國立臺灣大學電機資訊學院資訊工程學系
碩士論文
Department of Computer Science and Information Engineering
College of Electrical Engineering and Computer Science
National Taiwan University
Master Thesis

藉由半監督異常偵測進行廣義造假圖像鑑識
Generalized Fake Image Forensics via Semi-Supervised Anomaly Detection

劉彥廷
Yan-Ting Liou

指導教授：莊永裕博士, 林彥宇博士
Advisor: Yung-Yu Chuang, Ph.D. Yen-Yu Lin, Ph.D.

中華民國 109 年 8 月
August, 2020

doi:10.6342/NTU202003518

國立臺灣大學碩士學位論文
口試委員會審定書

藉由半監督異常偵測進行廣義造假圖像鑑識

Generalized Fake Image Forensics via Semi-Supervised Anomaly Detection

本論文係劉彥廷君（學號 R07922073）在國立臺灣大學資訊工程學系完成之碩士學位論文，於民國 109 年 8 月 13 日承下列考試委員審查通過及口試及格，特此證明

口試委員：莊永裕　林彥宇

（指導教授）

翁昭軒　陳駿丞

系主任　莊永裕

doi:10.6342/NTU202003518

誌謝

首先，我真的非常感謝蔡易軒博士和洪暐智博士一直以來的幫助，在我受到挫折時持續鼓勵，在我沒想法時提供方向，我不會忘記這些年星期二用 Skype meeting 的日子，他們是我成長的養分和紀錄。

感謝莊永裕教授和林彥宇教授願意共同擔任我的指導老師，每個精闢的建議都使我獲益良多。

謝謝嘉賡、庭綱、芝柔、宜蓁一起在實驗室奮鬥，謝謝國騰在系館整修時借我放主機到宿舍而且幫我準備口試，謝謝侑儒、昀宣特別回來給我口試報告的建議，謝謝 CMLAB 的所有好夥伴。謝謝 B03 子賢、定暄、俞安、新凱、冠鈞不離不棄陪我講垃圾話。謝謝國高中朋友的鼓勵。

最後，感謝我的家人，讓我從小就能沒有後顧之憂的讀書，特別是這些年，母親、奶奶和哥哥的辛勞。僅以此論文，獻給我敬愛的父親，願他的在天之靈感到欣慰。

doi:10.6342/NTU202003518

摘要

近年來，圖像生成以及圖像屬性改造的方法能產生以假亂真的結果，使其有可能被用來當作製造假新聞的工具。作為反擊，學術圈也投注更多的心力在自動偵測假圖片上面。本論文提出利用半監督異常偵測的方法來增進偵測未知造假方式的能力，此方法在特徵空間中縮小真實圖像和中心的距離並且放大偽造圖像和中心的距離。和一般二分類問題不同的是，此方法並沒有強制讓偽造的圖像聚成一類。我們相信這是有幫助的，因為他體現了不同造假方式的差異性。我們將此方法應用在偵測卷積神經網路產生的圖像和偽造人臉的資料集，實驗中發現共同使用此方法和標準二元分類用的交叉熵能增進模型的通用性，我們相信這個效能增進說明了此方法的可能性。

關鍵字： 深度學習，造假圖像鑑識

doi:10.6342/NTU202003518

Abstract

Recent photo-realistic image synthesis or attribute manipulation methods could be used as a tool to create fake news. As a fightback, researchers start to put more effort on automatically detecting fake images. This thesis proposes to leverage an objective in semi-supervised anomaly detection to increase the generalization ability on detecting images generated from unseen forgery methods. This objective tries to minimize the distances between real samples and the center and maximize the distances between fake samples and the center at the same time. Unlike standard binary classification, it doesn't assume that all the fake images should be near to each other in the feature space. We believe that this is helpful for generalization since it reflects the diversity of different fake methods. We examine this method on detecting CNN-generated images and fake faces. The improvement of the generalization ability can be found when incorporating with binary cross-entropy loss. The performance gains in experiments show the potential of this method.

Keywords: Deep Learning, Fake Image Forensics

doi:10.6342/NTU202003518

Contents

doi:10.6342/NTU202003518

doi:10.6342/NTU202003518

List of Figures

doi:10.6342/NTU202003518

List of Tables

doi:10.6342/NTU202003518

doi:10.6342/NTU202003518

Chapter 1

Introduction

Recent research of deep image synthesis have shown astonishing results. These techniques, such as Generative Adversarial Networks (GANs), make generating photo-realistic fake images much more feasible than ever before. Some high resolution results generated from latest GAN are hardly distinguishable from real ones for human. These alarm people and drive the academic community to study on automatically classifying real and fake images.

The simplest way to train the classifier is to collect the fake images from the target domain for training and test on the similar distribution. However, this is often not the case for the real world. The unknown ways of manipulation might be given at test time. The misalignment of training distribution and testing distribution would make this simple classifier failed.

A more practical setting would be assumed that none or few target samples are available. Combine with labeled source data to train a general classifier for fake forensics. Some literature [29, 12] have studied on the artifact generated from upconvolution layers, which are common components in the generative models. [8] have tried to encode real and fake images separately by leveraging autoencoder to boost the generalization ability of the classifier. A recent work [28] have shown empirical results about the amazing generalization ability of classifiers trained with fake samples generated by PGGAN.

