CUSTOM ATTRIBUTION LOSS FOR IMPROVING GENERALIZATION AND INTERPRETABILITY OF DEEPFAKE DETECTION

Pavel Korshunov[1]*, Anubhav Jain*[1]*, and Sébastien Marcel*[1 2]

[1]Idiap Research Institute, Martigny, Switzerland
[2]Université de Lausanne, Switzerland

{pavel.korshunov,anubhav.jain,sebastien.marcel}@idiap.ch

ABSTRACT

The simplicity and accessibility of tools for generating deepfakes pose a significant technical challenge for their detection and filtering. Many of the recently proposed methods for deeptake detection focus on a 'blackbox' approach and therefore suffer from the lack of any additional information about the nature of fake videos beyond the fake or not fake labels. In this paper, we approach deepfake detection by solving the related problem of attribution, where the goal is to distinguish each separate type of a deepfake attack. We design a training approach with customized Triplet and ArcFace losses that allow to improve the accuracy of deepfake detection on several publicly available datasets, including Google and Jigsaw, FaceForensics++, HifiFace, DeeperForensics, Celeb-DF, DeepfakeTIMIT, and DF-Mobio. Using an example of Xception net as an underlying architecture, we also demonstrate that when trained for attribution, the model can be used as a tool to analyze the deepfake space and to compare it with the space of original videos.

Index Terms— Deepfake attribution, deepfake detection, cross-database evaluations, ArcFace loss, Triplet loss

1. INTRODUCTION

Recent advances in automated video and audio editing tools, generative adversarial networks (GANs), and social media allow creation and fast dissemination of high quality tampered videos, which are generally called deepfakes. Typically, in these videos, a face is swapped with someone else's using GANs. Accessible open source software and apps for the face swapping led to a wide and rapid dissemination of the generated deepfakes, posing a significant technical challenge for their detection and analysis.

Many databases with deepfake videos were created to help develop and train deepfake detection methods. One of the first freely available database was DeepfakeTIMIT [1], followed by the FaceForensics database with deepfakes generated from 1000 Youtube videos [2], and which was later morphed into FaceForensics++ with more types of deepfakes and a separate set of original and deepfake videos provided by Google and Jigsaw [3]. Several independent extensions of FaceForensics++ were also proposed, including HifiFace [4] and DeeperForensics [5] datasets (see examples of different deepfakes from FaceForensics++ and its extensions in Figure 1). Another 5000 videos-large database of deepfakes generated from Youtube videos is Celeb-DF v2 [6]. But the most extensive and the largest database to date with more than $100K$ videos (80% of which are deepfakes) is the dataset from Facebook [7], which was available for download to the participants in the recent Deepfake Detection Challenge hosted by Kaggle[1].

Many methods for deepfake detection were proposed recently [3, 8, 9, 10, 11], however, as it is typical for deep learning-based approaches, they suffer from the lack of generalization on different types of generative models, video blending techniques used in deepfakes, and data unseen during training [12, 13]. This issue was clear at Facebook's Deepfake Detection Challenge[2] when the top approaches of the competition have shown a much lower error on the public validation set, compared to the error on the secret test set, which contained unseen data and deepfakes generated using undisclosed methods. This lack of generalization and also the lack of understanding the space of deepfake attacks and the differences between them are impeding the advances in deepfake detection and their wide employment.

In this paper, we address two problems of the generalization of deepfake detection and lack of their understanding by solving a single problem of attack attribution. The goal of an attribution approach is not just to distinguish real videos from all deepfakes but to assign a different label to each type of deepfake seen during training. This approach, of course, is only possible when the dataset provides information on which deepfake generation method was used for which video, which, for instance, Facebook or Google datasets do not provide. Once the model is trained for attribution, it can be used in two-fold manner: i) as a binary classifier during test time to simply distinguish deepfakes and real videos and ii) as a

The work is funded by Swiss center for biometrics research and testing.

[1]https://www.kaggle.com/c/deepfake-detection-challenge

(a) Original (b) Neural Text. (c) Deepfakes (d) Face Shifter (e) Face2Face (f) Face Swap (g) HifiFace (h) DF End2end (i) DF Reenact

Fig. 1: Cropped faces from videos of FaceForensics++ [3], HifiFace [4], and DeeperForensics (DF) [5] databases.

descriptor of the test dataset, since by attributing all test deepfake videos we can label them as being of a specific deepfake type. The hypothesis is that the attribution approach would allow us, regardless of the underlying model, to estimate the space of deepfake attacks better than a binary classifier. Then, the trained model can become a versatile tool for both well-generalizable deepfake detection tool and for an analysis of the deepfake space.

