A Generalizable Deepfake Detector based on Neural Conditional Distribution Modelling

Ali Khodabakhsh[1] Christoph Busch[2]

Abstract: Photo- and video-realistic generation techniques have become a reality following the advent of deep neural networks. Consequently, there are immense concerns regarding the difficulty in differentiating what content is real from what is synthetic. An example of video-realistic generation techniques is the infamous Deepfakes, which exploit the main modality by which humans identify each other. Deepfakes are a category of synthetic face generation methods and are commonly based on generative adversarial networks. In this article, we propose a novel two-step synthetic face image detection method in which general-purpose features are extracted in a first step, trivializing the task of detecting synthetic images. The anomaly detector predicts the conditional probabilities for observing every individual pixel in the image and is trained on pristine data only. The extracted anomaly features demonstrate true generalization capacity across widely different unknown synthesis methods while showing a minimal loss in performance with regard to the detection of known synthetic samples.

Keywords: Deepfake, Video Forensics, Generative Adversarial Networks, PixelCNN, Universal Background Model.

1 Introduction

Advancements in the computational capacity of modern graphical processing units (GPUs) in the past decades allowed the realization of deep neural network models. Deep learning, among other contributions, provided solutions for the synthesis of photo- and video-realistic content, challenging the existing manipulation detection methods in video forensics. An especial case of such synthetic signals is “Deepfakes”, which are typically generated by generative adversarial networks (GANs). Deepfakes in combination with obfuscation in various forms have shown to be effective at fooling human subjects [Ro19].

The research community has responded to this threat by developing various detection methods. Yu et al. in [YDF18] made use of unique GAN fingerprints for the detection of fake images generated by these models. RNNs have been used for temporal-aware detection of Deepfakes by Guera et al. in [GD18]. The spectrum domain is used by Zhang et al. [ZKC19] for the detection of GAN generated images.

Most of the existing detection methods are, however, complex and have narrow applicability as they are trained to detect specific types of synthetic signals and fail to generalize [Kh18]. Few publications try to address the detection of synthetic samples from unknown generation models. In [St19], Stehouwer et al. used attention mechanisms and achieved remarkable performance over various generation techniques. Nataraj et al. [Na19] used pixel co-occurrence matrices for generalized detection across different GAN architectures. In [Ma19], Marra et al. utilized multi-task learning incrementally for detecting

[1] NTNU, IIK, Norwegian Biometrics Lab, Gjovik, NO, ali.khodabakhsh@ntnu.no

[2] NTNU, IIK, Norwegian Biometrics Lab, Gjovik, NO, christoph.busch@ntnu.no

synthetic images coming from unknown GAN models. Zhou et al. [Zh17] proposed a two-stream classification network architecture based on steganalysis features. Afchar et al. [Af18] utilized mesoscopic features along with shallow networks gaining robustness against unknown synthetic images. Rossler et al. [Ro19] evaluated different detection systems on a large dataset of diverse synthetic samples and achieved the best performance with a pretrained XceptionNet neural network. For an extensive review on the related literature, please refer to [Ve20].

Despite major progress in the detection of synthetic face images, the generalization problem across widely different generation techniques remains a major issue. In this article, we propose a novel general-purpose feature. The subsequent trivialization enables a simple detector to reliably detect unknown attacks form widely different generation techniques. The proposed method achieves this by suppressing the content of the input signal while faithfully conserving the detection-relevant information. The rest of this article is organized as follows: Section 2 explains the proposed two-step method along with the rationale behind it. Section 3 explains the experimental setup used for showcasing the performance of the method, and Section 4 discusses the findings of the article. Finally, Section 5 concludes the article.

2 Methodology

Synthetic images contain artefacts that can be used for detection and can act like fingerprints for identification of their generation process. These traces, however, are often minuscule and can be severely obscured by the actual content of the images to the extent of becoming imperceptible to the eyes of the viewer as well as the automated detection systems. We hypothesize that in the synthetic face detection task, the actual content of images acts as a strong noise, and removing them would unveil these traces and greatly simplify the task of synthetic face detection. However, this approach requires knowledge of the actual content of the image for reference.

