Detecting Face Synthesis Using a Concealed Fusion Model[⋆]

Roberto Leyva[1,2], Gregory Epiphaniou[1], Carsten Maple[1], and Victor Sanchez[2]

[1] WMG, University of Warwick, Coventry, UK
[2] Department of Computer Science, University of Warwick, Coventry, UK

Abstract. Face image synthesis is gaining more attention in computer security due to concerns about its potential negative impacts, including those related to fake biometrics. Hence, building models that can detect the synthesized face images is an important challenge to tackle. In this paper, we propose a fusion-based strategy to detect face image synthesis while providing resiliency to several attacks. The proposed strategy uses a late fusion of the outputs computed by several undisclosed models by relying on random polynomial coefficients and exponents to conceal a new feature space. Unlike existing concealing solutions, our strategy requires no quantization, which helps to preserve the feature space. Our experiments reveal that our strategy achieves state-of-the-art performance while providing protection against poisoning, perturbation, backdoor, and reverse model attacks.

Keywords: Face synthesis · Deep Fake · Fusion Models · Biometrics

1 Introduction

Face image-based identification is widely used in many applications, making it an essential component of authentication systems [6]. Face image synthesis poses a problem for many profile-based systems linked to the users' face images, e.g., fake social media accounts and identity fraud. Moreover, existing presentation attacks, e.g., morphological attacks, can be upgraded with the advances of face image synthesis providing the attacker with concealing and extending capabilities as conducting these attacks initially requires real face images.

Face image synthesis has recently evolved drastically in terms of image quality [18]. Hence, it is not only important to detect synthesized face images but also to provide detection models with resiliency to common attacks. To that end, we present an effective strategy to protect a model for fake face image detection while providing competitive performance. Our contributions are as follows:

1. We present a conceal-features fusion strategy to detect fake face images.
2. Our fusion strategy provides resiliency against poisoning, perturbations, backdoor, and reverse model attacks.

[⋆] This work was partly supported by The Alan Turing Institute via Bill & Gates Foundation (INV-001309), Roberto Leyva is also with ITESM, email rleyv@tec.mx

The rest of this paper is organized as follows. In Section 2, we review the most related works. In Section 3, we present the proposed strategy. Section 4 provides experimental results and Section 5 concludes this paper.

2 Related Work

Detection of synthesized imaging data: these methods usually rely on detecting imperfections in any depicted face [14]. Afchar *et al.* [1] propose a Convolutional Neural Network (CNN) based on the InceptionV3 model to detect synthesized videos. Their method requires detecting the location of faces followed by registration, alignment, and scaling. Hsu *et al.* [6] propose a Generative Adversarial Network (GAN)-based detector that requires measuring the contrastive loss given by the GAN discriminator. Marra *et al.* [19] inspect a set of well-established generic models for image tasks, e.g. IV3, DenseNet, Xception, to detect synthesized imaging data. Their work reveals that standard architectures are natively structured for this task. Nataraj *et al.* [20] propose detecting synthesized face images via a set of co-occurrence matrices prior to using a CNN, as such matrices provide a more descriptive input space. Maiabno *et al.* [18] train several existing CNN backbones to detect the synthesis in several color spaces. Their results show that those architectures are very sensitive to color space. Rossler *et al.* [22] propose to perform a series of manipulations to obtain more synthesized faces to train a CNN. Zhang *et al.* [28], by using a GAN-based model, propose learning the synthesis process by solving an image-to-image translation problem. Guarnera *et al.* [5] propose an anspectral analysis of different transformations and intensity domains, which increases the input descriptiveness. Analyzing facial landmarks to detect synthesized face images is proposed by Tolosana *et al.* [26]. Their work suggests that separate models that are fused can detect the synthesis process by separately analyzing the face components, e.g., nose and eyes. This methodology is also supported by the fact that some synthesizing methods can only replace parts of a face instead of generating a whole new face. Local and global matching is explored by Favorskaya [4], however, their method heavily relies on additional features, e.g., those extracted from the background and any artifacts surrounding the face. Fusing models to detect the synthesized videos are explored by Coccomini *et al.* Their method requires analyzing the faces frame-by-frame by using a CNN and a Vision Transformer [3].

