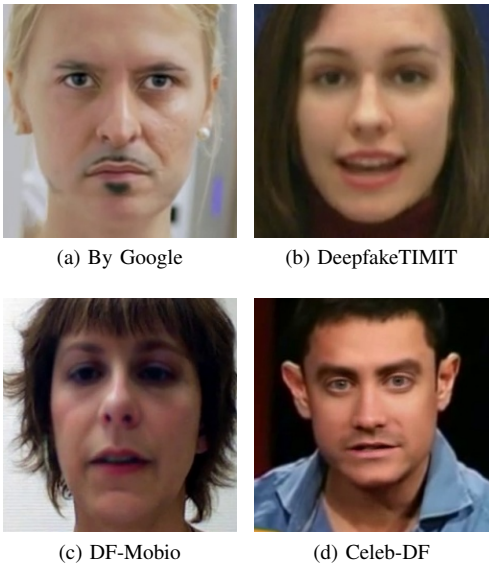
Improving Generalization of Deepfake Detection by Training for Attribution

Anubhav Jain, Pavel Korshunov, and Sébastien Marcel
Idiap Research Institute, Martigny, Switzerland
{anubhav.jain,pavel.korshunov,sebastien.marcel}@idiap.ch

***Abstract*—Recent advances in automated video and audio editing tools, generative adversarial networks (GANs), and social media allow the creation and fast dissemination of high-quality tampered videos, which are commonly called deepfakes. Typically, in these videos, a face is automatically swapped with the face of another person. The simplicity and accessibility of tools for generating deepfakes pose a significant technical challenge for their detection and filtering. In response to the threat, several large datasets of deepfake videos and various methods to detect them were proposed recently. However, the proposed methods suffer from the problem of over-fitting on the training data and the lack of generalization across different databases and generative approaches. 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. Using publicly available datasets from Google and Jigsaw, FaceForensics++, Celeb-DF, DeepfakeTIMIT, and our own large database DF-Mobio, we demonstrate that an XceptionNet and EfficientNet based models trained for an attribution task generalize better to unseen deepfakes and different datasets, compared to the same models trained for a typical binary classification task. We also demonstrate that by training for attribution with a triplet-loss, the generalization in cross-database scenario improves even more, compared to the binary system, while the performance on the same database degrades only marginally.**



(a) By Google (b) DeepfakeTIMIT

(c) DF-Mobio (d) Celeb-DF

Fig. 1. Examples of deepfakes (faces cropped from videos) in different databases.

I. Introduction

Autoencoders and generative adversarial networks (GANs) significantly improved the quality and realism of the automated image generation and face swapping, leading to a deepfake phenomenon. The proverb 'seeing is believing' is starting to lose its meaning when it comes to digital video[1]. The concern for the impact of the widespread deepfake videos on the societal trust in video recording is growing. This public unease prompted researchers to propose various datasets of deepfakes and methods to detect them. Some of the latest approaches demonstrate encouraging accuracies, especially, if they are trained and evaluated on the same datasets [8], [13], [14], [16], [25].

Many databases (see some examples in Figure 1) and Figure 2 with deepfake videos were created to help develop and train deepfake detection methods. One of the first freely available database was DeepfakeTIMIT [7], followed by the FaceForensics database, which contained deepfakes generated from 1000 Youtube videos [15] and which later was extended with a larger set of high-resolution videos provided by Google and Jigsaw [16]. Another recently proposed 5000 videos-large database of deepfakes generated from Youtube videos is Celeb-DF v2 [10]. 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 [4], which was available for download to the participants in the recent Deepfake Detection Challenge hosted by Kaggle[2].

Many methods for deepfake detection were proposed recently, however, as it is typical for deep learning-based approaches, the proposed methods for deepfake detection suffer from the lack of generalization on different types of generative models, video blending techniques used in deepfakes, and data unseen during training [11], [21]. This problem was also demonstrated during Facebook's Deepfake Detection Challenge[2] when the top approaches of the competition have consistently 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 is impeding the advances in deepfake detection and their wide employment.