In the absence of a reference to be subtracted from the image, the likelihood of the image to an accurate probability distribution of pristine face images would serve as a suitable proxy. To make the accurate modeling of the probability distribution over the face image space practical, the image can be broken down into smaller segments, and the probability distribution over individual segments of the image conditioned on the previous segments can be modeled.

2.1 Pixel RNN

The probability distribution of intensity values in each pixel conditioned on pixels before (in raster order) in pristine images can be modeled with a PixelRNN model [VDOKK16]. In this model, for each pixel i, the probability distribution (in the form of a Logistic mixture model) of observing the current value given all previous pixel values is learned by a recurrent or a masked convolutional neural network. This network would then be able to predict the probability distribution of pixel values for each pixel location conditioned on the pixel values before it. This probability distribution can then be used to measure the likelihood of observing a specific pixel value in location x_i given all pixel values before it ($log(p(x_i|x_{<i}))$). By repeating this operation over all the pixels in an input image, one can calculate a likelihood matrix with the same size as the input image. Consequently, the probability of observing the input image can be calculated as $log(p(x)) = \sum_{i=0}^{n} log(p(x_i|x_{<i}))$. For the purpose of this study, an improved variant of PixelRNN named PixelCNN++ [Sa17] is used.

2.2 Classification

The probability of the input image is a feature that can spot anomalies and can directly be used for classification. However, the conditional probability matrix corresponding to the log-likelihood of observing every single pixel intensity can serve as a better feature for classification as it contains additional information with respect to the location of anomalies and the anomaly strength at each location. For achieving a higher detection rate, one can use the model trained in the previous step as an anomaly feature extractor, or in more precise terms a universal background model (UBM). The term UBM signifies that the model is universally used regardless of the synthetic method in question in the detection task. Furthermore, it signifies that the model is a background preprocessing step which postpones the classification task to a second step. Consequently, a classifier can be trained on the output of the UBM model which is in the form of a conditional probability matrix in a supervised manner. Ideally, as the complexity of the detection problem is substantially reduced following the feature extraction step, a simple classifier should be sufficient for detection of synthetic faces. In this study, we use a very simple and small neural network for classification.

2.3 Generalization Performance

To measure the generalization capacity of a model, a common practice is to split the generation techniques to known and unknown methods. Next, the model is trained on synthetic data from the known methods and tested on the data from the unknown methods. To show the generalization capacity of our proposed method, we follow the same convention and do generalization tests in a leave-one-out (LOO) manner. For each generation method, we consider all other methods to be known and measure the detection performance on the single unknown method. The overall generalization performance is then measured by aggregating them over all the leave-one-out runs.