We have previously shown [14] that using a training-for-attribution approach works very well for deepfake detection, especially on unseen databases. In this paper, we extend that work by using more data for training attribution-based models, by proposing to use a custom loss aiming to improve detection and interpretability, and by using such models for analyzing the embedding space of deepfakes. Please note that our aim is not to show that an attribution-based approach works, for that, please refer to our previous work [14], but to extend it further.

Therefore, in this paper, we focus solely on Xception net [15] as an underlying model that we train for attribution using different approaches. Any other model can be used, but we use Xception due its popularity and because it is fast enough to train. We adjusted the model and the training process to perform an attribution for ten classes of real videos and various deepfakes from FaceForensics++ [2] (it has five types of deepfakes), Celeb-DF [6], HifiFace [4], and DeeperForensics [5] (two types of deepfakes) databases. We propose to use three different approaches to model the deepfake attacks space: i) by training with a simple dense layer of the size ten added at the top of the baseline model, referred to simply as *Attribution*, ii) by training using triplet loss [16], referred to as *TripleLoss*, and by iii) training using a modified ArcFace loss [17], referred to as *ArcFaceMod*. We compare these approaches to a baseline binary classifier trained on the same datasets and the same underlying model.

We evaluate the proposed approaches (Binary, Attribution, TripletLoss, and ArcFaceMod) by training on the combination of train subsets from FaceForensics++, Celeb-DF, HifiFace, and DeeperForensics and testing on three other datasets: DeepfakeTIMIT [1], Google and Jigsaw [3], and DF-Mobio [14].

Since we are unable to fit all the results and plots in the paper, we invite the reader to explore the evaluations in Jupyter notebook as part of our open-source Python package with reproducible experiments[2].

2. ATTRIBUTION APPROACHES

We consider three different techniques to train models for deepfake attribution: i) a simple attribution when a dense layer the size of the number of classes is added to the model, ii) an approach to train a model as a siamese network with triplet-loss [16], and (iii) an approach where we modified an ArcFace [17] loss to better model the space of deepfakes. We compare these approaches with a simple Binary classifier.

To keep the evaluations comparable, we use the same underlying Xception model [15] with weights pre-trained on ImageNet [18], the same preprocessing steps and training parameters. As input, we use 224×224 faces cropped with Blazeface face detector for 15 frames per video. We train for 20 epochs, with the best performing model selected based on the validation loss, on batches of size 16 with Adam optimizer and a learning rate of 0.0002.

The differences between training approaches lie in the top layers and losses. For Binary classifier, we use a single neuron for classification and a cross entropy loss, for Attribution, instead of a single neuron we use a layer of size ten, which is equal to the number of different classes, with the same loss, for TripletLoss [16], we use embedding layer of size 64 and a triple loss with semi-hard triplets, and for ArcFaceMod, we use the same 64 sized embedding layer but with a modified ArcFace [17] loss.

We train all the approaches on a combination of FaceForensics++ and its extensions with Celeb-DF, which amounts to nine different deepfake classes plus one for real videos (see face examples from FaceForensics++ and its extensions in Figure 1). We combine training sets of all these datasets to form a large training 'super-set'. Then, we test the trained models on each of these datasets to see how the models perform in the same-database scenario. Then, test them on the entire Google, DF-Mobio, and DeepfakeTIMIT databases to test how well the models generalize in a cross-database scenario. We also show how different attribution approaches model the space of deepfakes of different datasets.

[2] https://gitlab.idiap.ch/bob/bob.paper.deepfake_attribution/-/tree/icassp2022

For more details on Binary, Attribution, and TripletLoss based approaches, please refer to [14], where these approaches are evaluated in a similar settings, except that a smaller subset (without DeeperForensics and HifiFace deepfakes) of training data was used.