Protection of models: Prior work by Jin *et al.* [8] protects a model by quantizing the input samples using the Wavelet transform and random templates. Talreja *et al.* [25] encode face and iris features by using the Reed-Solomon encoder. A separate CNN is used on each source to produce the features to be hashed. Kaur and Khanna [12] propose to randomly project the input features and perform the detection in an alternative feature space following the random slope method. This idea is further investigated in [11] by performing a fusion between random numbers and local features with dimensionality reduction. Early fusion by hashing the product of random templates with biometric features computed by Gabor filters is proposed by [13]. Maneet *et al.* [24] present several strategies to protect models via multi-biometric sources. The authors provide the basis for processing face, finger, hand, and iris information at the sensor, feature, score,

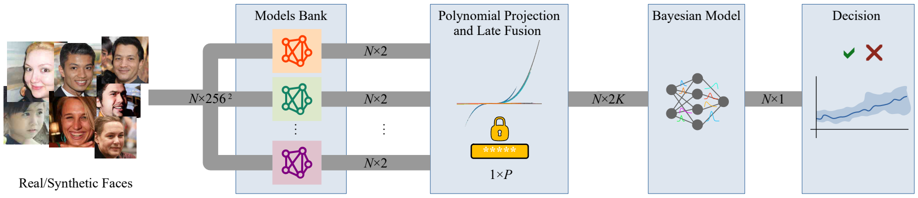


Fig. 1: Our strategy uses a bank of K models. It projects and encrypts the outputs of the decision layer of each individual model to a new feature space. The encrypted projection is used to train a Bayesian model to classify the samples as real or fake.

rank, and decision levels. The authors suggest that strongly protected models should be able to provide encryption at the template level with low distortion of the latent feature space.

3 Proposed Strategy

Existing strategies to protect models [25,11] usually quantize the input space, which inevitably leads to losing important information. The small details are the cornerstone of the state-of-the-art methods in face image synthesis detection [14]. Fusion strategies previously proposed to this end, e.g., [5,26], do not adjust the prediction according to the model posteriors. Such an adjustment can increase the model's security and detection performance simultaneously. However, knowing a priori which parts of the fusion process can boost the detection capabilities is challenging. We consider these aspects to develop our strategy. Specifically, as depicted in Fig. 1, our strategy requires a bank of models to perform late fusion. We protect the posteriors of all models before using a Bayesian model. This model gives the final score to decide if the face is synthetic or not. We explain the constituent components of our strategy next.

Model bank: Following [26], we process the input samples using several models. However, different from [26], we perform no region-based analysis. To this end, we pre-train separately $K = 6$ models and protect the outputs given by their last layer. Let us denote the output of model k by $\hat{x}^k$ for the input image x of size $n_x \times n_y$. The k^{th} model then produces the mapping $\mathbb{R}^{N \times n_x \times n_y} \to \mathbb{R}^{N \times 2}$, for a set of N images and two classes, i.e., real and synthetic. A model bank comprising K models produces the posterior matrix $\hat{X} \in \mathbb{R}^{N \times 2K}$, which we aim to protect. Note that our model bank can comprise any model, including proprietary ones, whose architecture may remain undisclosed [19].