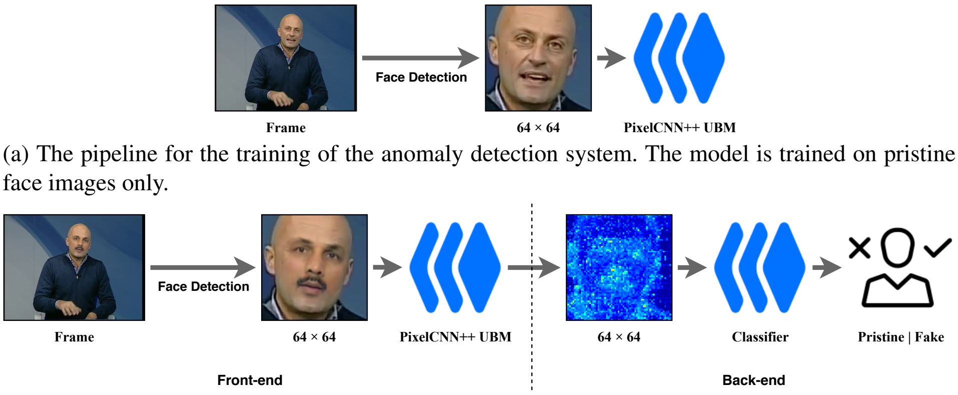
3 Experiment Setup

For the purpose of this study, the FaceForensics dataset [Ro19] is selected as a large dataset containing four manipulation techniques, namely Deepfakes[3], Face2Face [Th16], Faceswap[4], and Neural Textures [TZN19]. This dataset contains 1000 pristine videos along with 1000 from each manipulation technique, each split into three sets of training (with 700 videos), development (with 150 videos), and test (with 150 videos). The videos are collected from YouTube and have a minimum quality of 480p (VGA). The videos are provided in three different quality levels to simulate the conditions of video processing in

[3] https://github.com/deepfakes/faceswap

[4] https://github.com/MarekKowalski/FaceSwap/

social networks. For extraction of face images from the videos, the Dlib toolkit [Ki09] is used, and the detected face images are resized to 64 × 64. As the focus of this study is generalizability across the four completely different generation techniques, we limit the experiments to uncompressed data. Subsequently, the models are trained on individual cropped face images from frames as shown in Figure 1, and the detection performance is evaluated in terms of the frame-level detection accuracy.



(a) The pipeline for the training of the anomaly detection system. The model is trained on pristine face images only.

(b) The training and evaluation pipeline of the classifier. The pre-trained anomaly detection model is used as an anomaly feature extractor.

Fig. 1: The training and evaluation pipelines of the proposed method. UBM stands for universal background model and represents the probability distribution based anomaly extraction system.

The UBM model used for experiments is the Tensorflow implementation of PixelCNN++ [Sa17]. The default architecture, consisting of three blocks with five ResNet layers and 160 filters in each layer is used. A single model with 94 million parameters is trained for five epochs on natural images only from the training set, with a learning rate of 0.0001 on a single GPU in an end-to-end manner.

As the complexity of the detection problem is reduced in the anomaly feature extraction step to an extent that the synthesis artifacts are visible in its output (see Figure 4), a very simple classifier based on LeNet-5 [Le98] is used for detection of synthetic faces from known and unknown generation methods. The modified architecture summarized in Figure 2 is small enough to be trained on a CPU and has less than one million parameters. For each experiment, one classifier is trained on the available training data for 25 epochs with a learning rate of 0.001. The activation function used is the ReLU function, and to improve the convergence speed, batch normalization is used between the output of the layers and the activation function. The overall detection pipeline is shown in Figure 1b.

4 Results and Discussion

In this section, we first discuss the characteristics of the anomaly extraction method and then summarize the performance of the method on both known and unknown attack detection scenarios.

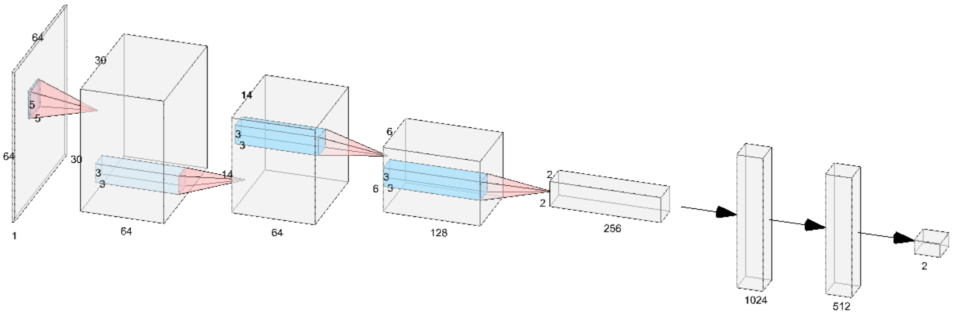


Fig. 2: The diagram of the classifier architecture. Each convolution is followed by a 2x2 maxpooling layer and ReLU activation. The network has a total of 933,442 parameters.