2.1. ArcFaceMod loss

To better model a space of deepfakes, we propose using a modified additive angular margin loss (ArcFace) [17], which led to high accuracies in face recognition benchmarks. The idea behind both triplet and angular loss approaches is similar: to minimize the distance between the sample of the same class and maximize the distance between samples of different classes. However, adapting an angular loss to the specifics of deepfake detection is more intuitive. In face recognition and many other classification problems, all classes are considered equal, however, in deepfake detection, this is not the case, since we have a distinct different class of real original videos. Hence, we would like a trained model to treat real class differently from all deepfake classes. And it is easier to adapt the angular distance metric to this situation compared to the Euclidean used in TripletLoss. Therefore, we propose changing the value of margin (see m in the equation (3) of [17]) depending on the type of training sample. To further separate real class in the embedding space from all the fake classes, we increase margin value from a recommended 0.5 to 0.6 for real samples and decrease it to 0.4 for all the deepfakes.

Similar to TripletLoss (see more details in [14]), we convert an embedding space into ten classes by training another set of classifiers: k-nearest neighbor (NN), logistic regression (LR), and support vector machine (SVM). We trained these classifiers using the embeddings extracted from the validation set of our combined 'super' dataset and the hyper parameters were tuned using grid search on the 'super' test set. Number $k = 14$ was found to be the best for k-nearest neighbor, 0.001 as the regularization parameter for logistic regression, and 0.01 as the regularization parameter for SVM.

2.2. Evaluation methodology

We evaluate all approaches, using several metrics: an area under the curve (AUC), which is a popular metric used in the literature on deepfake detection [3, 6, 19], false positive rate (FPR), false negative rate (FNR), and half total error rate (HTER), which are the commonly used metrics for evaluation of classification systems that rely on a detection threshold. We define the threshold θ_{fpr} to correspond to the FPR value of 10% on a validation set, which means 10% of fake videos are allowed to be misclassified as real. Using this threshold on the scores of the test sets will result in the test FPR and FNR values. Hence, HTER, defined as $HTER(\theta_{fpr}) = \frac{FPR_{test}+FNR_{test}}{2}$ can be used as a single value metric to compare all approaches.

Table 1: Same-database evaluation of deepfake detection.

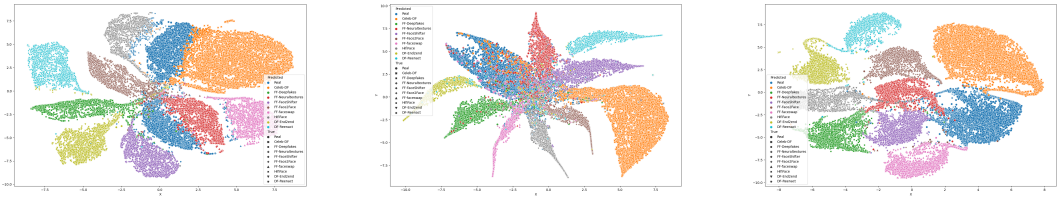
Approach	Test DB	AUC	FNR (%)	FPR (%)	HTER (%)
Binary	Celeb-DF	98.56	1.12	26.76	13.94
	FaceForensics++	48.03	95.00	4.20	49.60
	DeeperForensics	54.62	95.00	3.57	49.29
	HifiFace	25.29	95.00	8.57	51.79
Attribution	Celeb-DF	**100.00**	1.69	0.00	**0.84**
	FaceForensics++	**99.14**	0.71	10.36	5.54
	DeeperForensics	99.93	0.71	2.50	**1.61**
	HifiFace	96.57	0.71	35.00	17.86
TripletLoss-NN	Celeb-DF	99.76	0.56	10.88	5.72
	FaceForensics++	98.84	1.43	9.20	**5.31**
	DeeperForensics	**99.78**	1.43	2.86	2.14
	HifiFace	98.63	1.43	15.00	**8.21**
ArcFaceMod-LR	Celeb-DF	99.87	0.00	5.59	2.79
	FaceForensics++	99.06	1.43	11.34	6.38
	DeeperForensics	99.95	1.43	2.50	1.96
	HifiFace	**98.81**	1.43	16.43	8.93

3. EXPERIMENTAL RESULTS

Our same- and cross-database experiments are aimed to be as comparable as possible. We trained all detection approaches using the same training 'super-set' formed by combining Celeb-DF and FaceForensics++ with its extensions. We evaluated the approaches on the test sets of Celeb-DF, FaceForensics++, HifiFace, or DeeperForensics in the same-dataset scenario and on the entirely unseen DeepfakeTIMIT, Google (from Google and Jigsaw), and DF-Mobio datasets in the cross-database scenario. The results published in [14] can be considered as an ablation study, since it used a smaller subset of training data.