Late fusion: Following the random slope method [12], we propose to protect the decisions of the K models in the bank with random-degree polynomials. Because we can generate polynomials with coefficients and exponents, $\{\alpha_i, \beta_i\} \in \mathbb{Z}$, respectively, we have a fully discrete domain. The proposed fusion is then performed in a discrete rather than a continuous domain, which avoids quantization. Formally, the n^{th} sample $\hat{x}_n$ of the matrix produced by the model bank, i.e., $\hat{X} = \{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n\}$, is mapped by the function $\rho(\cdot)$:

$$\rho(\hat{x}_n \mathbf{Q}^k) = \sum_{i=1 \ldots Q^k} \alpha_i^k \, (\hat{x}_n)^{\beta_i^k} \tag{1a}$$

$$\mathbf{Q}^k = \left\{ \alpha_1^k \dots \alpha_{Q^k}^k, \beta_1^k \dots \beta_{Q^k}^k \right\}, \tag{1b}$$

where $\mathbf{Q}^k \in \mathbb{Z}$ is the set of randomly generated integers used for protection which constitute the key. The design matrix $\hat{X}$ is protected by a vector of size:

$$P = K \sum_k Q_k. \tag{2}$$

The mapping of the projected matrix $\hat{X} \to \hat{X}_\rho$ is then as:

$$\hat{X}_\rho = \left[\rho(\hat{X}, \mathbf{Q}^1), \rho(\hat{X}, \mathbf{Q}^2), \dots \rho(\hat{X}, \mathbf{Q}^k) \right] \tag{3}$$

For instance, if we map $\hat{X}$ with 3-degree polynomials using a bank of $K = 6$ models, we have P as a 36-integer set, where each (α_i, β_i) requires 8 bits, making the key's length equal to $8 \times P$ bits. Breaking such a long key is highly unfeasible using standard computing. Our strategy then *fuses and conceals* the posteriors of all models in the bank without quantization. This aspect adds an authentication-level capability to the inference model. Even if the attacker knows the architecture of the fused models, the key is still required to make predictions and inspect the outputs given by the final decision model.

Bayesian model: Bayesian models have been recently shown to be less prone to overfitting and capable of solving sub-parametrization problems [17]. We then use this model as a binary classifier to predict whether a face image is real or fake. The input to this classifier is the matrix $\hat{X}_\rho$ produced by the late fusion encoding and encryption, thus each input $\rho(\hat{x}) = \hat{x}_\rho$ has a dimension of $2K$. We use two fully connected (FC) layers to calculate the final score. The Bayesian model requires estimating the set of probabilistic parameters $\theta = \{\mu, \varSigma\}$, i.e., means $\{\mu\}$ and variances $\{\varSigma\}$, at each FC layer. Let us consider the target variable t from the vector $\hat{x}_\rho$, whose conditional distribution $p(t|\hat{x})$ is Gaussian[3]. For a neural network model mapping function $f(\hat{x}, w)$, with parameters w, and inverse variance $\varSigma^{-1}$, we have:

$$p(t|\hat{x}, w, \varSigma) = \mathcal{N}\left(t | f(\hat{x}, w), \varSigma^{-1}\right), \tag{4}$$

where $p(w, \mu) = \mathcal{N}(w|0, \mu^{-1}\mathbf{I})$. For N observations of $\hat{X}$ with target values $\mathcal{D} = \{t_1, t_2, \dots t_N\}$, the likelihood function is:

$$p(\mathcal{D}|w, \varSigma) = \prod_{\forall n} \mathcal{N}(t_n | f(\hat{x}_n, w), \varSigma^{-1}). \tag{5}$$

The desired posterior distribution is:

$$p(w|\mathcal{D}, \mu, \varSigma) \approx p(w|\mu)\ p(\mathcal{D}|w, \varSigma). \tag{6}$$

It can be proved that the parameter set given by the MAP estimation is [2]:

$$p(t|\hat{x}, \mathcal{D}, w, \varSigma) = \mathcal{N}\left(t | f(\hat{x}, w_{\mathrm{MAP}}), \sigma^2(\hat{x})\right), \tag{7}$$

where the input-dependent variance σ is given by:

[3] For notation simplicity, we use $\hat{x} = \rho(\hat{x})$.