978-1-6654-3288-7/21/$31.00 ©2021 IEEE

[1] https://edition.cnn.com/interactive/2019/01/business/pentagons-race-against-deepfakes/

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

(a) Original (b) Neural Textures (c) Deepfakes (d) Face Shifter (e) Face2Face (f) Face Swap

Fig. 2. Examples of deepfakes (faces cropped from videos) from the FaceForensics++ database generated from the same original video.

In this paper, we approach the problem of the generalization of deepfake detection by solving a related but different 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. However, at a test time, the model trained for attribution would still be used as a binary classifier, given the training is done by carefully balancing the real and fake data. 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, even if at a test time, the deepfake is of a different type than those in a training set, there would be still a higher chance that it would fall closer to the attack subspace than the subspace of real videos.

To test this hypothesis, we took XceptionNet [3] and EfficientNet [20] models as the popular baseline models for deepfakes detection [15], [16]. To compare with the previous work, we first trained and tested these baseline models as typical binary classifiers. Then, we adjusted the models and the training process to perform an attribution for seven classes of real videos and various deepfakes from FaceForensics++ [15] and Celeb-DF [10] databases to understand whether the model trained in this fashion will generalize better across different databases compared to the binary classifier. To further improve the modeling of the deepfake attacks space and to reduce the effect of over-fitting, we adapted a Siamese network with a triplet-loss (the same Xception and Efficient nets as base models) to train them for distinguishing the same seven real and deepfake classes.

Via an extensive set of experiments, we evaluate the performance of three different approaches (binary, simple attribution, and triplet-loss based) by training on the combination of train subsets from FaceForensics [15] and Celeb-DF [10] and testing on three other datasets: DeepfakeTIMIT [7], Google and Jigsaw [16], and the new large database 'DF-Mobio' of deepfake videos that we have generated[3].

To allow researchers to verify, reproduce, and extend our work, we provide the deepfake detection and attribution systems used in our experiments with corresponding scores as an open-source Python package[4].

[3]https://www.idiap.ch/dataset/df-mobio

[4]Source code: https://gitlab.idiap.ch/bob/bob.paper.deepfake_attribution

II. Related work

An extensive set of the recently available deepfake video databases (see Figure 1 and Figure 2 for examples and Table I for an overview) allow researchers to train and test detection approaches based on very deep neural networks, such as Xception [16], capsules networks [14], and EfficientNet [13], which were shown to outperform the methods based on shallow CNNs, facial physical characteristics [1], [22], or distortion features [9], [24].

Researchers have also shown interest in the attribution of deepfakes or GAN-generated images. Goebel *et al.* [5] proposed an approach to use co-occurrence matrices along with an Xception network for detection, attribution, and localization. Jain *et al.* [6] used a hierarchical CNN-based network for detection and attribution of retouching and GAN-based synthetic alterations. However, none of these methods were applied to the current publicly available deepfake datasets. It was shown that GAN-based images are relatively easier to detect and classify as there is no post-processing done on the images which is not the case in a real-world scenario [21]. Zhang *et al.* [23] proposed to attribute GAN generated images using the concept of seed reconstruction, where the latent variable used for the generation is searched using gradient descent and compared for different models to find the closest match. This approach lacks applicability to image-to-image translation, facial re-enactment, and other similar alterations where the concept of a latent variable does not exist.

Kumar *et al.* [8] were the first to use the triplet loss for detection of deepfakes on the FaceForensics++ datasets using a Facenet-based base model. Zhu *et al.* [25] proposed an XceptionNet based triplet loss network where the positive samples corresponding to every real anchor sample were self-generated images and the negatives were the corresponding deepfake images. But none of these approaches tried to use the models trained for attribution as a binary classifier during evaluation and none of the authors have not focused on generalization.

