https://doi.org/10.2352/EI.2023.35.4.MWSF-380
© 2023, Society for Imaging Science and Technology

Detecting GAN-Generated Synthetic Images Using Semantic Inconsistencies

Danial Samadi Vahdati, Philadelphia,PA, Matthew C. Stamm, Philadelphia, PA

Abstract

In the past several years, generative adversarial networks have emerged that are capable of creating realistic synthetic images of human faces. Because these images can be used for malicious purposes, researchers have begun to develop techniques to synthetic images. Currently, the majority of existing techniques operate by searching for statistical traces introduced when an image is synthesized by a GAN. An alternative approach that has received comparatively less research involves using semantic inconsistencies detect synthetic images. While GAN-generated synthetic images appear visually realistic at first glance, they often contain subtle semantic inconsistencies such as inconsistent eye highlights, misaligned teeth, unrealistic hair textures, etc. In this paper, we propose a new approach to detect GAN-generated images of human faces by searching for semantic inconsistencies in multiple different facial features such as the eyes, mouth, and hair. Synthetic image detection decisions are made by fusing the outputs of these facial-feature-level detectors. Through a series of experiments, we demonstrate that this approach can yield strong synthetic image detection performance. Furthermore, we experimentally demonstrate that our approach is less susceptible to performance degradation caused by post-processing than CNN-based detectors utilize statistical traces.

Introduction

In recent years, generative adversarial networks (GANs) have emerged as a technique to produce visually realistic synthetic images of people. These synthetic images can be used for a variety of malicious purposes, including as part of misinformation campaigns, creating fake social network profiles for information harvesting or phishing, etc.

To combat this problem, researchers have begun creating techniques to detect synthetic images [24, 26, 22]. The majority of existing techniques operate by searching for statistical traces introduced when an image is synthesized by a GAN. This is a common approach in multimedia forensics, which has been used to detect editing [2, 5, 8, 1], identify an image's source camera [4, 3, 13], and detect content forgery [16, 11, 21, 25]. While these approaches work well, their performance often degrades when an image is subject to post-processing such as JPEG compression or resizing.

An alternative approach that has received comparatively less research involves using semantic inconsistencies to detect synthetic images. While GAN-generated images appear visually realistic at first glance, they often contain subtle semantic inconsistencies. For example, Hu, et al. recently showed that synthetic images can be detected by identifying inconsistent corneal reflections in the eyes [14]. This work demonstrates that even though synthetic images look visually realistic, some GANs have trouble generating some semantically meaningful details. Other inconsistencies often occur including implausibly misaligned or off-centered teeth, unrealistic hair textures, mismatched earrings or inconsistent ear sizes, etc.

However, despite the existence of semantic inconsistencies, few techniques have been taking advantage of this for detection and the methods that do focus on singular semantic inconsistencies that may not occur in every facial feature. Developing synthetic image detectors that operate by searching for semantic inconsistencies is an important tool for the forensics community. This can yield several benefits, including: Utilizing multiple means of detecting synthetic images increases the likelihood that they are identified. Semantic inconsistencies should not be significantly affected by post-processing such as recompression or resizing, while statistical traces will likely be degraded or destroyed. Synthetic image detectors are vulnerable to anti-forensic attacks [7, 27]. Because semantic inconsistencies lie in a domain that is distinct from statistical traces, attacks on detectors that exploit statistical traces are unlikely to affect semantic detectors.

In this paper, we propose a new approach to detect GAN-generated images of human faces by searching for semantic inconsistencies in synthetic faces. GAN-generated images often contain several potential semantic inconsistencies, such as inconsistent eye highlights, misaligned teeth, unrealistic hair textures, etc. Despite this, we cannot be sure that any one type of inconsistency will occur in an image. As a result, relying on only a single form of semantic inconsistency is likely to yield suboptimal detection performance.