In order to tackle this practical issue, we pose the general fake forensics as a semi-supervised anomaly detection problem. We hypothesize that by incorporating the loss in

doi:10.6342/NTU202003518

[24], the diverse fake samples aren't forced to be clustered as one single class and the boundary of real samples can be more compact, which might be beneficial for generalization on detection unseen fake images.

doi:10.6342/NTU202003518

Chapter 2

Related Work

2.1 Fake Image Forensics

With the arising concerns of misuse of image synthesis and manipulation techniques, several methods have been explored to detect fake images. ForensicTransfer [8] design a novel autoencoder structure that aims to encoder fake and real feature into separate neurons. WatchUpConv [12] and AutoGAN [29] have found the artifact generated from up convolution layer. The former used spectral distortions to train a classifier and the latter tried to synthesize the artifact by training a GAN simulator. Deep Distribution Transfer [1] improved the generalization ability by using a mixture model-based loss and a spatial mixup augmentation. CNNDetection [28] found that classifiers trained with fake images generated from PGGAN are capable of detecting other CNN generated images.

2.2 Anomaly Detection

Anomaly detection (AD) is an old topic in machine learning. Out of distribution detection and one class classification are two related keywords. There are classical methods like One Class Support Vector Machine (OCSVM) [26] and Support Vector Data Description (SVDD) [27]. The former tried to maximize the margin of the hyperplane with respect to the origin, and the latter tired to find a hypersphere to enclose the given samples. Recently, there was some work leveraging deep learning. OCCNN [20] used a zero center Gaussian

doi:10.6342/NTU202003518

noise in feature space as the pseudo-negative class and train a classifier with cross entropy loss. In [22], they utilize an external unrelated labeled data to learn descriptive features while maintaining low intra-class variance in the feature space for the given class, and feed these features to classical methods to perform anomaly detection. DeepSVDD [25] directly train the neural network on an anomaly detection based objective, which tighten the given samples in latent space.

2.3 Semi-Supervised Anomaly Detection

Instead of only utilizing the normal samples, semi-supervised anomaly detection assume that we can access some labeled abnormal samples and try to build a method for this scenario. Since the abnormal samples are often diverse and can't be described as the same class, it is suboptimal to be solved as a binary classification problem. DeepSAD [24] extends loss in DeepSVDD [25] to also maximize the distances between abnormal samples and the center of normal samples in latent space. Daniel et al.[11] extend the standard loss of VAE to train an encoder such that it can separate latent codes of normal and abnormal samples.

doi:10.6342/NTU202003518

Chapter 3

Method

In this chapter, we first construct several experiments to examine the generalization ability of a simple real and fake classifier. Then, we define the problem setting formally. Finally, we describe the semi-supervised anomaly detection approach for this problem.

3.1 Preliminary

3.1.1 Generalization ability for unseen instances

In this experiment, we train the classifier with standard binary cross entropy on real and fake face images. There are totally 9 classifiers trained. They are allowed to access the same set of real samples (from CelebA) and equal number of fake images. The number of sources of fake face images that are used to train these classifiers are different. The first classifier is only trained with the fake images generated from one PGGAN instance, while the last classifier is able to see 9 PGGAN instances at training time. Notice that different PGGAN instances only differ in training seed. Afterward, these classifiers are tested with fake face images generated from an unseen PGGAN instance.

Table 3.1 shows that the accuracy of detecting images from an unseen PGGAN instance is getting better when the number of seen PGGAN instances in training is more. The big performance improvement (from 49.23% to 86.62%) can be observed between seeing 1 instance and seeing 2 instances in training. This experiment shows that the per-

doi:10.6342/NTU202003518

(a) CelebA

(b) PGGAN

(c) CramerGAN

(d) MMDGAN

(e) SNGAN

Figure 3.1: Sample images for our pilot study. (a) are real and others are fake.

formance would decay even when testing with fake images generated from same architecture but with different training seeds. But good news is that this can be easily remedied by accessing fake images generated from more instances in training time. It is an easy and cheap way to make the classifier robust when the architecture of fake image generator are known.

3.1.2 Generalization ability for unseen architectures

In order to check the generalization ability of fake face images detection across different GAN types. We tested the 9 classifiers in previous experiment with fake face images

doi:10.6342/NTU202003518

# seen PGGAN instances in training	PGGAN
1	49.23%
2	86.62%
3	95.98%
4	97.06%
5	98.01%
6	97.23%
7	98.06%
8	98.82%
9	98.72%

Table 3.1: Impact of number of seen PGGAN instance in training. Classifiers are tested on a set of fake-only samples, which are generated from an unseen PGGAN instance. The results are in accuracy.

generated from other GANs, include CramerGAN [2], MMDGAN [3], and SNGAN [19]. Samples of these GANs can be viewed in Figure 3.1. Table 3.2 shows that the more the number of seen PGGAN instances in training, the better the accuracy is.

It is interesting to see that the performance of detecting images from CramerGAN and MMDGAN are already relative high at start (86.75% and 75.26%). Their performances approximately reach the peak when number of seen PGGAN instances in training are 5. The performance gain after seeing fake images generated from more than 5 PGGAN instances are negligible. One possible reason is that the images generated from CramerGAN and MMDGAN are relatively fake compare to PGGAN, and their artifacts can be well represented by some bad samples generated from PGGAN. Therefore, it is easier for classifier to detect fake face images generated from them.

