A-PIXELHOP: A GREEN, ROBUST AND EXPLAINABLE FAKE-IMAGE DETECTOR

Yao Zhu[1], Xinyu Wang[1], Hong-Shuo Chen[1], Ronald Salloum[2], C.-C. Jay Kuo[1]

Ming Hsieh Department of Electrical Engineering, University of Southern California[1]
School of Computer Science and Engineering, California State University, San Bernardino[2]

ABSTRACT

A novel method for detecting CNN-generated images, called Attentive PixelHop (or A-PixelHop), is proposed in this work. It has three advantages: 1) low computational complexity and a small model size, 2) high detection performance against a wide range of generative models, and 3) mathematical transparency. A-PixelHop is designed under the assumption that it is difficult to synthesize high-quality, high-frequency components in local regions. It contains four building modules: 1) selecting edge/texture blocks that contain significant high-frequency components, 2) applying multiple filter banks to them to obtain rich sets of spatial-spectral responses as features, 3) feeding features to multiple binary classifiers to obtain a set of soft decisions, 4) developing an effective ensemble scheme to fuse the soft decisions into the final decision. Experimental results show that A-PixelHop outperforms state-of-the-art methods in detecting CycleGAN-generated images. Furthermore, it can generalize well to unseen generative models and datasets.

Index Terms— image forensics, fake-image detection, neural networks, generative models

1. INTRODUCTION

In recent years, there has been a rapid development of image-synthesis techniques based on convolutional neural networks (CNNs), such as generative adversarial networks (GANs) [1]. Such techniques have demonstrated the ability to generate high-quality fake images, and as a result, have raised concerns that it will become increasingly challenging to distinguish fake (or synthetic) and real (or authentic) images. Determining whether an image was synthesized by a specific CNN-based architecture is relatively straightforward. This can be accomplished by training a classifier using real and fake images generated by the specific CNN-based architecture. However, there exist many different fake-image generators, and thus, it is essential to develop a generic detection method that can generalize well to fake images generated by a wide range of generative models. This is the objective of our current research.

Most state-of-the-art methods for detection of CNN-generated images are based on deep neural networks. Different architectures have proven to be effective in detecting fake images. However, deep-learning-based detection methods need an enormous amount of data to maintain good performance. Because of the rapid evolution of image-synthesis techniques, training datasets from multiple generative models and/or extensive data augmentation are needed in order for these detection methods to generalize well to unseen generative models.

In contrast to deep-learning-based methods, a novel detector based on signal processing, called Attentive PixelHop (or A-PixelHop), is proposed in this work. It has three key characteristics: low computational and memory complexity (i.e., green), high detection performance against a wide range of generative models (i.e., robustness), and mathematical transparency (i.e., explainability). Its design is based on the assumption that high-quality, high-frequency components in local regions are more difficult to generate.

A-PixelHop has four building modules. Its first module selects edge/texture blocks that contain significant high-frequency components. Its second module applies multiple filter banks to them to obtain rich sets of spatial-spectral responses as features. Its third module feeds features to multiple binary classifiers to obtain a set of soft decisions. Its last module adopts an effective ensemble scheme to fuse the soft decisions into the final decision. It is demonstrated by experimental results that A-PixelHop outperforms state-of-the-art methods for CycleGAN-generated images. Furthermore, it is demonstrated that A-PixelHop can generalize well to unseen generative models and datasets.

2. RELATED WORK

2.1. Detecting CNN-generated Images

In this paragraph, we provide a brief summary of existing methods for detection of CNN-generated images. Inspired by solutions in steganalysis, Cozzolino *et al.* [2] proposed a CNN architecture that mimics the rich models in feature extraction and classification. Zhang *et al.* [3] proposed to feed spectral input (rather than pixel input) to a classifier. They also introduced a GAN simulator, called AutoGAN, that simulates artifacts produced by popular GAN models. Recently, Wang *et al.* [4] trained a classifier with a large number of ProGAN-generated images and evaluated it on images synthesized by eleven different generators. Their work showed the effectiveness of extensive data augmentation in improving the generalization ability of a classifier.