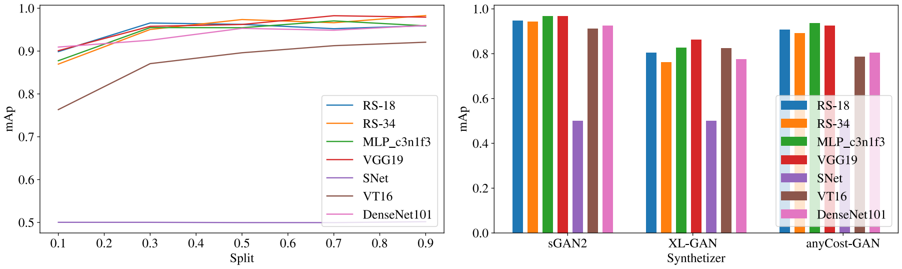


Fig. 2: (left) mAp values for several data splits using the 140K sGAN2 images – the horizontal axis shows the percentage of training data. (right) mAp values on the 140K images of each of the three synthesizers using an 80:20 data split.

$$\sigma^2(\hat{x}) = \Sigma^{-1} + g^\top(\mu\mathbf{I} + \Sigma\ \mathbf{H})^{-1}g, \tag{8a}$$

$$g = \nabla_w f(\hat{x}|w)\big|_{w=w_{\text{MAP}}}, \tag{8b}$$

where $\mathbf{H}$ is the Hessian matrix comprising the second derivatives of the sum of square errors with respect to the components of w. The distribution $p(t|\hat{x}, \mathcal{D})$ is a Gaussian distribution whose mean is given by the neural network function $f(\hat{x}, w_{\text{MAP}})$ and maximizes the posterior likelihood. We can then calculate the model posterior confidence as follows:

$$\max_c\ f(\hat{x}, w_{\text{MAP}}) = t_c, \tag{9}$$

where c denotes the two classes, i.e., real or fake. Since for each $\hat{x}$, $t_c \gg t_{\neq c}$, we can then use this result to set the model confidence and make predictions.

4 Experiments

We use the FFHQ [9] dataset, which comprises 70K real samples, to produce 70K fake samples using three different synthesizers: sGAN2 [10][4], XL-GAN [23][5], and anyCost-GAN [15][6]. Hence, we have 70K+70K=140K samples for each synthesizer. The Bayesian model fuses models that are pre-trained separately on each set of 140K images. We train the Bayesian model using a scheduler to detect error plateaus and scale the learning rate accordingly by a power of ten.

Detection accuracy: We first evaluate several detection models separately, i.e., with no fusion, in terms of the mean Average precision (mAp) for several data splits, where the training and test datasets contain equal proportions of fake and real images. This first experiment allows us to select the $K = 6$ models to be used in the model bank of our strategy. Note that one of the evaluated models is a CNN-Multi-Layer Perceptron (MLP) model we propose with pooling and only expanding convolutional filters to capture small artifacts commonly present in synthesized face images, hereinafter called the MLP_c3n1f3 model (see Appendix A). Fig. 2(a) shows mAp values for several detection models and data splits on the 140K sGAN2 images. Note that most of the evaluated models

[4] https://github.com/NVlabs/stylegan
[5] https://github.com/autonomousvision/stylegan-xl
[6] https://github.com/mit-han-lab/anycost-gan

Table 1: mAp values (↑) of several detection models on the 140K images of each of the three synthesizers using several data splits.