A. Databases of deepfakes

Table I summarizes the databases of deepfake videos that we have used in the experiments, including DeepfakeTIMIT [7], FaceForensics++ [16], a dataset by Google and Jigsaw [19], Celeb-DF (v2) [10], and a new database of deepfakes, DF-Mobio[3], created by us.

TABLE I
DATABASES OF DEEPFAKES.

Database	Number of swapped identities	Original videos	Deepfakes
DeepfakeTIMIT	32	320	640
DF-Mobio[3]	72	31 950	14 546
from Google and Jigsaw	approx. 150	360	3068
Celeb-DF	1711	590	5639
FaceForensics++	1000	1000	5000

In our experiments, we used FaceForensics++ and Celeb-DF datasets for training and for testing in intra-database scenario. The other three databases we use only for testing to demonstrate the generalization in the inter-database scenario.

The new DF-Mobio dataset is one of the largest databases available with almost $15K$ deepfake and $31K$ real videos (see Table I for the comparison with other databases). Original videos are taken from Mobio database [12], which contains videos of a single person talking to the camera recorded with a phone or a laptop. The scenario simulates the participation in a virtual meeting over Zoom or Skype. The GAN model (using an available open-source code) was trained on face size input of 256×256 pixels. The training images were generated from laptop-recorded videos at 8 fps, resulting in more than $2K$ faces for each subject, the training was done for $40K$ iterations (about 24 hours on Tesla P80 GPU). The whole dataset was used for testing in a cross-database scenario.

III. DETECTION AND ATTRIBUTION OF DEEPFAKES

In this section, we describe three different techniques for training models for deepfake detection: i) a typical binary classification approach, referred as *Binary*, ii) an approach when a model is trained for the attribution task, which we refer to as *Attribution*, and iii) an approach to train a model as a siamese network with triplet-loss for the attribution task, referred to as *TripletLoss*.

To evaluate three different techniques for training deepfake detection models in the most fair way possible, for all methods, we used the same data processing, input sizes, batch sizes, and training parameters. Based on the recommendations given in [2], from each video used for training, we took 15 random frames and extracted 224×224 pixels faces based on Blazeface face detector. We trained models for 20 epochs and selected the best performing model based on the validation loss. We set the batch size to 32 in the experiments compatible with previous work and to 16 for the experiments where we compare the performance between binary classifier and attribution-focused approaches.

A. A binary classifier

As a baseline models, we selected XceptionNet model [3], which was proposed for deepfake detection by Röessler *et al.* [16], and EfficientNet model [20], which was widely used by participants of in Deepfake Detection Challenge[2]. XceptionNet [3] is one of the popular models for deepfake detection and its performance can be compared with several previous results [2], [10], [14], [16]. For compatibility reasons, as it was originally suggested by the authors of FaceForensics++ database [16], we used both networks with weights pre-trained on ImageNet [17]. In a binary classification mode, the last fully connected layer was replaced with a single output layer and a sigmoid activation, the training was done using an Adam optimizer with a learning rate of 0.0002 and a categorical cross-entropy loss.

B. An attribution-based approach

We adapted the Xception and Efficient models to the attribution task (to distinguish different deepfakes and real videos) by replacing the output layer with a dense layer consisting of seven neurons, which correspond to six types of the deepfakes and one original class. First five deepfake types correspond to different attacks from FaceForensics++ dataset: Neural Textures, Face2Face, Face Swap, Face Shifter, and Deepfakes (see examples in Figure 2), and the last type corresponds to Celeb-DF dataset, where all deepfakes were generated using one approach.

Since our intention is to evaluate the model trained for attribution task as a simple binary classifier that separates real videos from all deepfakes, we took special care of how we compose the training batches. If it would be a typical attribution model, each of the seven classes would have an approximately equal representation within a single training batch. However, the model trained this way would give the same importance to one real class as to any of the deepfake classes, which does not reflect the practical scenario.