4.1 Features

Figure 3 shows the histogram of log-likelihoods for images in the validation data for pristine images as well as the synthetic images. The log-likelihood values for the pristine images are higher than the synthetic images, however, there is a significant overlap between the distributions. Deepfakes show higher log-likelihood values compared to the other synthesis methods. These results show the discrimination power of the observation probability of the images for synthetic face image detection. However, the image probability distributions have significant overlap, and cannot be relied on as a high-performance detection score.

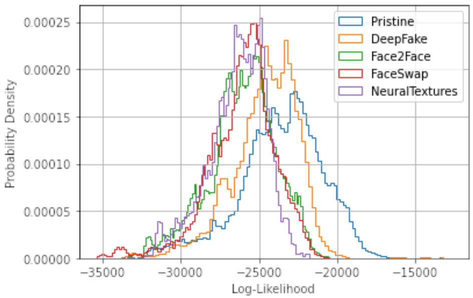
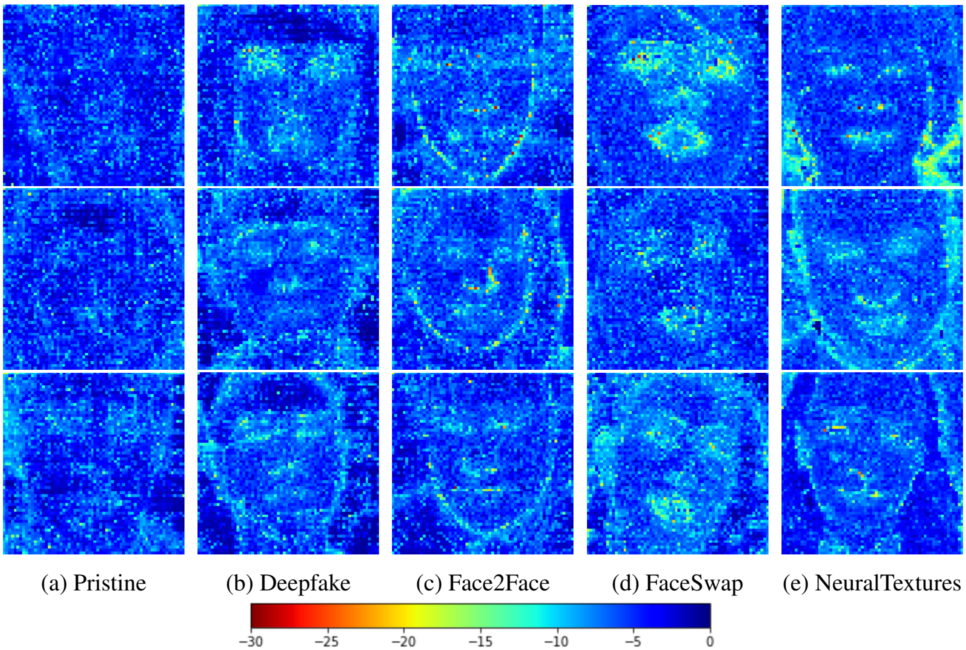


Fig. 3: The image log-likelihood probability for pristine images and synthetic images in the development data.

To achieve a better performance, we can rely on the pixel log-likelihood *images* extracted by the UBM model as anomaly features. Figure 4 visualizes examples of these *images* from the pristine data as well as the four generation techniques. In this figure, a drastic difference is observable between the pristine images and the synthetic images. The traces of the synthesis process are visible as low likelihood points in yellow and red on the image. Furthermore, each generation method shows a unique footprint in all examples. The Deepfakes have artifacts in the shape of the spliced synthetic face area over the background image. The Face2Face technique results in low likelihood pixel values on the edges of the 3D facial features such as nose and jawline. FaceSwap technique results in low likelihood areas around the eyes and the mouth. Lastly, NeuralTextures inhibits individual low-likelihood pixels on the nose and eye regions.