Model	Synthesizer	Data split (proportion of training data)								
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
VGG-19†	sGAN2 [10]	0.903	0.922	0.941	0.954	0.968	0.952	0.953	0.962	0.961
	XL-GAN [23]	0.771	0.810	0.856	0.889	0.915	0.917	0.921	0.928	0.918
	anyCost-GAN [15]	0.851	0.874	0.912	0.914	0.917	0.927	0.923	0.932	0.935
DF [18]‡	sGAN2	0.875	0.895	0.913	0.931	0.935	0.942	0.946	0.945	0.949
	XL-GAN	0.802	0.856	0.899	0.901	0.908	0.912	0.917	0.916	0.915
	anyCost-GAN	0.833	0.854	0.911	0.912	0.917	0.931	0.945	0.948	0.944
CoMat [20]	sGAN2	0.901	0.934	0.956	0.954	**0.964**	0.968	0.958	0.969	0.978
	XL-GAN	0.834	0.876	0.915	0.926	0.924	0.938	0.941	0.948	0.952
	anyCost-GAN	0.831	0.884	0.932	**0.955**	0.945	0.954	0.958	0.952	0.962
Bayesian fusion (ours)	sGAN2	**0.913**	**0.940**	**0.964**	0.954	0.951	**0.972**	**0.987**	**0.988**	0.982
	XL-GAN	0.832	0.878	0.925	0.952	0.945	0.947	0.967	0.965	0.968
	anyCost-GAN	0.871	0.910	0.951	0.961	0.944	0.957	0.968	0.974	**0.988**

†Best performing model based on the first experiment. ‡Only compared in the RGB space.

require about 30% of the training data to achieve competitive accuracy on these images. Fig. 2(b) shows mAp values for several detection models on the 140K images of each of the three synthesizers using an 80:20 data split. We can see that VGG-19 achieves the best performance. For the case of the 140K anyCost-GAN images, our MLP_c3n1f3 model outperforms VGG-19.

Next, we fuse the six best-performing models from the previous experiment using our proposed strategy. Table 1 tabulates mAp values of our strategy (Bayesian fusion) and other state-of-the-art methods, including the best-performing model in the previous experiment. i.e., VGG-19. The tabulated results are for several data splits and the 140K images of each of the three synthesizers. The proposed strategy attains competitive accuracy even for small data split values. Hence, it requires fewer training samples to perform very well.

Ablation studies: We analyze the posterior confidence as the loss value declines during the training of the Bayesian model used by our strategy (see Fig. 3 (left)). In general, the strategy makes fewer errors when it is more confident. Since we perform a non-linear mapping, the new feature space may not be as descriptive as the original one. We then evaluate our strategy's accuracy (mAp) for several key lengths, especially because we observe that the strategy may not converge to a high mAp value when using long keys. Fig. 3 (right) shows the effect of using long keys in terms of the number of training attempts needed for our strategy to converge as the key length increases. We observe that a key of length 36 easily makes the model converge in the first attempt.

Model attacks: We measure the success rate of the poisoning, perturbation, reverse, and backdoor attacks as performed on our strategy using the sGAN2 images. In other words, we measure the success of miss-detecting samples that are correctly detected before the attack.

Poisoning: We swap the labels in the training dataset to generate wrong detections by using several infection proportions, i.e., the ratio of swapped labels and the total number of samples [16]. Fig. 4a shows that as the infection proportion increases, the success rate increases but the accuracy during the training