Therefore, in our training batches, the number of samples from real videos is three times larger than the number of samples for each type of deepfake attacks. Moreover, we insure that each type of real video (one from FaceForensics++ dataset, one from Youtube part of Celeb-DF and one from celeb part of Celeb-DF) are equally represented in each batch as well. With this approach to batch construction, some of the real data was over-sampled, especially since Celeb-DF has much fewer real videos than deepfakes (see Table I).

All other parameters used during training are the same as we set for a baseline binary classifier in Section III-A.

C. An attribution based on triplet-loss

The inspiration for using triplet-loss to train an attribution model come from the FaceNet [18] model and training approach proposed for face recognition task. The aim of a Siamese network trained using triplet-loss is to reduce the intra-class distance and increase the inter-class distance in the embedding space, typically described by vectors from the layer

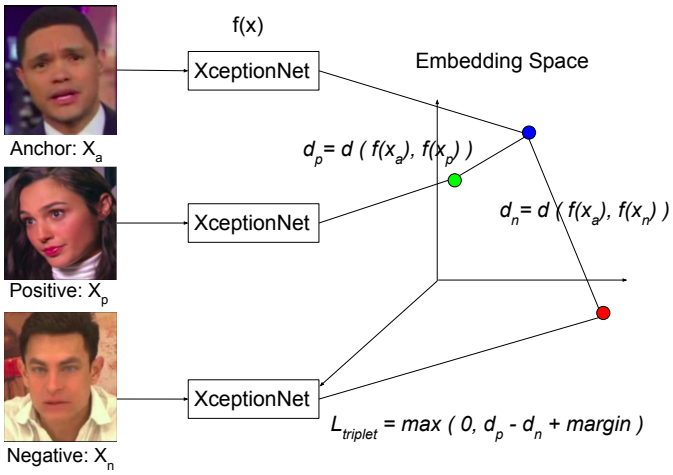


Fig. 3. Representation of the triplet loss network using the XceptionNet as the base model for generating embeddings of size 64.

before the last. We hypothesise that using triplet-loss training can also improve the separation of all classes of the deepfakes and, more importantly, a real class from all the deepfakes.

Therefore, we adapt the baseline models Xception and Efficient (see Section III-A) for triplet-loss training by replacing the last layer with an embedding layer of size 64. The layer is also normalized using l2 norm as proposed in the original paper [18]. Then, a typical siamese network is constructed where three baseline networks with shared weights are trained using negative, anchor, and positive samples and a triplet loss as illustrated in Figure 3.

The triplet loss can be represented using equation $TripletLoss = max(0, d(a,p) - d(a,n) + margin)$, where d is the distance metric, which in this case is euclidean distance between any two embeddings, a is the anchor image, p is the positive, and n is the negative. In the terminology of the triplet-loss, the anchor and positive images are from the same class (the distance between their embeddings is minimized) and the negative is from another class (the distance from an anchor is maximized). We use semi-hard triplets, which is the case where $d(a,p) < d(a,n) < d(a,p) + margin$ as suggested by the original authors of [18].

The complexity of training a siamese network lies in the way the training batches are constructed. As per the original approach, we use an online triplet mining methodology. The process allows having more triplets for a single batch of images and does not require finding triplets offline.

When training of triplet-loss based attribution model, we use learning rate 0.00001, which we experimentally found allows the model to converge more efficiently. Since siamese networks are memory demanding, we used batch size 16, and for the fair comparison, we used the same batch when comparing the results with binary and simple attribution classifiers.

We convert a triplet-loss network to an attribution classifier by training another classifier using the TripletLoss embeddings. We tried different classifiers: k-nearest neighbor (NN), logistic regression (LR), and support vector machine (SVM). We trained these classifiers using the embeddings extracted for validation sets of the corresponding databases, while their hyper parameters were tuned using the test set. We also experimentally found the best K value for the nearest neighbor classifier using grid search as 14, the best regularization parameter for logistic regression as 0.1, and regularization parameter for SVM as 100.