2.2. Subspace Learning

Our work follows the methodology of subspace learning. Although subspace learning has a long history, Kuo *et al.* [5] recently built a link between subspace learning and the convolutional operations in CNNs. A set of convolutional filters in a given convolutional layer of a CNN can be interpreted as a set of filters in one filter bank. The filter parameters in CNNs are obtained via end-to-end optimization through back-propagation. However, in subspace learning, filter parameters are derived by the statistical analysis of pixel correlations inside a local region covered by the filter. For example, for a filter of size $5 \times 5 \times 3$, where 5×5 corresponds to the spatial window size and 3 corresponds to the three color channels, we would examine the correlations between the $5 \cdot 5 \cdot 3 = 75$ pixels. A variant of principal

978-1-6654-0540-9/22/$31.00 ©2022 IEEE

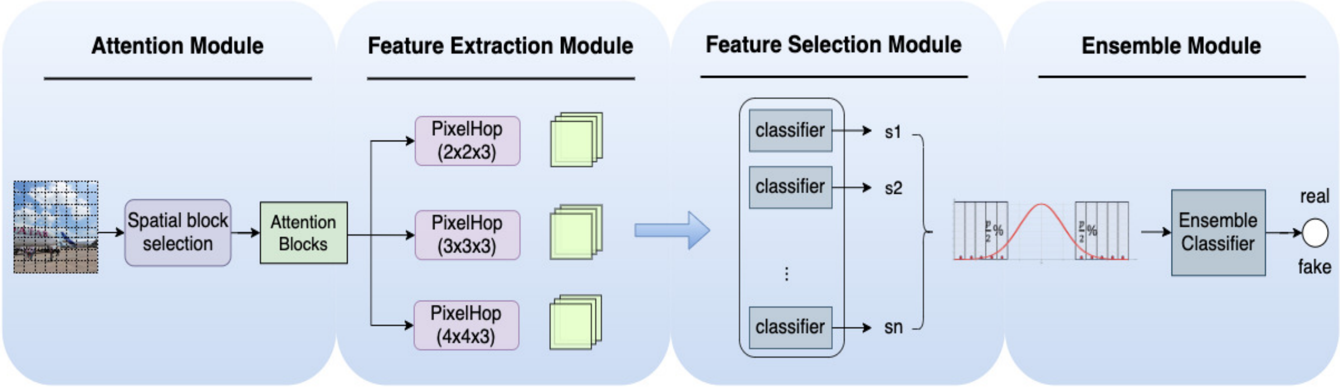


Fig. 1. The block-diagram of the A-PixelHop method.

component analysis (PCA), called the Saab (Subspace approximation via adjusted bias) transform, was introduced in [5] and used to determine filter parameters. The concept of multiple convolutional layers can be ported to subspace learning, leading to successive subspace learning (SSL), and the corresponding architecture is called the PixelHop [6]. PixelHop offers an unsupervised and feedforward feature learning process. Neither back-propagation nor labels are needed in deriving filter parameters. SSL has been applied to image classification [6, 7] and 3D point cloud classification [8, 9], among many others. In image forensics, DefakeHop [10] was proposed to detect deepfake videos.

It is worthwhile to emphasize that our work is different from DefakeHop in two main aspects. First, DefakeHop used facial landmarks to crop out eyes, nose and mouth regions and perform detection in each region. However, in this work, we need to consider generic fake images and cannot rely on the special facial regions here. Second, DefakeHop leveraged cascaded PixelHop units; it belongs to the category of SSL. On the other hand, our work adopts multiple single-stage PixelHop units in parallel, and thus, belongs to the category of parallel subspace learning (PSL).

3. PROPOSED A-PIXELHOP METHOD

An overview of the proposed A-PixelHop method is given in Fig. 1. It takes authentic or CNN-generated fake images as input and generates a binary decision - true or fake. Its four modules are elaborated below.

3.1. Spatial-Block Selection

An image is first partitioned into non-overlapping blocks of size 16×16. Under the assumption that it is more difficult for CNN-based generators to synthesize high-frequency components in images, the spatial attention module is developed to select blocks that contain complex and/or fine details. There are many ways to implement this idea. Here, for each block, we calculate its pixel-domain variance for the R,G,B channels and add them up to yield the block variance. We then select the blocks with the largest variances. These blocks typically correspond to edge/texture regions in images. Examples of selected spatial regions are shown in Fig. 2.