On the other hand, fake face images generated from SNGAN are harder to detect, but the performance improvement while accessing images generated from more PGGAN instances can be constantly observed within the experiment. This concludes that the generalization ability of detecting fake face images from unseen GANs can also be improved by allowing access to fake images generated from more PGGAN instances.

In Table 3.3, 3.4, 3.5 and 3.6, we examine the generalization ability on detecting unseen GAN types for classifiers trained on CramerGAN, MMDGAN and SNGAN. The good generalization ability between CramerGAN and MMDGAN could be an indirect evidence that CramerGAN and MMDGAN share some common artifacts. Other than that,

doi:10.6342/NTU202003518

# seen PGGAN instances in training	CramerGAN	MMDGAN	SNGAN
1	86.75%	75.26%	17.87%
2	98.37%	96.05%	49.13%
3	99.11%	98.63%	68.07%
4	99.65%	99.06%	69.57%
5	99.60%	99.46%	76.15%
6	99.45%	99.28%	74.11%
7	99.65%	99.31%	74.02%
8	99.59%	99.62%	77.86%
9	99.72%	99.56%	80.17%

Table 3.2: Impact of number of seen PGGAN instance in training. Classifiers are tested on three set of fake-only samples, which are generated from CramerGAN, MMDGAN, and SNGAN, respectively. The results are in accuracy.

# seen CramerGAN instances in training	MMDGAN	PGGAN	SNGAN
1	1.92%	0.01%	0.04%
2	47.44%	0.03%	0.71%
3	71.78%	0.02%	0.03%

Table 3.3: Train on CramerGAN and test on other GANs. The results are in accuracy.

the overall generalization ability is quite poor for them. It is also worth noting that none of them have the generalization ability like the one trained with fake face images generated from PGGANs.

To give a small conclusion from these experiments, we can say that by training with diverse and relative well fake samples (in this case, the samples generated from PGGANs), one can get a pretty generalized real and fake classifier.

3.2 Target Domain Agnostic Fake Image Forensics

In this section, we first formally define the problem setting and explain our method in detail.

# seen MMDGAN instances in training	CramerGAN	PGGAN	SNGAN
1	68.89%	0.01%	0.02%
2	99.00%	0.00%	0.02%
3	99.98%	0.00%	0.02%

Table 3.4: Train on MMDGAN and test on other GANs. The results are in accuracy.

doi:10.6342/NTU202003518

# seen SNGAN instances in training	CramerGAN	MMDGAN	PGGAN
1	0.82%	0.54%	0.15%
2	0.77%	0.41%	0.42%
3	2.97%	2.00%	1.40%

Table 3.5: Train on SNGAN and test on other GANs. The results are in accuracy.

# seen instances per architecture in training	PGGAN
1	0.14%
2	0.86%
3	1.90%

Table 3.6: Train on CramerGAN, MMDGAN, and SNGAN then test on PGGAN. The results are in accuracy.

3.2.1 Problem Definition

Given datasets $\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_M$, where M is the number of datasets. Let $\mathcal{S}_i = \{(\boldsymbol{x}_{i,j}, y_{i,j})\}_{j=1}^{N_i}$ of N_i labeled samples, where each $\boldsymbol{x}_{i,j} \in \mathbb{R}^D$ is a D-dimensional input image and $y_{i,j} \in \{0, 1\}$ is the corresponding label. Here, $y = 0$ denotes a real image and $y = 1$ denotes a fake one. The goal is to leverage these datasets and learn a neural network $\phi(\cdot; \theta) : \mathbb{R}^D \to \mathbb{R}$, where θ are the parameters of the neural network, such that it can predict fakeness scores for images generated from unseen fake sources.

3.2.2 Improve Generalization Ability through Semi-Supervised Anomaly Detection

We make the connection between target domain agnostic fake image forensics and semi-supervised anomaly detection since they share similar properties. In semi-supervised anomaly detection, the goal is to create a tight boundary around real samples given real samples and few kinds of fake samples. It is also the same for target domain agnostic fake image forensics, which we are only allowed to access real images and limited kinds of fake images since there are unlimited kinds of ways to produce fake images.

We utilize the loss from recently success semi-supervised anomaly detection method [24] to tighten the boundary of real samples. Let $N = \sum_{i=1}^{M} N_i$ and $f(\cdot; \theta_f) : \mathbb{R}^D \to \mathbb{R}^E$ be a neural network that map an D-dimensional input into a latent code in E-dimensional

doi:10.6342/NTU202003518

feature space.

$$\mathcal{L}_{sad} = \frac{1}{N}\sum_{i=1}^{M}\sum_{j=1}^{N_i}(\|f(\boldsymbol{x}_{i,j};\theta_f) - \boldsymbol{c}\|_2^2)^{1-2y_{i,j}} \tag{3.1}$$

,where c is obtained by calculating the center of features of real samples. Namely,

$$\boldsymbol{c} = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N_i} f(\boldsymbol{x}_{i,j};\theta_f)\cdot\mathbf{1}(y_{i,j}=0)}{\sum_{i=1}^{M}\sum_{j=1}^{N_i}\mathbf{1}(y_{i,j}=0)} \tag{3.2}$$

where $\mathbf{1}(\cdot)$ is an indicator function.