D. Evaluation methodology

We evaluate the proposed approaches for intra- and inter-database (i.e., cross-database) scenarios. In the first part, the evaluation of the classifier is performed on the same datasets that were used for training, that is, on the same types of attacks and real videos. The cross-database evaluation tests the generalization capabilities of the models on unseen attacks. In this scenario, we train and tune hyper parameters of the model on one dataset and test on completely different dataset.

In the intra-database scenario, we have trained and tested our three approaches on a combination of Celeb-DF and FaceForensics++ databases.

In the cross-database scenario, models trained and tuned on Celeb-DF and FaceForensics++ databases are evaluated on completely new databases with unseen real videos and deepfakes such as the Google, DF-Mobio, and the Deepfake-TIMIT datasets. The models trained for attribution are treated as binary classifiers during this phase. We simply treat all the fake classes as one, implying that any fake video classified into any of these classes is correctly classified.

Recent deepfake detection approaches are mostly evaluated either by using an area under the curve (AUC) or an accuracy [1], [6], [10], [13], [16]. However neither of them are well suited for the evaluation of highly unbalanced deepfake datasets (see Table I). The accuracy metric implicitly uses threshold 0.5 and requires an equal number of real videos and attacks in a test set, while AUC does not have an operational threshold, which is important for practical systems.

Therefore, we compute commonly used metrics for evaluation of classification systems: false positive rate (FPR), which is the the proportion of fake samples classified as real, and false negative rate (FNR), which is the proportion of real samples classified as fake. These rates are generally defined, for a given threshold θ, as follows:

$$\begin{aligned} FPR(\theta) &= \frac{|\{h_{neg} \mid h_{neg} \geq \theta\}|}{|\{h_{neg}\}|} \\ FNR(\theta) &= \frac{|\{h_{pos} \mid h_{pos} < \theta\}|}{|\{h_{pos}\}|}, \end{aligned} \tag{1}$$

where h_{pos} is a score (typically close to 1 if classification is accurate) for original real samples and h_{neg} is a score (typically close to 0) for the deepfakes.

We define the threshold θ_{fpr} on the validation set to correspond to the FPR value of 10%, which means 10% of fake videos are allowed to be misclassified as real. Using this threshold θ_{fpr} on the scores of the test set will result in test FPR and FNR values. As a single value metric, we can then use the half total error rate (HTER) defined as $HTER(\theta_{fpr}) = \frac{FPR_{test} + FNR_{test}}{2}$.

TABLE II
EVALUATION OF BINARY CLASSIFIER WHEN TRAINED AND EVALUATED ON THE SAME DATABASE, WITH FPR=10% THRESHOLD ON VALIDATION SET.

Dataset	AUC	FPR (%)	FNR (%)	HTER (%)
Celeb-DF	99.60	4.16	2.09	3.12
FaceForensics++	97.28	9.43	8.69	9.06

TABLE III
EVALUATION OF XCEPTION-BASED DEEPFAKE DETECTION METHODS, TRAINED ON CELEB-DF AND FACEFORENSICS++, IN AN INTRA-DATABASE SCENARIO.

Approach	Test DB	AUC	FPR (%)	FNR (%)	HTER (%)
Binary	Celeb-DF	97.99	0.83	30.02	15.42
	FaceForensics++	60.52	93.95	3.34	48.65
Attribution	Celeb-DF	**99.21**	1.88	9.55	**5.71**
	FaceForensics++	**98.91**	4.05	6.46	**5.25**
TripletLoss-LR	CelebDF	97.42	7.13	8.59	7.86
	FaceForensics++	95.73	9.19	10.68	9.93
TripletLoss-NN	CelebDF	96.72	6.50	12.84	9.67
	FaceForensics++	93.75	14.33	7.59	10.96
TripletLoss-SVM	CelebDF	97.37	6.42	9.59	8.00
	FaceForensics++	95.81	9.10	10.19	9.64

In addition to reporting FPR, FNR, and HTER values for the scores of the test set, we still report the area under the curve (AUC) metric for comparison with the current deepfake detection literature.