(a) Pristine (b) Deepfake (c) Face2Face (d) FaceSwap (e) NeuralTextures

Fig. 4: Examples of the log-likelihood output matrix of the universal background model on pristine and synthetic face images. The name of the generation method is mentioned below each column. As shown in the color bar, red signifies low log-likelihood probability, while blue signifies high.

4.2 Known Synthetic Face Detection

To measure the discriminative power of the likelihood images, we used the simple classifier explained in the previous section for synthetic face detection on each individual method. The results are reported in Table 1 along with the performance of the baseline methods from [Ro19]. The proposed method performs on par with the baseline methods despite having a smaller input image size and a much smaller number of parameters. These results confirm that the log-likelihood images conserve the information valuable for detection faithfully while reducing the detection complexity by removing the unhelpful information.

4.3 Unknown Synthetic Face Detection

The performance of the proposed method in the unknown synthetic face detection scenario is summarized in Table 2. The proposed method shows an acceptable detection rates for all four synthesis methods while showing above 96% on three out of four in LOO generalization experiments. The performance of Face2Face method gets slight improvement over the known case due to the larger training data available in the LOO scenario.

5 Conclusion

In this article, we introduced a truly generalizable synthetic face image detection method which achieves an outstanding average detection accuracy of 95.73% on unknown synthetic methods. The synthetic methods are from widely different synthesis mechanisms

Tab. 1: The performance of the proposed method in terms of detection accuracy in known synthetic face image detection scenario in comparison with existing methods adapted from [Ro19]. (DF: DeepFakes, F2F: Face2Face, FS:FaceSwap, NT:NeuralTextures)

	Input Size	DF [%]	F2F [%]	FS [%]	NT [%]
Steg. Features+SVM [FK12]	128×128	99.03	99.13	98.27	99.88
Cozzolinoet al. [CPV17]	128×128	98.83	98.56	98.89	99.88
Bayar and Stamm [BS16]	128×128	99.28	98.79	98.98	98.78
Rahmouniet al. [Ra17]	100×100	98.03	98.96	98.94	96.06
MesoNet [Af18]	256×256	98.41	97.96	96.07	97.05
XceptionNet [Ch17]	299×299	99.59	99.61	99.14	99.36
Proposed Method	64×64	99.30	98.25	99.11	98.46

Tab. 2: The performance of the proposed method on unknown synthetic samples in terms of detection accuracy. For each method, the system is trained on the other three synthesis data and did not observe a single sample of the method in question during training. The average detection accuracy is also reported. (DF: DeepFakes, F2F: Face2Face, FS:FaceSwap, NT:NeuralTextures)

	DF [%]	F2F [%]	FS [%]	NT [%]	Avg [%]
LOO Detection Accuracy	89.26	98.41	96.80	98.44	95.73

ranging from Deepfakes from generative adversarial networks to FaceSwap. The proposed method consists of a preprocessing step where the content of the image is suppressed, and the anomaly locations and anomaly strengths are extracted. The classification is then done by a simple classifier. The anomaly extraction step is trained on natural images only and preserves the detection-relevant information faithfully in the form of observation log-likelihood probability. The detectors' success provides new hopes for addressing the generalization problem over widely different generation processes.

References

[Af18] Afchar, Darius; Nozick, Vincent; Yamagishi, Junichi; Echizen, Isao: Mesonet: a compact facial video forgery detection network. In: WIFS. IEEE, pp. 1–7, 2018.

[BS16] Bayar, Belhassen; Stamm, Matthew C: A deep learning approach to universal image manipulation detection using a new convolutional layer. In: ACM IH&MMSec. pp. 5–10, 2016.

[Ch17] Chollet, François: Xception: Deep learning with depthwise separable convolutions. In: IEEE CVPR. pp. 1251–1258, 2017.

[CPV17] Cozzolino, Davide; Poggi, Giovanni; Verdoliva, Luisa: Recasting residual-based local descriptors as convolutional neural networks: an application to image forgery detection. In: ACM IH&MMSec. pp. 159–164, 2017.