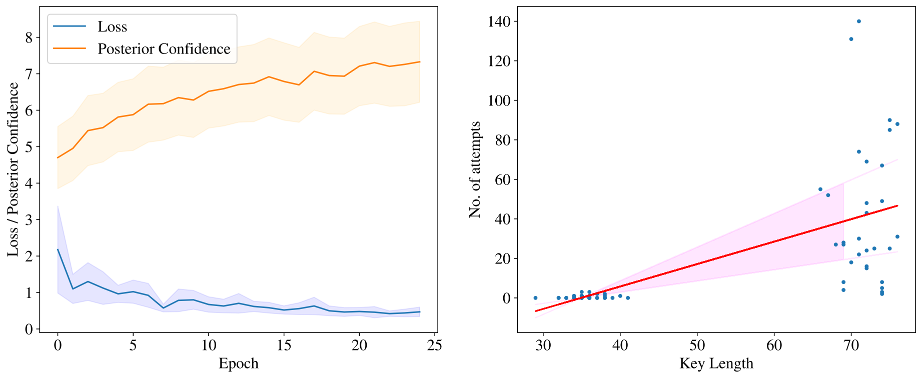
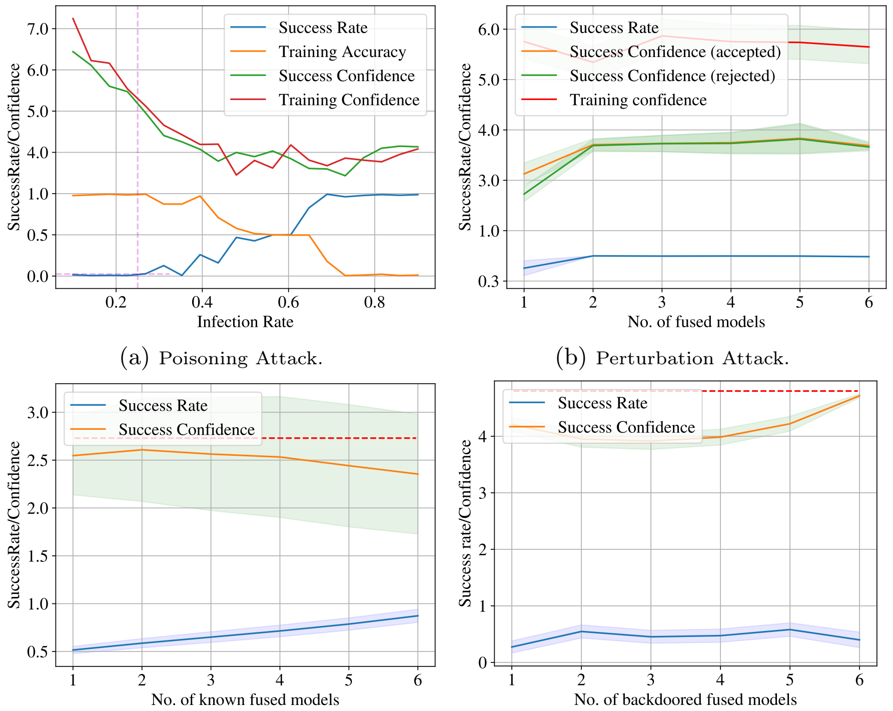


Fig. 3: (left) Loss and posterior confidence values during training. (right) Number of attempts needed for the Bayesian fusion model to converge for several key lengths.



(a) Poisoning Attack.

(b) Perturbation Attack.

(c) Reverse Model Attack.

(d) Backdoor Attack.

Fig. 4: Success rate of several attacks against the proposed Bayesian fusion strategy.

decreases. We observe that the attack is most dangerous when 20%∼30% of the labels are poisoned. In such a case, the training accuracy and confidence are high and the model is cheated in ∼2% of the testing samples.

Perturbation: We corrupt samples by adding noise and blurring them [21]. Fig. 4b shows the results as the number of fused models increases. Despite a very high success rate, the confidence is low compared to the training confidence. Therefore, this attack can be easily detected by inspecting the model's posterior.

Reverse Model: This attack is via manipulating the decision layers to make the model fail with specific samples, for example, by feature vector angle deformation or weight surgery without retraining [27]. We assume the attacker knows some of the fused models. Fig. 4c shows that the attacker requires knowledge of the majority of the fused models to succeed, i.e., when the *Training confidence* curve, represented as the dotted red line, is reached. The attack have high success confidence even when only one of the fused models is known.

Backdoor: We mark samples using a black patch following [7], to maliciously change the classification result. Fig. 4d shows how the success confidence increases as more fused models are attacked. Although the success rate remains almost constant, as the number of attacked models increases, it is likely that the strategy miss-classifies the marked samples as intended by the attacker because the success confidence increases. About 5% of the marked samples are miss-classified as intended when all fused models are attacked, see the *Success Confidence* curve reaching the *Training confidence* curve (red dotted line).

Note that although our strategy is robust by design to poisoning, backdoor, and perturbation attacks, reverse model attacks pose an important threat, which can be mitigated by not disclosing the architecture.