To overcome this challenge, we propose building multiple detectors to identify semantic inconsistencies in different facial features, including the eyes, mouth, and hair. Synthetic image detection decisions are made by fusing the outputs of these facial-feature-level detectors. While an individual facial-feature- level detector may have weak performance due to the infrequent occurrence of semantic inconsistencies in that feature, it is unlikely that a synthetic image will fool all facial-feature-level detectors.

We experimentally demonstrate that this approach can achieve strong synthetic image detection performance on images created using multiple GANs. Furthermore, we demonstrate that this approach is less susceptible to performance degradations caused by post-processing than other approaches which use convolutional neural networks (CNNs) to directly identify statistical traces of synthetic images.

Proposed Approach

Overview

In this paper, we propose a new approach to detect GAN-generated images of human faces. Our approach does this by identifying the presence of semantic inconsistencies in the facial

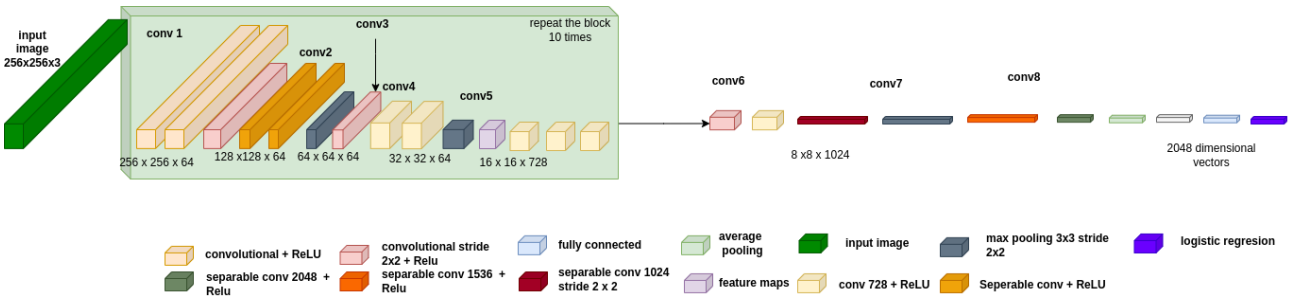

Figure 1: Overview of our single-feature semantic analysis CNN's architecture.

features of images created by GANs. To detect these semantic inconsistencies, we first need to extract the facial features that we wish to analyze. We do this by first using a facial landmark detector to identify the extrema points of each facial feature, then create a bounding box or shape that fully encloses it. After this, we extract the facial feature and scale it to a common size. This provides us with a series of facial feature sub-images from each face with normalized sizes.

To analyze these facial feature sub-images, we initially categorize the facial features into two classes: single facial features such as the mouth and hair, and dual facial features such as the eyes. For single facial features, we utilize a convolutional neural network (CNN) trained trained to produce a semantic consistency score for that feature. These CNNs are trained to identify semanitic inconsistencies within a single facial feature (i.e. different CNNs are trained for each facial feature under analysis). For dual facial features, we search for semantic inconsistencies between the pair of features (e.g. between two eyes) using a pseudo-Siamese network. Here, a single semantic consistency score is obtained for the pair of features.

After obtaining all semantic consistency scores for each facial feature, we must make an image-level authenticity decision. To do this, we fuse all semantic consistency scores by concatenating them into a single vector. This vector is then provide this to a support vector machine trained differentiate real from synthetic images.

Details of each of these algorithmic components are provided below.

Facial Feature Identification and Extraction

In order to analyze facial features for semantic inconsistencies, we must first locate them on the face, then isolate them so that they can be analyzed by a targeted neural network. In this work, we examine three types of facial features: the eyes, mouth, and hair.

We locate these facial features using keypoints identified by facial landmark detectors. For the eyes and mouth, we use keypoints provided Open-CV's facial landmark detector [6]. We identify each facial feature's extrema landmarks and use them to create a bounding box around the eye or mouth. Specifically, the sides of the rectangle that form the bounding box are determined by the farthest facial landmark that encompasses the respective feature. We then extract the pixels within each bounding box to create a new sub image containing only one facial feature.