[FK12] Fridrich, Jessica; Kodovsky, Jan: Rich models for steganalysis of digital images. IEEE Transactions on Information Forensics and Security, 7(3):868–882, 2012.

[GD18] Güera, David; Delp, Edward J: Deepfake video detection using recurrent neural networks. In: IEEE AVSS. IEEE, pp. 1–6, 2018.

[Kh18] Khodabakhsh, A.; Ramachandra, R.; Raja, K.; Wasnik, P.; Busch, C.: Fake Face Detection Methods: Can They Be Generalized? In: BIOSIG. pp. 1–6, 2018.

[Ki09] King, Davis E: Dlib-ml: A machine learning toolkit. Journal of Machine Learning Research, 10(Jul):1755–1758, 2009.

[Le98] LeCun, Yann; Bottou, Léon; Bengio, Yoshua; Haffner, Patrick: Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

[Ma19] Marra, Francesco; Saltori, Cristiano; Boato, Giulia; Verdoliva, Luisa: Incremental learning for the detection and classification of GAN-generated images. arXiv preprint arXiv:1910.01568, 2019.

[Na19] Nataraj, Lakshmanan; Mohammed, Tajuddin Manhar; Manjunath, BS; Chandrasekaran, Shivkumar; Flenner, Arjuna; Bappy, Jawadul H; Roy-Chowdhury, Amit K: Detecting GAN generated fake images using co-occurrence matrices. Electronic Imaging, 2019(5):532–1, 2019.

[Ra17] Rahmouni, Nicolas; Nozick, Vincent; Yamagishi, Junichi; Echizen, Isao: Distinguishing computer graphics from natural images using convolution neural networks. In: WIFS. IEEE, pp. 1–6, 2017.

[Ro19] Rossler, Andreas; Cozzolino, Davide; Verdoliva, Luisa; Riess, Christian; Thies, Justus; Nießner, Matthias: Faceforensics++: Learning to detect manipulated facial images. In: IEEE ICCV. pp. 1–11, 2019.

[Sa17] Salimans, Tim; Karpathy, Andrej; Chen, Xi; Kingma, Diederik P: Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. arXiv preprint arXiv:1701.05517, 2017.

[St19] Stehouwer, Joel; Dang, Hao; Liu, Feng; Liu, Xiaoming; Jain, Anil: On the detection of digital face manipulation. arXiv preprint arXiv:1910.01717, 2019.

[Th16] Thies, Justus; Zollhofer, Michael; Stamminger, Marc; Theobalt, Christian; Nießner, Matthias: Face2face: Real-time face capture and reenactment of rgb videos. In: IEEE CVPR. pp. 2387–2395, 2016.

[TZN19] Thies, Justus; Zollhöfer, Michael; Nießner, Matthias: Deferred neural rendering: Image synthesis using neural textures. ACM Transactions on Graphics (TOG), 38(4):1–12, 2019.

[VDOKK16] Van Den Oord, Aäron; Kalchbrenner, Nal; Kavukcuoglu, Koray: Pixel Recurrent Neural Networks. In: ICML - Volume 48. JMLR.org, p. 1747–1756, 2016.

[Ve20] Verdoliva, Luisa: Media Forensics and DeepFakes: an overview. arXiv preprint arXiv:2001.06564, 2020.

[YDF18] Yu, Ning; Davis, Larry P; Fritz, Mario: Attributing fake images to gans: Analyzing fingerprints in generated images. 2018.

[Zh17] Zhou, Peng; Han, Xintong; Morariu, Vlad I; Davis, Larry S: Two-stream neural networks for tampered face detection. In: CVPRW. IEEE, pp. 1831–1839, 2017.

[ZKC19] Zhang, Xu; Karaman, Svebor; Chang, Shih-Fu: Detecting and simulating artifacts in gan fake images. arXiv preprint arXiv:1907.06515, 2019.