The key idea of this loss is to minimize the distances between real samples in feature space and maximize the distances between real and fake samples in feature space. This loss doesn't force the fake samples to be clustered into one class. Instead, this loss only encourages real samples to form a class and force the fake samples to be as far as possible to the real samples. This is suitable in our cases since the number of ways to create fake images are countless and diverse, which might be suboptimal to make them into one category.

The fakeness scoring function can be defined by the distance between a latent code and the center of normal samples in latent space. The larger the score is, the faker the image is. We can formulate the equation of fakeness score computed by the code distance as follows,

$$S_{CD}(\boldsymbol{x}) = \|f(\boldsymbol{x}) - \boldsymbol{c}\|_2^2 \tag{3.3}$$

Figure 3.2 shows that $\mathcal{L}_{sad}$ is also capable of being used with standard binary cross-entropy loss jointly. We first recap the binary cross-entropy loss. Let $l(\cdot;\theta_l) : R^E \to R$ be a neural network followed by a sigmoid function. The binary cross-entropy loss can be formulated as

$$\mathcal{L}_{bce} = \frac{1}{N}\sum_{i=1}^{M}\sum_{j=1}^{N_i} y_{i,j}\log\left(l(f(\boldsymbol{x}_{i,j}))\right) + (1-y_{i,j})\log\left(1-l(f(\boldsymbol{x}_{i,j}))\right) \tag{3.4}$$

By attaching a classifier l such that taking the feature output from f as input, we can jointly

doi:10.6342/NTU202003518

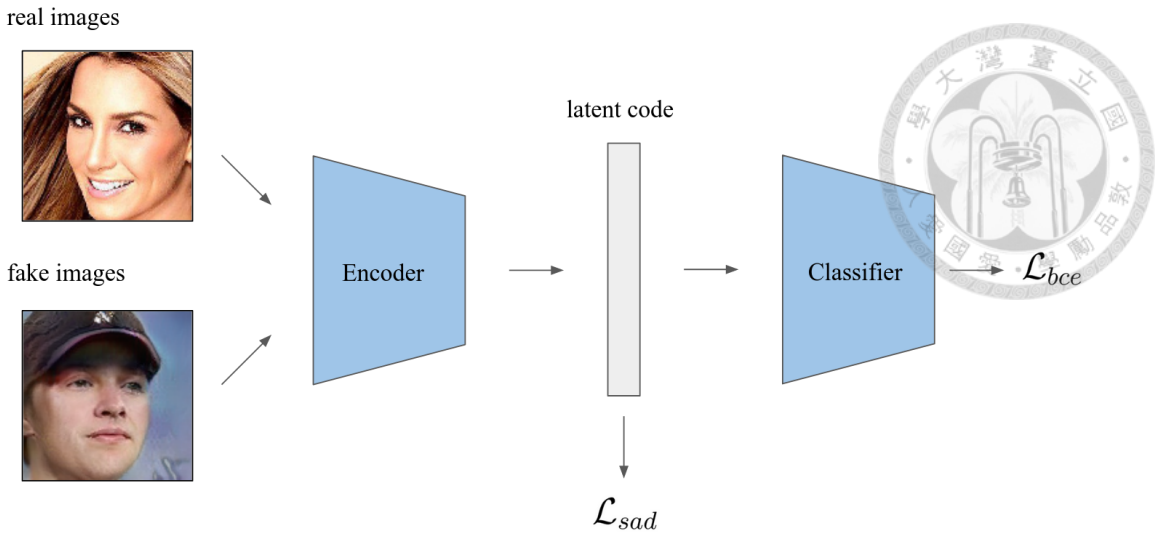


Figure 3.2: Optimize $\mathcal{L}_{sad}$ and $\mathcal{L}_{bce}$ jointly.

optimize these two objective. Therefore, the total loss can be expressed as

$$\mathcal{L} = \lambda \mathcal{L}_{sad} + \mathcal{L}_{bce} \tag{3.5}$$

where λ is a hyperparameter to control the effectness of $\mathcal{L}_{sad}$. We can also use the logit prediction of classifier to define a fakeness scoring function. After applying the sigmoid function on the logit prediction of the classifier, we can get the probability of fraud. The fakeness function according to the prediction of classifier can be formulated as,

$$S_{SL}(\boldsymbol{x}) = l(f(\boldsymbol{x})) \tag{3.6}$$

Notice that the output is between 0 and 1. The larger the score is, the faker the image is.

doi:10.6342/NTU202003518

Chapter 4

Experiments and Discussion

This chapter is divided into three parts. First, we describe the datasets. Second, we elaborate the implementation details and the evaluation metrics. Finally, the quantitative results are reported and analyzed.