We use a slightly different approach to isolate hair becausevit does not take a regular shape. For hair, we use Google's MediaPipe library to produce a 468 keypoint mesh on the face [20]. Keypoints corresponding to the contour of the hair are then identified and used to produce an outline of the hair region. All parts of the image outside of the hair region are then set to white.

Due to variation in pose, head size, and anatomy, some extracted facial features may be larger than others. To control for this and to standardize the inputs to our semantic analysis neural networks, we resize all sub-images of a particular facial feature to a common size.

Facial Feature Semantic Analysis

Once each facial feature has been extracted, it is analyzed using a feature-specific deep neural network that returns a semantic consistency score. The architecture of the network we use to analyze a facial feature depends on if it has a single occurrence (e.g. mouth and hair) or two occurrences (e.g. eyes).

Single Feature Analysis

We analyze facial features that occur in only a single location on the face, such as the mouth and hair, using feature-specific CNNs. Both CNNs share a common architecture shown in Fig. 1, but are trained using labeled examples of their specific feature. Further details of this training process are described in the next section

Our semantic consistency CNN, shown in Fig. 1, is inspired by Xception [10]. It takes as input 256 x 256 pixel sub-images of a facial feature. The network begins with two consecutive convolutional layers with Relu activation and a max pooling layer. This is followed by another convolutional layer with a stride of two, and an additional set of convolutional layers with Relu activation, along with a max pooling layer. Finally, we repeatedly run the results through this block of layers ten times, before feeding the resulting vectors to a convolutional layer with a stride of 2 x 2, and a 728-layer convolutional layer with Relu activation. Following this, we pass the output through three separate convolutional layers of 1024, 1536, and 2048 dimensions, each paired with Relu activation. After applying average pooling and fully connected layers, we end up with a 2048-dimensional vector that we feed into a logistic regression model to classify the specific facial feature as either synthesized or real.

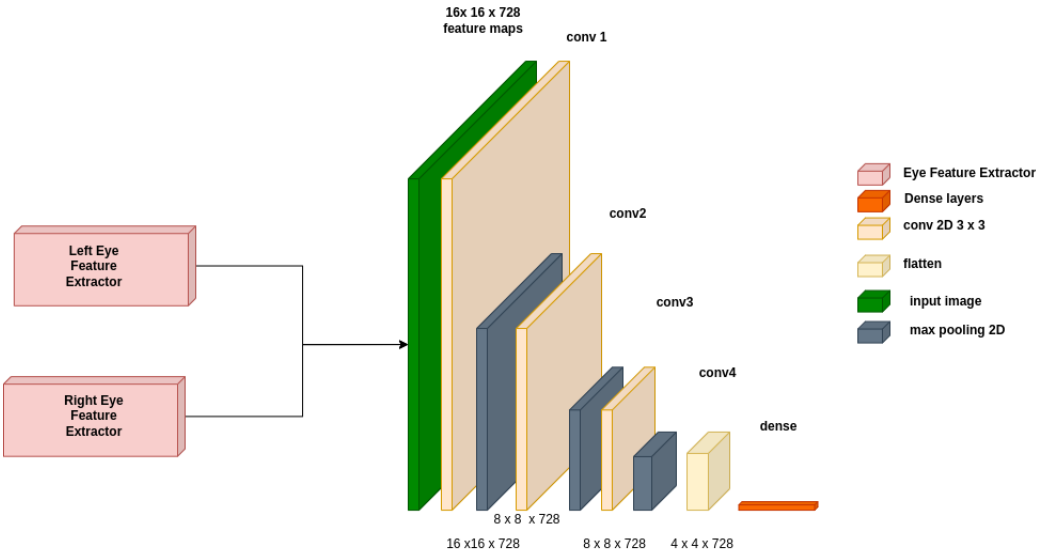

Figure 2: Overview of our proposed pseudo-Siamese network architecture for measuring the semantic consistency between dual facial features such as eyes.