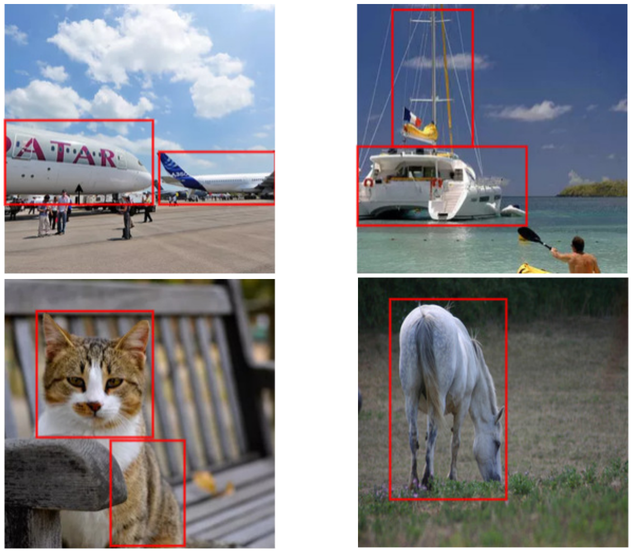


Fig. 2. Illustration of selected spatial regions that contain complex and fine details.

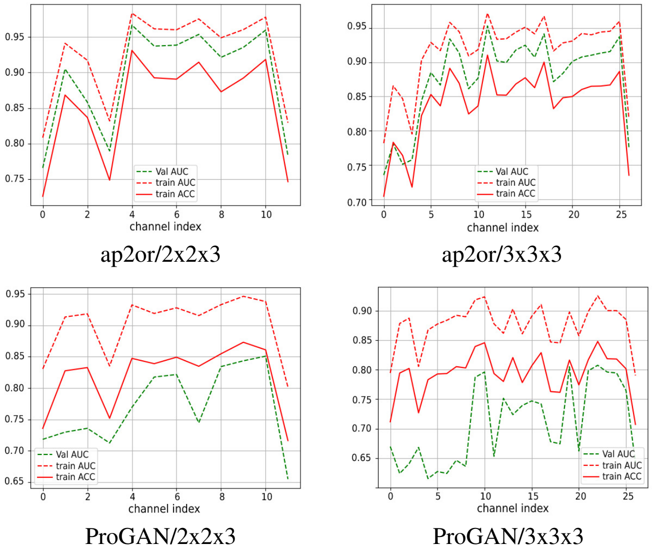
3.2. Parallel PixelHops

A PixelHop unit consists of a set of filters of the same size that operate on all pixels in a block in parallel. A filter is a 3D tensor of size $s \times s \times c$, where $s \times s$ are the spatial dimensions and c is the spectral dimension. Typically, $s = 2, 3, 4$ and $c = 3$ for color images. We employ multiple PixelHop units in parallel for feature extraction to increase feature diversity. Filter weights are determined by a variant of PCA called the Saab transform [5]. As shown in Fig. 1, we use three parallel PixelHop units for feature extraction, where the filters are of sizes $2 \times 2 \times 3$, $3 \times 3 \times 3$ and $4 \times 4 \times 3$, respectively. For filters of size $s \times s \times c$, there are s^2c channels. We use a stride of 1 so that each channel has $(16 - s + 1)^2 = (17 - s)^2$ spatial responses.

3.3. Classification and Discriminant Channel Selection

We use channel-wise spatial responses to train an XGBoost classifier [11] to select discriminant channnels. The channel-wise classification performance curves measured by the area-under-the-curve (AUC) and the accuracy (ACC) for the apple vs. orange subset in the

CycleGAN [12] and ProGAN [13] datasets are shown in Fig. 3. We observe a consistent trend between the validation AUC and training AUC curves and can select a couple of discriminant channels based on the peaks of the validation AUC curves. In the experiments, we select two and three optimal filters from each PixelHop unit for CycleGAN and ProGAN, respectively. As a result, we have six and nine discriminant filters for CycleGAN and ProGAN, respectively, and use their soft decision scores from the XGBoost classifier as features for the image-level decision ensemble.



ap2or/2x2x3

ap2or/3x3x3

ProGAN/2x2x3

ProGAN/3x3x3

Fig. 3. AUC/ACC performance of channel-wise classification using filters of size 2x2x3 and 3x3x3, for the apple-orange subset under CycleGAN (top two rows) and ProGAN (bottom two rows).