4.1 Datasets

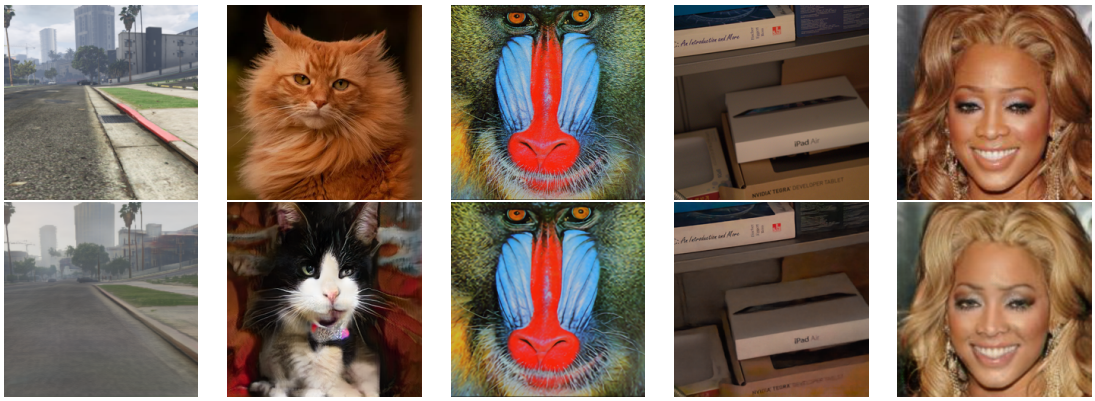
In this section, we describe details of the datasets.

4.1.1 CNN Synth

CNN Synth is recently provided by [28]. It consists of total 13 sub datasets. They are BigGAN [4], CRN [6], CycleGAN [30], DeepFake [23], GauGAN [21], IMLE [17], PGGAN [13], SAN [9], SITD [5], StarGAN [7], StyleGAN [14], StyleGAN2 [15], WhichFaceIsReal. Detail statistical information can be found in Table 4.1. Figure 4.1 shows real and fake samples of them. Although the tasks are different between these sub datasets, all of them are related to generate images through CNN. Thus, in this dataset, CNN-touched images are considered fake. Notice that all the sub datasets in CNN Synth are only used for testing.

doi:10.6342/NTU202003518

(a) BigGAN (b) CRN (c) CycleGAN (d) DeepFake (e) GauGAN



(f) IMLE (g) PGGAN (h) SAN (i) SITD (j) StarGAN

(k) StyleGAN (l) StyleGAN2 (m) WhichFaceIsReal

Figure 4.1: Sample images of CNN Synth. In every subfigure, the top image is the real one and the bottom image is the fake one.

doi:10.6342/NTU202003518

Family	Method	Image Source	# Images	
			real	fake
Unconditional GAN	PGGAN	LSUN	4000	4000
	StyleGAN	LSUN	5991	5991
	StyleGAN2	LSUN	7988	7988
	WhichFaceIsReal	FFHQ	1000	1000
	BigGAN	ImageNet	2000	2000
Conditional GAN	CycleGAN	Style/object transfer	1321	1321
	StarGAN	CelebA	1999	1999
	GauGAN	COCO	5000	5000
Perceptual loss	CRN	GTA	6382	6382
	IMLE	GTA	6382	6382
Low-level vision	SITD	Raw camera	180	180
	SAN	Standard SR benchmark	219	219
Deepfake	FaceForensics++	Videos of faces	2707	2698

Table 4.1: The statistical information of CNN Synth

4.1.2 Many Fake Faces (MFF)

We collect a dataset called Many Fake Faces (MFF), which contains real face images and fake face images generated from different forgery methods. We use FFHQ, StyleGAN (FFHQ), FaceApp (FFHQ), StarGAN (CelebA), PGGAN (CelebA), StyleGAN (CelebA) collected in DFFD [10] videos frames in Celeb DF [18]. All the faces are cropped and aligned using dlib [16]. The detail statistic information is summarized in Table 4.2, while Figure 4.2 shows some sample images.