The output of the network is a single semantic consistency score s taking a value between 0 and 1. Low values of s correspond to a semantically inconsistent feature - i.e. the facial feature is likely from a synthetic image.

Two-Feature Comparative Analysis

Since eyes occur in pairs, we obtain a comparative score that measures how semantically consistent two eyes are. This is done by using a pseudo-Siamese neural network. Our Siamese network is created by first using two CNNs whose architecture is shown in Fig. 1 to produce neural embeddings. One CNN is dedicated to the left eye and the other is dedicated to the right eye, which are pre-trained using the single-feature analysis training protocol.

Instead of extracting a semantic score for each facial feature, here we retain the final 16 x 16 x 728 feature map for each facial feature (i.e. each eye). The two sets of feature maps are concatenated and analyzed by the convolutional similarity network shown in Fig. 2 This neural network consists of three 2-dimensional convolution layers each followed by a max pooling layer followed by flatten and dense layers that gives us a semantic score for the pair of the eyes.

As with single facial feature analysis, the output of this network is a single semantic consistency score s between 0 and 1. This score has the same interpretation as before, with low values indicating low semantic consistency, i.e. the facial features are likely from a GAN-generated image.

Fusion

Once all the feature-level semantic consistency scores are measured, we fuse them to produce a single image-level authenticity decision. We do this by concatenating all of the feature-level scores into a single vector, then training a support vector machine (SVM) with a radial basis function kernel to classify an image as real or synthetic. Confidence scores for decisions can be obtained by using Platt scaling. An overview of this is shown in Fig. 3.

We use this method because it is highly unlikely for semantic inconsistencies to always be present within a particular facial feature. However, more often than not, a semantic inconsistency is present in at least one of the facial features. By fusing the outputs of each feature-level semantic consistency network, this allows us to determine if at least one semantic inconsistency is present in the image, thus increasing our likelihood of catching a synthetic image.

Training Protocol

Each of our networks must be trained to identify semantic inconsistencies using labeled data. Below, we describe data labeling strategies and our two-stage training protocol designed to exploit both large volumes of quickly (though possibly inaccurately) labeled data along with small volumes of highly accurately labeled data.

Data Labeling Protocols

In order to train our network, we require a labeled dataset. However, when labeling images as real or synthesized, based on their origin (i.e. real from a camera or synthetic from a GAN), we may encounter mislabeling issues. Labeling all synthetic images as semantically inconsistent is not an effective solution, as inconsistencies may not be present in every facial feature. By doing this, one would effectively label all facial features in a synthetic image as semantically inconsistent. Because some facial features in a synthetic image will be semantically consistent, this approach will result in a significant amount of mislabeled data, that will in turn decrease the accuracy of detection. Therefore, we have developed two different labeling approaches and corresponding training phases to address this issue.

Image Level Labeling (ILL) – To begin, we initially label our data on an image-level basis, categorizing each image as real or synthetic based on its source. Images retrieved from an authentic dataset such as Celeb-A HQ are labeled as real, while those

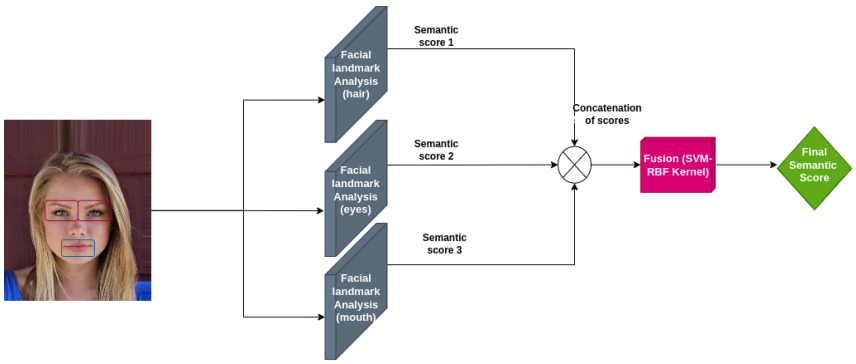

Figure 3: Overview of our overall system, including individual facial feature semantic consistency score fusion.