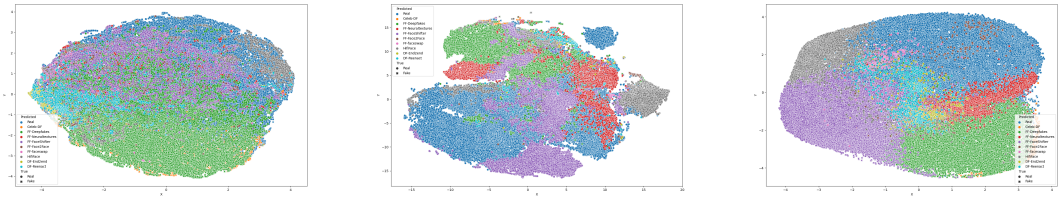
Table 1 shows the same-database scenario results for the baseline binary classifier, the classifier trained for an attribution task, a triplet-loss classifier, and a modified ArcFace-loss classifier. We report only NN-based results for TripletLoss and LR-based for ArcFaceMod because of the limited space in the paper and because k-nearest neighbor classifier led to the best results for TripletLoss, while logistic regression was the best for ArcFaceMod (for a complete set of results please refer to the open source package[2]). Table 1 demonstrates that a Binary classifier does not even generalize well when it is trained on a combination of the databases and evaluated on one of them. Table 1 also shows that Attribution, TripletLoss, and ArcFaceMod techniques perform very well on the individual databases in terms of both AUC and HTER metrics, with Attribution not doing very well on HifiFace, on which both TripletLoss and ArcFaceMod are able to excel.

The cross-database results shown in Table 2 is a mixed bag. ArcFaceMod underperformed compared to Attribution and TripletLoss, which was probably due to the small batch size that we set to 16 for compatibility with other methods, while it is advised to use 512 sized batches [17, 20]. Despite this, all attribution methods generalize better than Bi-



(a) Attribution (b) ArcFaceMod-LR (c) TripletLoss-NN

Fig. 2: t-SNE plots for attribution approaches on the test set of the combined Celeb-DF, FF++, HifiFace, and DeeperForensics. Color shows predicted labels, real videos in blue; marker style – true labels. (Zoom-in to see details or see more online[2]).



(a) Attribution (b) ArcFaceMod-LR (c) TripletLoss-NN

Fig. 3: t-SNE plots for attribution approaches on DF-Mobio database. Color shows predicted labels; marker style – true labels.

nary. However, both Attribution and TripletLoss faired worse compared to the previous work, when a smaller subset of data was used for training (only Celeb-DF and FaceForencisc++). With a smaller training set, TripleLoss led to 13.26 HTER on DeepfakeTIMIT, 26.55 on Google and 23.16 on DF-Mobio (see Table IV in [14]). It may be caused by even larger imbalance in our 'super-set' training data since there a very few real videos compared to deepfakes. Employing training techniques to compensate for the imbalance and to fight against overfitting will be our future work.

Besides the generalization abilities, attribution-based approaches can also be used as tools to analyze the test data, since we can use these pre-trained models to label test data by different deepfake type as illustrated by t-SNE plots in Figure 2 and Figure 3. We computed t-SNE plots from the embeddings for TripleLoss and ArcFaceMod and from the layer before last for Attribution. We colored the test samples of a given database in the colors corresponding to the deepfake type predicted by the attribution method. The markers style (shapes of the dots) correspond to the true labels provided by the database. As we are the authors of DF-Mobio database, we know that its deepfakes are the most similar to Face Swap and Deepfakes types from FaceForensics++, which is what Figure 3 show. This observation testifies to the soundness of using an attribution approach for interpreting the unknown sets of deepfakes. Figure 2 also shows how different attribution methods model the embedding space with ArcFaceMod resulting in angular clusters and TripleLoss in Euclidean ones.

Table 2: Cross-database evaluation of deepfake detection.

