of the 1st + 2nd + 5th solution with a log-loss of only 0.4608.

D. Test on unseen dataset

We also tested the different proposed ensembles on an external dataset with unseen deepfakes videos (video generated by an algorithm not used in the training phase). This dataset is part of [21]. The distribution of real/fake videos in this dataset is unbalanced, there are many more deepfakes than original videos. We decided to randomly select 1000 altered videos and 1000 original videos to be able to evaluate it more fairly.

As it is reported in [6], the results of the models trained on the DFDC challenge dataset drop on unseen videos. This

Ensemble	LR Best score	SVM Best score	RF Best score	AdaBoost Best score	MLP Best score
Ensemble by 2	**0.3540** | 81.92	**0.3540** | 82.56	0.4752 | 82.08	0.5865 | **83.36**	0.5865 | 82.56
Ensemble by 3	0.3489 | 82.24	0.3526 | 82.56	**0.3380** | **84.16**	0.5729 | 83.36	0.3472 | 83.36
Ensemble by 4	0.3446 | 82.40	0.3502 | 82.40	0.3736 | **84.48**	0.5763 | 84.00	**0.3303** | 84.00

TABLE VI
RESULTS OF THE BEST ENSEMBLE ON EACH OF THE ASSEMBLY STRATEGIES ON THE EXTERNAL DATASET (LOG LOSS| ACCURACY %)

Final rank [6]	Log-Loss	Accuracy (%)	New rank
1st	0.6802	75.24	4th
2nd	0.4028	81.04	2nd
3rd	0.7035	70.60	5th
4th	**0.1527**	**92.04**	**1st**
5th	0.5335	76.56	3rd

TABLE VII
SCORE OF THE WINNING SOLUTIONS ON THE EXTERNAL DATASET

is the case for all models except for the 4th model. To avoid including a potential bias in the ensembling we did not include this model in our experiments. In the remainder of the paper, we use the new ranks from table VII to indicate the models which are used. All the results are presented in table VI. The combination which improves the most the log loss (by 21%) and the accuracy (by 3.44%) is the one combining all four models.

V. CONCLUSIONS AND FUTURE WORK

In this paper we proposed a study of the winning solutions of the DFDC. We have demonstrated that it is possible to improve results by making appropriate deep ensembles. In this way it is possible to improve the log loss by 41% and the accuracy by 2.26% on the public test set from the DFDC. On an external dataset, it is possible to improve the log loss by 21% and the accuracy by 3.44%.

We have observed that the best methods make thoughtful use of data augmentations in the training stage. However, this is still not enough to make these methods generalisable. Indeed, when using a different dataset composed of deepfake videos not seen during training, the results are not as good.

Interpretability and explanability are areas of great interest in AI [1] and not only for the problem of deepfake detection. We believe that these are the areas we really need to work on now.

We believe that in order to solve a problem such as the generalization of deepfake detection methods, it is necessary to understand the predictions of the models and investigate the complementarity of architectures used.

REFERENCES

[1] Chist-era. 2020. xaiface: Measuring and improving explainability for ai-based face recognition. https://www.chistera.eu/projects/xaiface. Accessed: 2022-03-06.

[2] Faceapp. https://www.faceapp.com/. Accessed: 2022-03-06.

[3] Faceswap web. https://faceswapweb.com/. Accessed: 2022-03-06.

[4] Zao. https://apps.apple.com/cn/app/zao. Accessed: 2022-03-06.

[5] Irene Amerini, Leonardo Galteri, Roberto Caldelli, and A. Bimbo. Deepfake video detection through optical flow based cnn. *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 1205–1207, 2019.

[6] Brian Dolhansky, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes, Menglin Wang, and Cristian Canton-Ferrer. The deepfake detection challenge dataset. *ArXiv*, abs/2006.07397, 2020.

[7] David Guera and Edward J. Delp. Deepfake video detection using recurrent neural networks. *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6, 2018.

[8] Tao Hu and Honggang Qi. See better before looking closer: Weakly supervised data augmentation network for fine-grained visual classification. *ArXiv*, abs/1901.09891, 2019.

[9] Liming Jiang, Wayne Wu, Ren Li, Chen Qian, and Chen Change Loy. Deeperforensics-1.0: A large-scale dataset for real-world face forgery detection. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2886–2895, 2020.

[10] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *ArXiv*, abs/1710.10196, 2018.

[11] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4396–4405, 2019.

[12] Pavel Korshunov and Sébastien Marcel. Deepfakes: a new threat to face recognition? assessment and detection. *ArXiv*, abs/1812.08685, 2018.

[13] Pavel Korshunov and Sébastien Marcel. Deepfake detection: humans vs. machines. *ArXiv*, abs/2009.03155, 2020.

[14] Yuezun Li, Ming-Ching Chang, and Siwei Lyu. In ictu oculi: Exposing ai created fake videos by detecting eye blinking. *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–7, 2018.

[15] Yuezun Li and Siwei Lyu. Exposing deepfake videos by detecting face warping artifacts. *ArXiv*, abs/1811.00656, 2019.

[16] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-df: A large-scale challenging dataset for deepfake forensics. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3204–3213, 2020.

[17] Paarth Neekhara, Brian Dolhansky, Joanna Bitton, and Cristian Canton-Ferrer. Adversarial threats to deepfake detection: A practical perspective. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 923–932, 2021.

[18] Google Research Nick Dufour and Jigsaw Andrew Gully. Contributing data to deepfake detection research. https://ai.googleblog.com/2019/09/contributing-data-to-deepfake-detection.html. Accessed: 2022-03-06.

[19] Yuval Nirkin, Yosi Keller, and Tal Hassner. Fsgan: Subject agnostic face swapping and reenactment. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7183–7192, 2019.

[20] Ivan Petrov, Daiheng Gao, Nikolay Chervoniy, Kunlin Liu, Sugasa Marangonda, Chris Umé, Mr. Dpfks, RP Luis, Jian Jiang, Sheng Zhang, Pingyu Wu, Bo Zhou, and Weiming Zhang. Deepfacelab: A simple, flexible and extensible face swapping framework. *ArXiv*, abs/2005.05535, 2020.

[21] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1–11, 2019.

[22] Yaohui Wang and Antitza Dantcheva. A video is worth more than 1000 lies. comparing 3dcnn approaches for detecting deepfakes. *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, pages 515–519, 2020.

[23] Xin Yang, Yuezun Li, and Siwei Lyu. Exposing deep fakes using inconsistent head poses. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8261–8265, 2019.