Facial Feature Network	Baseline Accuracy
Mouth	85.96%
Hair	64.15%
Eyes	67.09%
Fusion (Proposed)	**91.36%**

Table 1: Baseline detection performance of each facial feature network as well as our overall proposed approach.

	Training Protocol	
Network	**ILL only**	**ILL + FLL**
Proposed	**83.10%**	**91.36%**
Xception	68.29%	73.60%
Inception	60.84%	68.27%

Table 2: Influence of the training protocol and CNN architecture.

from the rest of the dataset are labeled as synthetic. This labeling method enables us to efficiently generate high volumes of data. However, as previously discussed, labeling entire images as synthetic presents a challenge. Since not every facial feature in a synthetic image may exhibit semantic inconsistencies, labeling an entire image as synthetic will result in all facial features being labeled as semantically inconsistent. Training using only this data is likely to degrade the performance of a feature level classifier.

Feature Level Labeling (FLL) – In this strategy, training data consists of semantically consistent facial features taken from real images and semantically inconsistent facial features taken from synthetic images. This involves using a human to manually inspect a specific facial feature in a synthetic image and determine if it is semantically inconsistent. While this data is well suited for training a neural network to identify semantic inconsistencies, it is difficult and time consuming to produce. As a result, it is often impractical to create large volumes of training data with feature level labels for training from scratch.

Two-Phase Training Protocol

To overcome the shortcomings of both data labeling strategies, we train our neural networks with a two phase training protocol described below.

Training Phase 1 – First, each of our semantic consistency networks are trained using a large volume of training data with image level labels that have been automatically generated. While training only using this data is suboptimal, this initial phase enables our networks to learn a reasonable model of the facial feature to be analyzed. This cannot be done using only FLL datasets because they are likely too small.

Training Phase 2 – Next, the network is fine-tuned using a smaller volume of training data with feature level labels that has been manually labeled generated. Specifically, we take the previously frozen weights and use our feature-level labeled data to fine-tune the model for every layer except the output layer. This enables our networks to gain improve their performance based on very accurately labeled FLL data, but does not require large volumes of FLL data needed to train from scratch.

Experiments

We conducted a series of experiments to validate the performance of our proposed approach. To run these experiments, we created a dataset consisting of 60,000 images where 30,000 images were real photos taken from celeb-A HQ[17] and 30,000 images were GAN synthesized using 10,000 from StyleGAN2[19], 10,000 from StyleGAN3[18], and 10,000 from StarGAN[9]. In these experiments, 80 percent of our dataset was used for training and 20 percent was used for testing our algorithm. During training, all images were given image level labels. Additionally, 1,040 images were given feature level labels for use in our two-phase training protocol.

Baseline Performance

In our first experiment, we evaluated the baseline performance of our overall system as well as the performance of each facial-feature-level semantic consistency network.

Table 1 shows our system's performance. From this table, we cau see that our proposed approach achieved an overall GAN-generated image detection rate of 91.63%. This indicates that our proposed system is able to accurately detect GAN-generated synthetic images.

	Test Accuracy					
Network	**Baseline**	**JPEG Q=90**	**JPEG Q = 80**	**JPEG Q = 70**	**Resize R = 1.5**	**Resize R = 2.0**
Proposed	91.36%	**90.05%**	**88.29%**	**84.57%**	**86.11%**	**89.47%**
Dense-Net	**95.25%**	89.83%	84.52%	80.98%	79.15%	85.05%
ResNet-50	84.16%	74.62%	71.35%	69.81%	63.55%	67.18%
Inception	93.87%	88.29%	85.42%	81.78%	78.05%	82.30%

Table 3: Experimental results showing the robustness to post-processing of our proposed approach as well as several CNNs directly trained to detect GAN traces.