4.2 Implementation Details and Evaluation Metrics

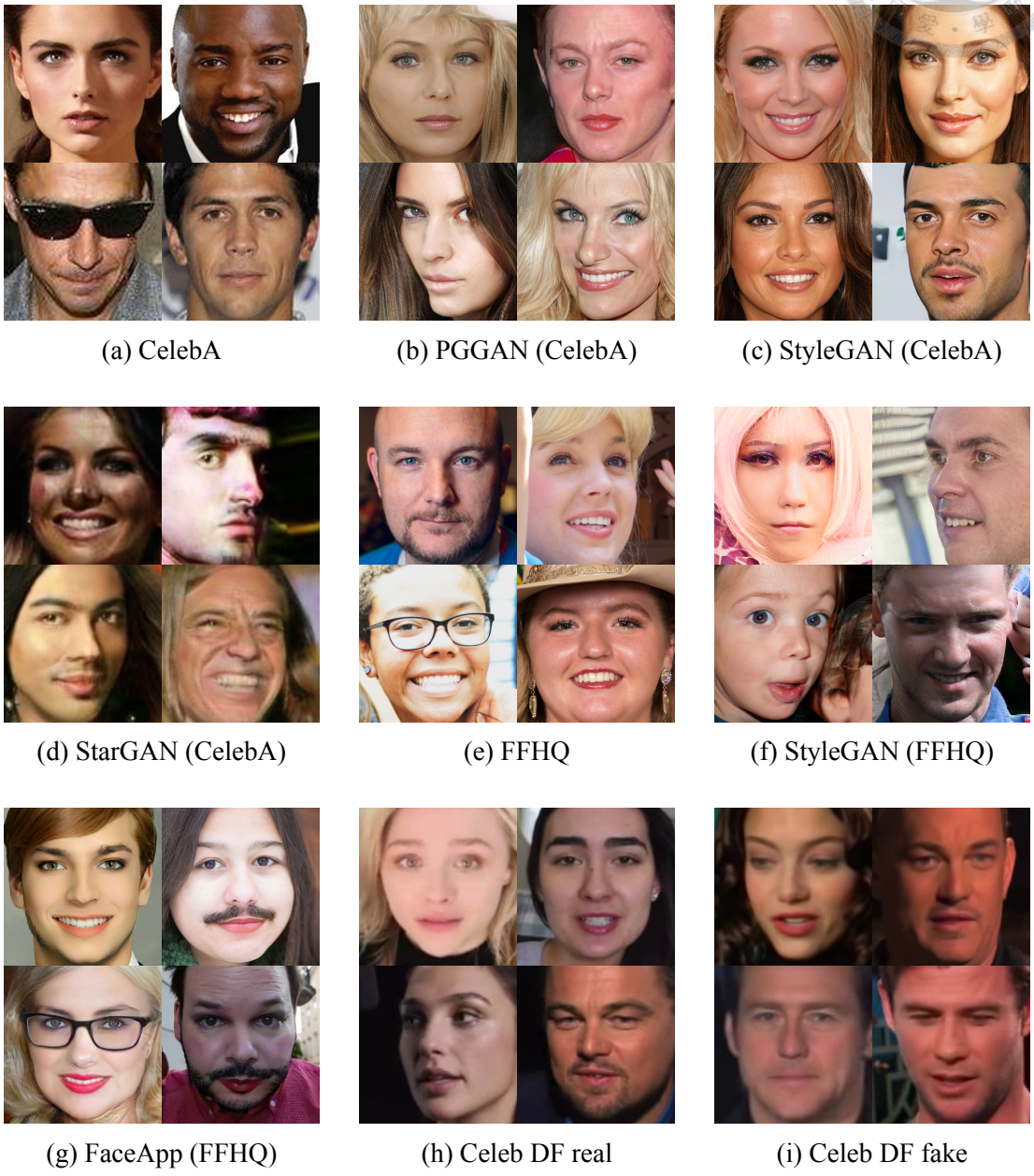
We use ResNet-18 as our backbone model and use the latent code output from layer 4 to compute $\mathcal{L}_{sad}$. The model is pretrained on ImageNet then finetune on fake image forensics dataset. We choose Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ for optimization. We try $\lambda \in \{0.001, 0.01, 0.1, 1\}$ and report the best results. The batch size is 64 and the initial learning rate is $1e-4$. The exponential learning rate decay scheduler is adopted by setting γ to 0.9. Early stopping by setting the patience to 10 is used for preventing overfitting, and we select the best model by choosing the model with the lowest loss on validation set.

doi:10.6342/NTU202003518

Name	Method	Image Source	# Images	
			real	fake
Real vs StyleGAN (FFHQ)	StyleGAN	FFHQ	10000	9999
			999	1000
			4000	4000
Real vs FaceApp (FFHQ)	FaceApp	FFHQ	10000	6309
			999	999
			4000	4000
Real vs StyleGAN (CelebA)	StyleGAN	CelebA	10000	10000
			1000	1000
			4000	4000
Real vs StarGAN (CelebA)	StarGAN	CelebA	10000	10000
			1000	1000
			4000	4000
Real vs PGGAN (CelebA)	PGGAN	CelebA	10000	9975
			1000	998
			4000	4000
Celeb DF	Improved DeepFake	Celeb DF	15561	15160
			5271	5312
			5644	5320

Table 4.2: The statistical information of Many Fake Faces. For each sub-dataset, the top, middle and bottom row shows the number of images in training, validation and testing set, respectively.

doi:10.6342/NTU202003518



(a) CelebA (b) PGGAN (CelebA) (c) StyleGAN (CelebA)

(d) StarGAN (CelebA) (e) FFHQ (f) StyleGAN (FFHQ)

(g) FaceApp (FFHQ) (h) Celeb DF real (i) Celeb DF fake

Figure 4.2: Sample images of Many Fake Faces.

doi:10.6342/NTU202003518

Like [28], we evaluate our models with Average Precision (AP), the equation of AP can be formulated as follows,

$$AP = \sum_{n} (R_n - R_{n-1}) P_n \tag{4.1}$$

where P_n and R_n are the precision and recall at the n-th threshold. Unlike accuracy, AP is a threshold-less metric. It summarizes the precision-recall curve into one single value and is desired to analyze a sequence of scores and their labels in binary classification.

4.3 Results on CNN Synth

Follow [28], we use a large scale dataset that only contains PGGAN-generated images and real images as the training set. There are total 20 PGGAN models are used. Each of them are trained on a different LSUN object category. Each model generated 36000 images for training and 200 images for validation. In total, there are 720000 images for training and 4000 images for validation. After training, models are evaluated on CNN Synth. Table 4.3 shows the comparison results. Although the model trained solely with $\mathcal{L}_{sad}$ doesn't outperform the one trained with $\mathcal{L}_{bce}$, the one trained with jointly objective has shown superior result in average AP. This indicates the potential benefit of using the semi-supervised anomaly detection based loss in this task.