IV. EXPERIMENTAL RESULTS

To verify the choice of our baseline Xception and Efficient models and to be able to compare with the previous work, we have trained and tested a binary Xception-based classifier separately on FaceForensics++ and Celeb-DF datasets. Table II shows the results for each database. The results are consistent with those reported in the literature [2], [10], [16], with only notable exception to the results reported by the authors of Celeb-DF [10], where they claimed that the Xception-based method had the AUC value of 65.5%, while we managed to achieve 99.60% AUC on the test set.

Our intra- and cross-database experiments aimed to make the results as comparable as possible. We trained all detection approaches using training sets of both Celeb-DF and FaceForensics++ datasets. We evaluated these approaches on the test sets of either Celeb-DF or FaceForensics++ in the intra-dataset scenario and on the entirely unseen datasets (where both real videos and deepakes are different from the training) of DeepfakeTIMIT, Google (from Google and Jigsaw), and Mobio in the inter-database or cross-database scenario. Since the training and test data are completely independent, it removes the need for an ablation study.

Tables III and V show the results for three methods: the baseline binary classifiers for Xception and Efficient nets, the same classifiers trained for an attribution task, and three versions of the triplet-loss classifier when its embeddings were used to further train k-nearest neighbor (NN), logistic regression (LR) and support vector machine (SVM). These tables demonstrate that a Binary classifier does not even generalize well when it is trained on two databases and evaluated on one of them, since then its performance degrades significantly compared to the same train/test scenario in Table II. Note that exactly the same test sets are used to compute values for Tables III, V and Table II.

Tables III and V also show that Attribution and all variants of TripletLoss technique generalize better than Binary classifier on both Celeb-DF and FaceForensics++ datasets. The Attribution approach demonstrates a better performance in intra-database experiments with both higher AUC and lower HTER compared to TripletLoss.

However, where TripletLoss really shines is in the cross-database scenario, results for which are shown in Tables IV and VI. Note that in this case, the approaches and models, trained in the same way as for Tables III and V on a combination of Celeb-DF and FaceForensics++, are now evaluated on the other three databases that contain very different real videos and different types of deepfakes. Tables IV and VI demonstrate that TripletLoss variants, generalize better on completely unseen databases compared to other models, especially when looking at HTER, FPR, and FNR values (more practical and important metrics as discussed in Section III-D). The results for DeepfakeTIMIT are a bit of an exception for the Binary classifier, but it is a very small and simple dataset, so this anomaly can be neglected. What is more important is the ability of TripletLoss, especially its NN variant, to generalize relatively well on Google and Mobio datasets, which contain very different deepfakes compared to Celeb-DF and FaceForensics++. Overall, from the results of intra- and inter-database experiments, the TripletLoss-NN model seem to perform the best. These results are very encouraging for the idea that the generalization of any deep model can be improved if it is trained for attribution task instead of as a typical binary classifier.

TABLE IV
EVALUATION OF XCEPTION-BASED DETECTION METHODS, TRAINED ON CELEB-DF AND FACEFORENSICS++, IN A CROSS-DATABASE SCENARIO.