In this table we can also see the detection accuracies obtained by individually examining each facial feature for semantic inconsistencies. Here, the highest single-feature accuracy obtained is 85.96%, with the other features obtaining much lower accuracies. This reinforces our intuition that searching for semantic inconsistencies withiin multiple facial feature is more likely to reveal synthetic images rather than examining only one, high-value facial feature.

Influence of Training Protocol

Next, we conducted an experiment to examine the importance of our proposed two-phase training protocol. In this experiment, we evaluated the performance of our approach using only the image level labeling (ILL) approach as well as our proposed approach that additionally fine tunes each network using data with feature level labels (FLL). Additionally, we repeated this experiment while using two alternate CNNs instead of our proposed CNN to see if the impact of two-phase training is architecture dependent.

Table 2 shows the results of our experiment. From this table, we can see that our two-phase training protocol substantially improves the performance of our system for all CNN architectures. Specifically, for our proposed CNN, fine tuning using only roughly 1,000 data points of FLL data improves our system's accuracy by over 8 percentage points.

Influence of CNN Architecture

Additionally, we used the results of the previous experiment to examine the performance of our proposed semantic consistency CNN architecture. We compared our CNN's performance to that of two other CNNs commonly used in computer vision: Inception [23] and Xception [10].

As demonstrated in Table 2, our proposed network achieves a $15-17\%$ improvement in detection accuracy over these networks. We note that this may be because our network is much smaller, thus less prone to overfitting on our training dataset.

Robustness to Post-Processing

In theory, utilizing semantic inconsistencies instead of statistical traces has the advantage of being more robust to post-processing. This is particularly important for resizing and recompression, which are common when images are uploaded to social media websites. Almost all photos uploaded to social media undergo these processes.

To evaluate our proposed approach's robustness to post-processing, we subjected each image in our test set to JPEG compressing using quality factors ranging from Q=90 to Q=10, and to resizing with scaling factors of 1.5 and 2.0. We then evaluated our system's detection performance on this post-processed data. Additionally, we compared our system's performance to that of three CNNs directly trained to detect statistical traces left by GAN generators: Res-Net 50[12], Dense-Net[15], and Inception[23].

Table 3 shows the results of this experiment. From this table, we can see that our network exhibits significant robustness to various post-processing operations. Furthermore, our network achieves significantly higher performance on post-processed data than other networks directly trained to detect GAN-generated images. We also note that our system's performance also changes the least when confronted with post-processed images. This makes sense, because semantic inconsistencies should be much less effected by post-processing than statistical traces left by GAN generators, which are heavily degraded by post-processing.

Conclusions

In this paper, we proposed a new system to detect GAN-generated synthetic images of human facees by searching for semantic inconsistencies. Our approach works by examining multiple facial features (eyes, mouth, hair) individually for semantic inconsistencies using facial-feature-specific neural networks, then fusing the resulting semantic consistency scores. We proposed a new, two-phase training protocol to leverage both high volumes of training data wit image-level labels as well as low volumes of highly accurate training data with feature-level labels. We conducted a series of experiments to evaluate our proposed system, which show that not only can our system achieve strong synthetic image detection performance, but also that it is highly robust to post-processing.

.

References

[1] Belhassen Bayar and Matthew C Stamm. A deep learning approach to universal image manipulation detection using a new convolutional layer. In *Proceedings of the 4th ACM workshop on information hiding and multimedia security*, pages 5–10, 2016.

[2] Belhassen Bayar and Matthew C. Stamm. Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection. *IEEE Transactions on Information Forensics and Security*, 13:2691–2706, 11 2018.

[3] Belhassen Bayar and Matthew C Stamm. Towards open set camera model identification using a deep learning framework. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 2007–2011. IEEE, 2018.