4.4 Results on MFF

First, we choose FFHQ and two forgery methods applied on it: StyleGAN (FFHQ) and FaceApp (FFHQ) to examine on the generalization ability between forgery methods in the same real domain. Table 4.4 shows the generalization results of training on FFHQ.vs.StyleGAN (FFHQ). All methods achieve perfect or almost perfect performance when testing with the fake images generated from the method that already seen in the training stage. When testing with fake images generated from an unseen method (FaceApp), the performance drop is expected. However, the models trained with $\mathcal{L}_{sad}$ are capable of remaining higher per-

doi:10.6342/NTU202003518

Testing Set	BCE	SAD	BCE+SAD	
			S_{SL}	S_{CD}
WhichFaceIsReal	91.62%	89.40%	91.57%	93.50%
StyleGAN2	99.97%	98.26%	96.90%	99.73%
StyleGAN	98.51%	91.38%	87.93%	95.35%
StarGAN	100.00%	100.00%	99.97%	100.00%
SITD	98.77%	98.81%	91.10%	98.71%
SAN	73.17%	81.99%	79.04%	81.31%
PGGAN	100.00%	100.00%	100.00%	100.00%
IMLE	95.57%	94.25%	72.04%	98.96%
GauGAN	70.45%	68.12%	69.51%	71.12%
Deepfake	98.43%	98.66%	98.45%	98.71%
CycelGAN	91.02%	85.69%	83.97%	87.23%
CRN	96.86%	93.63%	72.06%	98.77%
BigGAN	77.12%	72.40%	72.13%	74.43%
avg	91.65%	90.20%	85.74%	**92.14**%

Table 4.3: Experimental results on CNN Synth.

Testing Set	BCE	SAD	BCE+SAD	
			S_{SL}	S_{CD}
Real vs StyleGAN (FFHQ)	100.00%	99.99%	99.97%	99.99%
Real vs FaceApp (FFHQ)	68.95%	70.51%	59.32%	**71.43**%

Table 4.4: Train on StyleGAN (FFHQ) and test on FaceApp (FFHQ).

formance. Table 4.5 shows the generalization results of training on FFHQ.vs.FaceApp (FFHQ). The models trained with $\mathcal{L}_{sad}$ don't outperform the model trained with $\mathcal{L}_{bce}$ this time.

Next, we examine the generalization ability when the training sets are not limited to one domain. In Table 4.6, we trained the classifiers with all the images in MFF except for FaceApp (FFHQ). It can be observed that all models perform pretty well when given the seen fake images. For generalization ability, We have similar observation like the

Testing Set	BCE	SAD	BCE+SAD	
			S_{SL}	S_{CD}
Real vs StyleGAN (FFHQ)	**57.43**%	45.03%	56.70%	53.08%
Real vs FaceApp (FFHQ)	99.63%	96.39%	99.45%	99.18%

Table 4.5: Train on FaceApp (FFHQ) and test on StyleGAN (FFHQ).

doi:10.6342/NTU202003518

Testing Set	BCE	SAD	BCE+SAD	
			S_{SL}	S_{CD}
Real vs StyleGAN (FFHQ)	99.99%	99.95%	100.00%	100.00%
Real vs FaceApp (FFHQ)	67.29%	63.16%	64.84%	**70.93%**
Real vs StyleGAN (CelebA)	100.00%	100.00%	100.00%	100.00%
Real vs StarGAN(CelebA)	100.00%	100.00%	100.00%	100.00%
Real vs PGGAN (CelebA)	99.94%	99.88%	99.93%	99.89%
Celeb DF	99.58%	98.61%	99.20%	99.15%

Table 4.6: Train on all images in MFF except for FaceApp (FFHQ) and test on FaceApp (FFHQ).

Testing Set	BCE	SAD	BCE+SAD	
			S_{SL}	S_{CD}
Real vs StyleGAN (FFHQ)	61.23%	47.26%	54.84%	**69.41%**
Real vs FaceApp (FFHQ)	99.60%	99.35%	99.39%	95.97%
Real vs StyleGAN (CelebA)	99.99%	99.93%	100.00%	99.94%
Real vs StarGAN (CelebA)	100.00%	99.99%	100.00%	99.93%
Real vs PGGAN (CelebA)	99.55%	99.46%	99.77%	98.32%
Celeb DF	99.21%	98.76%	99.61%	98.88%

Table 4.7: Train on all images in MFF except for StyleGAN (FFHQ) and test on StyleGAN (FFHQ).

one in Table 4.3 such that the model trained with the jointly objective achieve the best performance and the model trained solely with $\mathcal{L}_{sad}$ shows the worse result than the model trained with $\mathcal{L}_{bce}$. In Table 4.7, we leave the StyleGAN (FFHQ) out of training instead. We have the similar observation as the previous one.

Finally, we examine the generalization ability on a harder setting. In Table 4.8, we leave FFHQ, StyleGAN (FFHQ) and FaceApp (FFHQ) out of training, which means that classifiers aren't aware of the knowledge about FFHQ. Again, the model trained with both objective shows the best result in terms of the average AP of all the test sets contains unseen fake images. In Table 4.9, we leave the real and fake samples in Celeb DF out of training. The model trained with both objective has the slightly performance improvement over the one trained with BCE. However, all of them are near chance. This shows that Celeb DF is a quite challenging domain to generalize.