5 Conclusion

We have proposed a strategy based on fusion to provide concealment of a model trained to detect synthesized face images while simultaneously increasing accuracy when fewer training samples are available. The proposed strategy projects and encrypts the output of the decision layers of several models into a new feature space. Our proposed strategy is simple yet effective and achieves very competitive accuracy. Our findings have the potential to help protect models used for face validation while providing resiliency to common attacks. Future work focuses on cross-dataset evaluations and robustness against more sophisticated attacks, e.g., backdoor injection, adversarial patches, and weight surgery.

References

1. Afchar, D., Nozick, V., Yamagishi, J., Echizen, I.: Mesonet: a compact facial video forgery detection network. In: 2018 IEEE international workshop on information forensics and security (WIFS). pp. 1–7. IEEE (2018)
2. Bishop, C.M., Nasrabadi, N.M.: Pattern recognition and machine learning, vol. 4. Springer (2006)
3. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
4. Favorskaya, M., Yakimchuk, A.: Fake face image detection using deep learning-based local and global matching. CEUR Workshop Proceedings (2021)
5. Guarnera, L., Giudice, O., Nastasi, C., Battiato, S.: Preliminary forensics analysis of deepfake images. In: 2020 AEIT international annual conference (AEIT). pp. 1–6. IEEE (2020)

6. Hsu, C.C., Lee, C.Y., Zhuang, Y.X.: Learning to detect fake face images in the wild. In: 2018 international symposium on computer, consumer and control (IS3C). pp. 388–391. IEEE (2018)
7. Huang, S., Peng, W., Jia, Z., Tu, Z.: One-pixel signature: Characterizing cnn models for backdoor detection. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Computer Vision – ECCV 2020. pp. 326–341. Springer International Publishing, Cham (2020)
8. Jin, A.T.B., Ling, D.N.C., Goh, A.: Biohashing: two factor authentication featuring fingerprint data and tokenised random number. Pattern recognition **37**(11), 2245–2255 (2004)
9. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196 (2017)
10. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 1–1 (2020)
11. Kaur, H., Khanna, P.: Random distance method for generating unimodal and multimodal cancelable biometric features. IEEE Transactions on Information Forensics and Security **14**(3), 709–719 (2018)
12. Kaur, H., Khanna, P.: Random slope method for generation of cancelable biometric features. Pattern Recognition Letters **126**, 31–40 (2019), robustness, Security and Regulation Aspects in Current Biometric Systems
13. Kaur, H., Khanna, P.: Privacy preserving remote multi-server biometric authentication using cancelable biometrics and secret sharing. Future Generation Computer Systems **102**, 30–41 (2020)
14. Khodabakhsh, A., Ramachandra, R., Raja, K., Wasnik, P., Busch, C.: Fake face detection methods: Can they be generalized? In: 2018 International Conference of the Biometrics Special Interest Group (BIOSIG). pp. 1–6 (Sep 2018)
15. Lin, J., Zhang, R., Ganz, F., Han, S., Zhu, J.Y.: Anycost gans for interactive image synthesis and editing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14986–14996 (2021)
16. Liu, Y., Ma, X., Bailey, J., Lu, F.: Reflection backdoor: A natural backdoor attack on deep neural networks. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Computer Vision – ECCV 2020. pp. 182–199. Springer International Publishing, Cham (2020)
17. Lotfi, S., Izmailov, P., Benton, G., Goldblum, M., Wilson, A.G.: Bayesian model selection, the marginal likelihood, and generalization. In: International Conference on Machine Learning. pp. 14223–14247. PMLR (2022)
18. Maiano, L., Papa, L., Vocaj, K., Amerini, I.: Depthfake: a depth-based strategy for detecting deepfake videos. arXiv preprint arXiv:2208.11074 (2022)
19. Marra, F., Gragnaniello, D., Cozzolino, D., Verdoliva, L.: Detection of gan-generated fake images over social networks. In: 2018 IEEE conference on multimedia information processing and retrieval (MIPR). pp. 384–389. IEEE (2018)
20. Nataraj, L., Mohammed, T.M., Chandrasekaran, S., Flenner, A., Bappy, J.H., Roy-Chowdhury, A.K., Manjunath, B.: Detecting gan generated fake images using co-occurrence matrices. arXiv preprint arXiv:1903.06836 (2019)
21. Ning, R., Li, J., Xin, C., Wu, H.: Invisible poison: A blackbox clean label backdoor attack to deep neural networks. In: IEEE INFOCOM 2021 - IEEE Conference on Computer Communications. pp. 1–10 (May 2021)
22. Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., Nießner, M.: Faceforensics++: Learning to detect manipulated facial images. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1–11 (2019)