[4] Luca Bondi, Luca Baroffio, David Güera, Paolo Bestagini, Edward J Delp, and Stefano Tubaro. First steps toward camera model identification with convolutional neural networks.

IEEE Signal Processing Letters, 24(3):259–263, 2016.

[5] Mehdi Boroumand and Jessica Fridrich. Deep learning for detecting processing history of images. *Electronic Imaging*, 30:213–1–213–9, 01 2018.

[6] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[7] Nicholas Carlini and Hany Farid. Evading deepfake-image detectors with white-and black-box attacks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 658–659, 2020.

[8] Jiansheng Chen, Xiangui Kang, Ye Liu, and Z Jane Wang. Median filtering forensics based on convolutional neural networks. *IEEE Signal Processing Letters*, 22(11):1849–1853, 2015.

[9] 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. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 06 2018.

[10] Francois Chollet. Xception: Deep learning with depthwise separable convolutions. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 07 2017.

[11] Davide Cozzolino and Luisa Verdoliva. Noiseprint: A cnn-based camera model fingerprint. *IEEE Transactions on Information Forensics and Security*, 15:144–159, 2019.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 06 2016.

[13] Brian Hosler, Owen Mayer, Belhassen Bayar, Xinwei Zhao, Chen Chen, James A Shackleford, and Matthew Christopher Stamm. A video camera model identification system using deep learning and fusion. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8271–8275. IEEE, 2019.

[14] Shu Hu, Yuezun Li, and Siwei Lyu. Exposing gan-generated faces using inconsistent corneal specular highlights. *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 06 2021.

[15] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely Connected Convolutional Networks. 2016.

[16] Minyoung Huh, Andrew Liu, Andrew Owens, and Alexei A Efros. Fighting fake news: Image splice detection via learned self-consistency. In *Proceedings of the European conference on computer vision (ECCV)*, pages 101–117, 2018.

[17] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation. 2017.

[18] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-Free Generative Adversarial Networks. 2021.

[19] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and Improving the Image Quality of StyleGAN. 2019.

[20] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg, and Matthias Grundmann. Mediapipe: A framework for building perception pipelines. *arXiv:1906.08172 [cs]*, 06 2019.

[21] Owen Mayer and Matthew C Stamm. Exposing fake images with forensic similarity graphs. *IEEE Journal of Selected Topics in Signal Processing*, 14(5):1049–1064, 2020.

[22] Yuyang Qian, Guojun Yin, Lu Sheng, Zixuan Chen, and Jing Shao. Thinking in frequency: Face forgery detection by mining frequency-aware clues. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII*, pages 86–103. Springer, 2020.

[23] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 06 2016.

[24] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A. Efros. Cnn-generated images are surprisingly easy to spot... for now. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 06 2020.

[25] Yue Wu, Wael AbdAlmageed, and Premkumar Natarajan. Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9543–9552, 2019.

[26] 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)*, pages 1–6. IEEE, 2019.

[27] Xinwei Zhao and Matthew C Stamm. Making gan-generated images difficult to spot: a new attack against synthetic image detectors. *International Conference on Pattern Recognition (ICPR), Workshop on Artificial Intelligence for Multimedia Forensics and Disinformation Detection*, 2022.

Author Biography

Danial Samadi Vahdati*, received his bachelor's degree(2020) in Electrical Engineering from Imam Khomeini International University and is currently a PhD. student in Electrical Engineering at Drexel University focusing on multimedia forensics, deep learning and computer vision.*

Matthew C. Stamm*, received his B.S., M.S., and Ph.D. degrees in electrical engineering from the University of Maryland at College Park, College Park, MD, USA, in 2004, 2011, and 2012, respectively. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, Drexel University, Philadelphia, PA, USA. He leads the Multimedia and Information Security Lab where he and his team conduct research on multimedia forensics and machine learning.*