3.4. Image-Level Ensemble

A-PixelHop makes the final image-level decision by ensembling block-level soft decisions. We first sort the soft decision scores of discriminant channels of various spatial blocks from the smallest to the largest values in the unit interval. They form a distribution as indicated by the red curve in Fig. 4. Soft decisions close to the center are not as discriminant as those lying at the two ends. We sample soft decision values from the two ends of the distribution since the two ends indicate higher probabilities of being real or fake. As shown in Fig. 4, representative soft decisions (denoted by red dots) are selected. The sampling percentage from each end (given by $0.5p\%$) is a hyperparameter, which is chosen based on the validation dataset. As a result, $p\%$ of representative soft decisions are selected and form a feature vector, which is fed to the image-level ensemble classifier. Based on the hyperparameter optimization, we found that typical values for p are 10, 20, and 30. For images of size 256×256, it is fine to simply select the top and bottom $0.5p\%$ soft decisions without sampling. However, for images of higher resolution (e.g. 3000×4000), the number of discriminant blocks is very large if $10 \leq p \leq 30$. On the other hand, if we set p to a small value (say, $p = 1$), the selected samples are likely to be outliers and they are not representative enough. Thus, a sampling scheme offers a good balance between representation and discrimination.

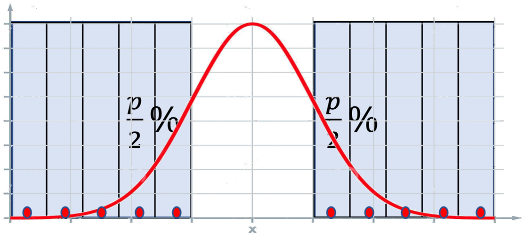


Fig. 4. Image-level ensemble based on soft decisions selected from two ends of the distribution.

4. EXPERIMENTS

In this section, we compare the performance of our proposed method with several existing methods in two different experimental setups.

4.1. Experiment I

In the first experiment, we utilize a dataset consisting of 10 subsets, where each subset contains both real and CycleGAN-generated images [3, 12, 14]. For example, the *horse2zebra* subset includes real horse and zebra images for training CycleGAN and corresponding fake horse and zebra images generated from the trained model. It has 14 semantic categories, including Apple, Orange, Horse, Zebra, Yosemite summer, Yosemite winter, Facades, CityScape_Photo, Satellite Image, Ukiyoe, Van Gogh, Cezanne, Monet and Photo. There are over 36K images in this dataset.

We compared the proposed A-PixelHop method with several state-of-the-art methods, including Cozzolino2017 [2] and Auto-GAN with spectral input (Auto-Spec) [3]. Here, we follow the leave-one-out setting described in [3, 14], where one subset is set aside for testing while the other nine subsets are used in the training process. Table 1 shows the test accuracy of the proposed method as well as the existing methods. We see that A-PixelHop reaches 100% accuracy for five (out of ten) subsets. Its average accuracy over the 10 subsets is 98.7%, which is best among all methods.

4.2. Experiment II

This second experimental setup was utilized in [4] in order to evaluate how well a given detection method generalizes to unseen generative models. The dataset used here contains images synthesized by a wide variety of generative models. All of them have an upsampling-convolutional structure. In the training set, fake images from 20 object categories are generated by the ProGAN model only. There are 720K real/fake image pairs in the training set and 4K images in the validation set. In the testing set, fake images are generated by the following eleven models: ProGAN [13], StyleGAN [15], BigGAN [16], CycleGAN [12], StarGAN [17], GauGAN [18], CRN [19], IMLE [20], SITD [21], SAN [22], and Deepfake [23].