23. Sauer, A., Schwarz, K., Geiger, A.: Stylegan-xl: Scaling stylegan to large diverse datasets. In: ACM SIGGRAPH 2022 conference proceedings. pp. 1–10 (2022)
24. Singh, M., Singh, R., Ross, A.: A comprehensive overview of biometric fusion. Information Fusion **52**, 187–205 (2019)
25. Talreja, V., Valenti, M.C., Nasrabadi, N.M.: Multibiometric secure system based on deep learning. In: 2017 IEEE Global conference on signal and information processing (globalSIP). pp. 298–302. IEEE (2017)
26. Tolosana, R., Romero-Tapiador, S., Fierrez, J., Vera-Rodriguez, R.: Deepfakes evolution: Analysis of facial regions and fake detection performance. In: Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part V. pp. 442–456. Springer (2021)
27. Zehavi, I., Shamir, A.: Facial misrecognition systems: Simple weight manipulations force dnns to err only on specific persons. arXiv preprint arXiv:2301.03118 (2023)
28. Zhang, X., Karaman, S., Chang, S.F.: Detecting and simulating artifacts in gan fake images. In: 2019 IEEE international workshop on information forensics and security (WIFS). pp. 1–6. IEEE (2019)

A Proposed MLP Detector

We propose a vanilla-style model (see Table 2) to detect small artifacts found in synthetic face images (see Fig. 5). To this end, our model uses progressively larger filter banks, performs a unitary stride with the convolutional operations, and does not perform pooling operations in the feature extraction stage as these operations discard important information.

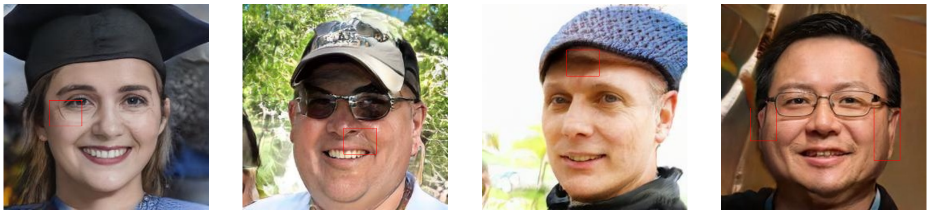


Fig. 5: Artifacts in synthetic face images (left to right): a large scar below the eye, a scar below the nose, a dented forehead, and uneven ears.

Table 2: Architecture of the proposed MLP detector.

Layer Type	Number of Filters	Feature Map Size	Kernel Size	Stride	Activation
Input		$256^2 \times 3$			
Center-Crop		$224^2 \times 3$			
Convolution Layer 1^{st}	16	$224^2 \times 16$	5×5	1×1	ReLu
Convolution Layer 2^{nd}	24	$220^2 \times 24$	5×5	1×1	ReLu
Convolution Layer 3^{th}	32	$216^2 \times 32$	5×5	1×1	ReLu
Batch Normalization					
Fully Connected Layer 1^{st}		$212^2 \times 32 \rightarrow 120$			
Fully Connected Layer 2^{nd}		$120 \rightarrow 84$			
Fully Connected Layer 3^{th}		$84 \rightarrow 2$			