Approach	Test DB	AUC	FPR (%)	FNR (%)	HTER (%)
Binary	DeepfakeTIMIT	**99.15**	0.00	57.07	28.54
	Google	61.66	50.13	35.36	42.74
	DF-Mobio	47.36	83.83	16.18	50.00
Attribution	DeepfakeTIMIT	97.12	0.64	55.54	28.09
	Google	80.23	16.14	42.68	29.41
	Mobio	73.06	35.21	33.38	34.29
TripletLoss-LR	DeepfakeTIMIT	94.90	4.56	22.53	13.55
	Google	80.82	13.16	42.50	27.83
	Mobio	**83.50**	14.55	39.89	27.22
TripletLoss-NN	DeepfakeTIMIT	91.54	11.53	14.98	**13.26**
	Google	79.38	19.73	33.38	**26.55**
	Mobio	83.24	24.23	22.10	**23.16**
TripletLoss-SVM	DeepfakeTIMIT	94.73	4.06	27.19	15.63
	Google	**80.78**	12.74	45.58	29.16
	Mobio	83.26	18.02	36.63	27.32

TABLE V
EVALUATION OF EFFICIENT-BASED DEEPFAKE DETECTION METHODS, TRAINED ON CELEB-DF AND FACEFORENSICS++, IN AN INTRA-DATABASE SCENARIO.

Approach	Test DB	AUC	FPR (%)	FNR (%)	HTER (%)
Binary	Celeb-DF	**98.64**	0.83	26.12	13.47
	FaceForensics++	57.05	98.19	1.27	49.73
Attribution	Celeb-DF	97.08	2.18	22.65	12.41
	FaceForensics++	**97.95**	25.43	1.54	13.49
TripletLoss-LR	Celeb-DF	95.58	5.63	15.96	**10.80**
	FaceForensics++	97.08	7.95	7.10	**7.53**
TripletLoss-NN	Celeb-DF	94.07	7.85	15.49	11.67
	FaceForensics++	94.93	10.14	6.22	8.18
TripletLoss-SVM	Celeb-DF	95.46	5.26	17.12	11.19
	FaceForensics++	97.07	9.05	6.54	7.80

TABLE VI
EVALUATION OF EFFICIENT-BASED DETECTION METHODS, TRAINED ON CELEB-DF AND FACEFORENSICS++, IN A CROSS-DATABASE SCENARIO.

Approach	Test DB	AUC	FPR (%)	FNR (%)	HTER (%)
Binary	DeepfakeTIMIT	99.04	0.59	11.02	5.80
	Google	68.94	64.22	17.72	40.97
	DF-Mobio	48.55	92.50	5.24	48.87
Attribution	DeepfakeTIMIT	**99.11**	0.20	14.58	7.39
	Google	83.68	42.56	13.83	28.19
	DF-Mobio	85.26	76.21	2.37	39.29
TripletLoss-LR	DeepfakeTIMIT	98.96	0.47	15.76	8.11
	Google	**85.14**	9.17	41.50	25.33
	DF-Mobio	90.35	5.23	42.02	23.62
TripletLoss-NN	DeepfakeTIMIT	98.24	1.67	9.94	**5.81**
	Google	82.73	13.12	35.96	**24.54**
	DF-Mobio	89.72	9.30	29.53	**19.41**
TripletLoss-SVM	DeepfakeTIMIT	99.01	0.54	18.86	9.70
	Google	84.64	9.53	43.26	26.39
	DF-Mobio	**91.01**	6.27	37.17	21.72

V. CONCLUSION

The evaluation of a binary deepfake detection classifier in the cross-database scenario demonstrates that it performs very well when trained and evaluated on a single database but struggles significantly when trained on one database and evaluated on another. However, in this paper, we proposed to use a model trained for an attribution task as the binary classifier, and it showed to generalize significantly better when distinguishing real and deepfake videos even from totally different databases. Using Siamese networks with triplet-loss improves the generalization further.

It means that a typical binary detection algorithm, struggling to generalize to unseen and unknown data, can be trained to perform attribution, and this simple adjustment can significantly improve its generalization at detection time.

ACKNOWLEDGEMENTS

This work was funded by Hasler Foundation's VERIFAKE project and Swiss Center for Biometrics Research and Testing.