In this experiment, we follow the procedure specified in [4], by first training A-PixelHop with real and ProGAN-generated fake images, and then evaluating its detection performance on real images or fake images generated by the aforementioned eleven generative models. The performance comparison between A-PixelHop and two existing methods, Auto-Spec and the method proposed by Wang *et al.* in [4], is shown in Table 2. It is worthwhile to emphasize three points. First, since we do not include augmentation in the training of A-PixelHop, we compare against [4] under the no augmentation setting. Second, we evaluate the performance in terms of average

Methods	ap2or	ho2zeb	win2sum	citysc.	facades	map2sat	Ukiyoe	VanGogh	Cezanne	Monet	ave.
DenseNet	79.1	95.8	67.7	93.8	99.0	78.3	99.5	97.7	99.9	89.8	89.2
XceptionNet	95.9	99.2	76.7	100.0	98.6	76.8	100.0	99.9	100.0	95.1	94.5
InceptionNet	85.0	94.8	58.8	99.4	94.0	70.5	99.8	98.8	99.9	89.9	89.1
Cozzolino2017	99.9	99.9	61.2	99.9	97.3	99.6	100.0	99.9	100.0	99.2	95.1
Auto-Spec	98.3	98.4	93.3	100.0	100.0	78.6	99.9	97.5	99.2	99.7	97.2
Ours	99.2	99.8	100.0	94.4	100.0	94.1	100.0	100.0	100.0	99.4	**98.7**

Table 1. Comparison of test accuracy of fake-image detectors against 10 CycleGAN subsets in Experiment I. Performance numbers for DenseNet, XceptionNet, InceptionNet and Cozzolino2017 are taken from [14]. **Boldface** is used to indicate best performance.

Methods	ProGAN	StyleGAN	BigGAN	CycleGAN	StarGAN	GauGAN	CRN	IMLE	SITD	SAN	Deepfake	mAP
Auto-Spec	75.6	68.6	84.9	100.0	100.0	61.0	80.8	75.3	89.9	66.1	39.0	76.5
Wang *et al.*	100.	96.3	72.2	84.0	100.	67.0	93.5	90.3	96.2	93.6	98.2	90.1
Ours (10%)	99.9	99.9	77.1	97.8	100.0	94.6	76.4	92.8	76.5	84.4	94.5	90.4
Ours (20%)	99.9	99.9	75.2	97.3	100.0	94.7	86.8	95.3	76.5	83.8	93.3	91.2
Ours (30%)	99.9	99.9	74.4	96.7	99.9	94.8	91.3	98.2	76.7	80.3	91.3	**91.2**

Table 2. Comparison of the average precision of fake-image detectors against eleven generative models in Experiment II. **Boldface** is used to indicate best performance.

precision (AP) so as to be consistent with [4]. Third, we collect a total of 10%, 20% and 30% of samples from the two ends for the image-level ensemble and show the corresponding mean AP (mAP) values. We see from the table that A-PixelHop outperforms both Auto-Spec and the method from [4] in all three cases (i.e., 10%, 20% and 30% of samples) in terms of mAP. A-PixelHop outperforms both Auto-Spec and the method from [4] by a large margin in the case of Gau-GAN. A-PixelHop performs worse in the case of Big-GAN, SITD and SAN, indicating a weaker transferability from ProGAN to these three generative models. It demands further exploration. SITD and SAN generate images of very high resolution (e.g., 3Kx4K pixels), which does not match well with that of the training images generated by ProGAN. One possible fix is to rescale these large images to smaller ones in the pre-processing step.

Based on Experiments I and II, we conclude that the proposed A-PixelHop method is robust in the sense that it generalizes well to different semantic categories as well as unseen generative models. In addition, it offers state-of-the-art detection performance.

4.3. Model Size Computation

We compare the model size (in terms of number of parameters) of the proposed A-PixelHop method with that of the other methods in Table 3. Auto-Spec and the method from [4] utilized Resnet34 and Resnet50, respectively. The method from [2] used a light weight CNN that has two convolutional layers and one fully connected layer. The model has 1K parameters. A-PixelHop has different model sizes in Experiments I and II. As shown in Table 4, it selects 6 and 9 discriminant channels in Experiments I and II, respectively, and trains one XGBoost classifier for each channel. Each XGBoost classifer has 100 trees with a maximum depth of 6 and has a model size of 19K. Furthermore, it trains an ensemble XGBoost classifier that has 10 trees with depth equal to one. Its model size is 40.

Exp.	Ours	Auto-Spec	Cozzolino2017	Wang *et al.*
I	114K	21.8M	1K	–
II	171K	21.8M	–	25.6M

Table 3. Model size comparison (in terms of number of parameters).

components	para #	components	para #
2 (2x2x3)	24	3 (2x2x3)	36
2 (3x3x3)	54	3 (3x3x3)	81
2 (4x4x3)	96	3 (4x4x3)	144
6 XGBoost	6x19K	9 XGBoost	9x19K
1 XGBoost	40	1 XGBoost	40
Total	114K	Total	171K

Table 4. Model size computation for A-PixelHop for Experiment I (left) and II (right).