Approach	Test DB	AUC	FNR (%)	FPR (%)	HTER (%)
Binary	DF-Mobio	36.90	95.43	3.70	49.56
	Google	54.01	54.27	34.58	44.43
	DeepfakeTIMIT	70.54	38.60	34.38	36.49
Attribution	DF-Mobio	75.52	12.36	59.66	36.01
	Google	87.89	2.20	56.39	**29.30**
	DeepfakeTIMIT	**84.97**	4.88	46.56	**25.72**
TripletLoss-NN	DF-Mobio	**83.15**	22.60	26.03	**24.32**
	Google	84.15	6.06	54.11	30.08
	DeepfakeTIMIT	70.08	52.33	21.25	36.79
ArcFaceMod-LR	DF-Mobio	79.98	9.73	58.65	34.19
	Google	**88.79**	0.55	69.46	35.00
	DeepfakeTIMIT	63.55	27.21	62.19	44.70

4. CONCLUSION

In this paper, we proposed to train a neural network model for attribution task using custom angular loss in addition to triplet loss based training and a simple attribution-based classifier. We showed that regardless of which loss is specifically used, the models trained for attribution on a set that contained nine types of deepfakes perform very well in both same- and cross-databases scenarios. These models can also be used to automatically categorize deepfakes into different types that can help in the forensics analysis of unknown deepfake videos.

5. REFERENCES

[1] Pavel Korshunov and Sébastien Marcel, "Vulnerability assessment and detection of Deepfake videos," in *International Conference on Biometrics (ICB 2019)*, Crete, Greece, June 2019.

[2] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner, "Faceforensics: A large-scale video dataset for forgery detection in human faces," *arXiv preprint arXiv:1803.09179*, 2018.

[3] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "FaceForensics++: Learning to detect manipulated facial images," in *International Conference on Computer Vision (ICCV)*, 2019.

[4] Yuhan Wang, Xu Chen, Junwei Zhu, Wenqing Chu, Ying Tai, Chengjie Wang, Jilin Li, Yongjian Wu, Feiyue Huang, and Rongrong Ji, "HifiFace: 3D shape and semantic prior guided high fidelity face swapping," in *Joint Conference on Artificial Intelligence, IJCAI*, 8 2021, pp. 1136–1142.

[5] Liming Jiang, Ren Li, Wayne Wu, Chen Qian, and Chen Change Loy, "DeeperForensics-1.0: A large-scale dataset for real-world face forgery detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[6] Y. Li, P. Sun, H. Qi, and S. Lyu, "Celeb-DF: A Large-scale Challenging Dataset for DeepFake Forensics," in *IEEE Conference on Computer Vision and Patten Recognition (CVPR)*, 2020.

[7] Brian Dolhansky, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes, Menglin Wang, and Cristian Canton Ferrer, "The deepfake detection challenge dataset," *arXiv preprint arXiv:2006.07397*, 2020.

[8] H.H. Nguyen, J. Yamagishi, and I. Echizen, "Capsule-forensics: using capsule networks to detect forged images and videos," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 2307–2311.

[9] Ruben Tolosana, Sergio Romero-Tapiador, Julian Fierrez, and Ruben Vera-Rodriguez, "Deepfakes evolution: Analysis of facial regions and fake detection performance," *arXiv preprint arXiv:2004.07532*, 2020.

[10] Akash Kumar, Arnav Bhavsar, and Rajesh Verma, "Detecting deepfakes with metric learning," in *International Workshop on Biometrics and Forensics (IWBF)*. IEEE, 2020, pp. 1–6.

[11] Kui Zhu, Bin Wu, and Bai Wang, "Deepfake detection with clustering-based embedding regularization," in *IEEE International Conference on Data Science in Cyberspace (DSC)*. IEEE, 2020, pp. 257–264.

[12] Rayhane Mama and Sam Shi, "Towards deepfake detection that actually works," Dessa, Nov. 2019.

[13] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros, "CNN-generated images are surprisingly easy to spot...for now," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[14] Anubhav Jain, Pavel Korshunov, and Sébastien Marcel, "Improving generalization of deepfake detection by training for attribution," in *International Workshop on Multimedia Signal Processing (MMSP)*, Oct. 2021.

[15] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1800–1807.

[16] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 815–823.

[17] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4685–4694.

[18] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[19] D. M. Montserrat, H. Hao, S. K. Yarlagadda, S. Baireddy, R. Shao, J. Horvath, E. Bartusiak, J. Yang, D. Güera, F. Zhu, and E. J. Delp, "Deepfakes detection with automatic face weighting," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020, pp. 2851–2859.

[20] Fadi Boutros, Naser Damer, Florian Kirchbuchner, and Arjan Kuijper, "ElasticFace: Elastic margin loss for deep face recognition," *arXiv preprint arXiv:2109.09416*, 2021.