REFERENCES

[1] S. Agarwal, T. El-Gaaly, H. Farid, and S. Lim. Detecting deep-fake videos from appearance and behavior. *arXiv preprint*, 2020. 2, 4

[2] N. Bonettini, E. D. Cannas, S. Mandelli, L. Bondi, P. Bestagini, and S. Tubaro. Video face manipulation detection through ensemble of CNNs. *arXiv preprint arXiv:2004.07676*, 2020. 3, 5

[3] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, pages 1800–1807, 2017. 2, 3

[4] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. C. Ferrer. The deepfake detection challenge dataset. *arXiv preprint arXiv:2006.07397*, 2020. 1

[5] M. Goebel, L. Nataraj, T. Nanjundaswamy, T. M. Mohammed, S. Chandrasekaran, and B. Manjunath. Detection, attribution and localization of GAN generated images. *arXiv preprint arXiv:2007.10466*, 2020. 2

[6] A. Jain, P. Majumdar, R. Singh, and M. Vatsa. Detecting GANs and retouching based digital alterations via DAD-HCNN. In *CVPR*, pages 672–673, 2020. 2, 4

[7] P. Korshunov and S. Marcel. Vulnerability assessment and detection of Deepfake videos. In *International Conference on Biometrics (ICB 2019)*, Crete, Greece, June 2019. 1, 2

[8] A. Kumar, A. Bhavsar, and R. Verma. Detecting deepfakes with metric learning. In *International Workshop on Biometrics and Forensics (IWBF)*, pages 1–6. IEEE, 2020. 1, 2

[9] Y. Li, M. Chang, and S. Lyu. In ictu oculi: Exposing ai created fake videos by detecting eye blinking. In *IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–7, 2018. 2

[10] 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*, 2020. 1, 2, 3, 4, 5

[11] R. Mama and S. Shi. Towards deepfake detection that actually works. Dessa, Nov. 2019. 1

[12] C. McCool, S. Marcel, A. Hadid, M. Pietikäinen, P. Matejka, J. Cernocký, N. Poh, J. Kittler, A. Larcher, C. Lévy, D. Matrouf, J. Bonastre, P. Tresadern, and T. Cootes. Bi-modal person recognition on a mobile phone: Using mobile phone data. In *IEEE International Conference on Multimedia and Expo Workshops*, pages 635–640, 2012. 3

[13] 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)*, pages 2851–2859, 2020. 1, 2, 4

[14] 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)*, pages 2307–2311, 2019. 1, 2, 3

[15] 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. 1, 2

[16] 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. 1, 2, 3, 4, 5

[17] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 3

[18] F. Schroff, D. Kalenichenko, and J. Philbin. FaceNet: A unified embedding for face recognition and clustering. In *CVPR*, pages 815–823, June 2015. 3, 4

[19] D. Stanton, P. Karlsson, A. Vorobyov, T. Leung, J. Childs, A. Sud, and C. Bregler. Contributing data to deepfake detection research. In *Blogpost*, 2019. 2

[20] M. Tan and Q. V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019. 2, 3

[21] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros. CNN-generated images are surprisingly easy to spot...for now. In *CVPR*, 2020. 1, 2

[22] X. Yang, Y. Li, H. Qi, and S. Lyu. Exposing GAN-synthesized faces using landmark locations. In *ACM Workshop on Information Hiding and Multimedia Security*, pages 113–118, June 2019. 2

[23] B. Zhang, J. P. Zhou, I. Shumailov, and N. Papernot. On attribution of deepfakes. *arXiv preprint arXiv:2008.09194*, 2021. 2

[24] Y. Zhang, L. Zheng, and V. L. L. Thing. Automated face swapping and its detection. In *IEEE International Conference on Signal and Image Processing (ICSIP)*, pages 15–19, Aug 2017. 2

[25] K. Zhu, B. Wu, and B. Wang. Deepfake detection with clustering-based embedding regularization. In *IEEE International Conference on Data Science in Cyberspace (DSC)*, pages 257–264. IEEE, 2020. 1, 2