5. CONCLUSION AND FUTURE WORK

A green, robust, high-performance and explainable method, called A-PixelHop, to detect CNN-generated fake images was presented in this work. A-PixelHop used the filter-bank signal processing tool to extract discriminant joint spatial-spectral components as features and fed them to the XGBoost classifer to derive the block-level decision. Finally, it adopted an ensemble learning tool to fuse multiple block-level soft decisions to obtain the final image-level decision. The superior performance of A-PixelHop was demonstrated by experimental results. As future extension, we plan to apply A-PixelHop to distinguish real/fake images that are manipulated by other operations such as compression, blurring, additive noise, etc.

6. REFERENCES

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[2] Davide Cozzolino, Giovanni Poggi, and Luisa Verdoliva, "Recasting residual-based local descriptors as convolutional neural networks: an application to image forgery detection," in *Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security*, 2017, pp. 159–164.

[3] Xu Zhang, Svebor Karaman, and Shih-Fu Chang, "Detecting and simulating artifacts in gan fake images," in *2019 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2019, pp. 1–6.

[4] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros, "Cnn-generated images are surprisingly easy to spot... for now," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8695–8704.

[5] C-C Jay Kuo, Min Zhang, Siyang Li, Jiali Duan, and Yueru Chen, "Interpretable convolutional neural networks via feedforward design," *Journal of Visual Communication and Image Representation*, vol. 60, pp. 346–359, 2019.

[6] Yueru Chen and C-C Jay Kuo, "Pixelhop: A successive subspace learning (ssl) method for object recognition," *Journal of Visual Communication and Image Representation*, vol. 70, pp. 102749, 2020.

[7] Yueru Chen, Mozhdeh Rouhsedaghat, Suya You, Raghuveer Rao, and C-C Jay Kuo, "Pixelhop++: A small successive-subspace-learning-based (ssl-based) model for image classification," in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 3294–3298.

[8] Min Zhang, Haoxuan You, Pranav Kadam, Shan Liu, and C-C Jay Kuo, "Pointhop: An explainable machine learning method for point cloud classification," *IEEE Transactions on Multimedia*, vol. 22, no. 7, pp. 1744–1755, 2020.

[9] Min Zhang, Yifan Wang, Pranav Kadam, Shan Liu, and C-C Jay Kuo, "Pointhop++: A lightweight learning model on point sets for 3d classification," in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 3319–3323.

[10] Hong-Shuo Chen, Mozhdeh Rouhsedaghat, Hamza Ghani, Shuowen Hu, Suya You, and C-C Jay Kuo, "Defakehop: A light-weight high-performance deepfake detector," in *2021 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2021, pp. 1–6.

[11] Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, et al., "Xgboost: extreme gradient boosting," *R package version 0.4-2*, vol. 1, no. 4, pp. 1–4, 2015.

[12] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[13] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *arXiv preprint arXiv:1710.10196*, 2017.

[14] Francesco Marra, Diego Gragnaniello, Davide Cozzolino, and Luisa Verdoliva, "Detection of gan-generated fake images over social networks," in *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*. IEEE, 2018, pp. 384–389.

[15] Tero Karras, Samuli Laine, and Timo Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4401–4410.

[16] Andrew Brock, Jeff Donahue, and Karen Simonyan, "Large scale gan training for high fidelity natural image synthesis," *arXiv preprint arXiv:1809.11096*, 2018.

[17] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8789–8797.

[18] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2337–2346.

[19] Qifeng Chen and Vladlen Koltun, "Photographic image synthesis with cascaded refinement networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1511–1520.

[20] Ke Li, Tianhao Zhang, and Jitendra Malik, "Diverse image synthesis from semantic layouts via conditional imle," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4220–4229.

[21] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun, "Learning to see in the dark," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3291–3300.

[22] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang, "Second-order attention network for single image super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11065–11074.

[23] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner, "Faceforensics++: Learning to detect manipulated facial images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1–11.