To summarize, according to the experiments, though it is often not helpful while only

doi:10.6342/NTU202003518

	Testing Set	BCE	SAD	BCE+SAD	
				S_{SL}	S_{CD}
a.	Real vs StyleGAN (FFHQ)	61.76%	54.41%	**70.37**%	62.73%
b.	Real vs FaceApp (FFHQ)	**49.25**%	36.47%	45.65%	46.67%
c.	Real vs StyleGAN (CelebA)	100.00%	100.00%	99.99%	100.00%
d.	Real vs StarGAN (CelebA)	100.00%	99.98%	99.99%	99.99%
e.	Real vs PGGAN (CelebA)	98.37%	99.83%	99.85%	96.61%
f.	Celeb DF	99.65%	98.64%	99.50%	98.23%
	average of a. and b.	55.51%	45.44%	**58.01**%	54.70%

Table 4.8: Train on CelebA, StyleGAN (CelebA), StarGAN (CelebA), PGGAN (CelebA) and Celeb DF. Then, test on StyleGAN (FFHQ) and FaceApp (FFHQ).

Testing Set	BCE	SAD	BCE+SAD	
			S_{SL}	S_{CD}
Real vs StyleGAN (FFHQ)	99.98%	99.88%	99.98%	99.98%
Real vs FaceApp (FFHQ)	99.62%	99.11%	99.60%	99.44%
Real vs StyleGAN (CelebA)	100.00%	100.00%	100.00%	100.00%
Real vs StarGAN (CelebA)	100.00%	100.00%	100.00%	100.00%
Real vs PGGAN (CelebA)	99.84%	99.84%	99.89%	99.89%
Celeb DF	49.31%	49.79%	**50.33**%	49.09%

Table 4.9: Train on CelebA, StyleGAN (CelebA), StarGAN (CelebA), PGGAN (CelebA), FFHQ, StyleGAN (FFHQ) and FaceApp (FFHQ). Then, test on Celeb DF.

doi:10.6342/NTU202003518

Testing Set	Precision	Recall	Optimal Threshold	Accuracy
WhichFaceIsReal	86.5%	87.5%	0.0731	86.8%
StyleGAN2	95.3%	94.3%	6.22×10^{-6}	94.8%
StyleGAN	96.7%	97.0%	5.08×10^{-5}	96.8%
StarGAN	93.0%	92.6%	0.1392	92.8%
SITD	92.8%	92.8%	0.2127	92.5%
SAN	58.3%	92.7%	4.34×10^{-11}	63.0%
PGGAN	100.0%	100.0%	0.9950	100.0%
IMLE	94.8%	96.2%	0.9998	95.4%
GauGAN	82.6%	89.8%	0.0006	85.4%
Deepfake	75.8%	87.1%	8.17×10^{-6}	79.6%
CycelGAN	83.3%	90.1%	0.0173	86.0%
CRN	94.8%	95.8%	0.9998	95.3%
BigGAN	75.9%	90.2%	3.64×10^{-5}	80.7%

Table 4.10: The optimal threshold across sub-datasets in CNN Synth varies much.

optimizing $\mathcal{L}_{sad}$. it is beneficial for generalization ability while optimizing $\mathcal{L}_{sad}$ and $\mathcal{L}_{bce}$ jointly.

4.5 Future Work

Since the experimental results show the benefit of generalization when utilizing a method that was originally used to tackle semi-supervised anomaly detection, it might be an interesting direction to leverage other methods in that domain. We might be able to train a VAE with an additional constrain that separate real and fake samples in feature space.

In Table 4.10, we have observed the optimal threshold across domains varies much. It will be an issue in practical when we are trying to predict a crisp output such as 0 (real) or 1 (fake) for an image given from an unknown domain. This issue can't be shown in AP since it is a threshold-less metric. Therefore, another practical direction for this problem would be tried to come up with a universal threshold or domain adapted threshold to perform crisp prediction on target domain agnostic fake image forensics.

doi:10.6342/NTU202003518

Chapter 5

Conclusion

In this thesis, we analyze the generalization ability of a real and fake classifier across different GAN instances and different GAN types. Then, we make the connection between target domain agnostic fake image forensics and semi-supervised anomaly detection. Experimental results on generic CNN-touched dataset and face specific forgery dataset MFF show the improvement in AP for the classifier trained with semi-supervised anomaly detection objective in feature space and standard binary cross entropy loss in output space jointly. We believe that this direction can benefit the generalization ability on detection unseen forgery images.

doi:10.6342/NTU202003518

Bibliography

[1] S. Aneja and M. Nießner. Generalized zero and few-shot transfer for facial forgery detection, 2020.

[2] M. G. Bellemare, I. Danihelka, W. Dabney, S. Mohamed, B. Lakshminarayanan, S. Hoyer, and R. Munos. The cramer distance as a solution to biased wasserstein gradients. *CoRR*, abs/1705.10743, 2017.

[3] M. Binkowski, D. J. Sutherland, M. Arbel, and A. Gretton. Demystifying MMD gans. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[4] A. Brock, J. Donahue, and K. Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2018.

[5] C. Chen, Q. Chen, J. Xu, and V. Koltun. Learning to see in the dark. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3291–3300, 2018.

[6] Q. Chen and V. Koltun. Photographic image synthesis with cascaded refinement networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1511–1520, 2017.

[7] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Pro-*

doi:10.6342/NTU202003518

ceedings of the IEEE conference on computer vision and pattern recognition, pages 8789–8797, 2018.

[8] D. Cozzolino, J. Thies, A. Rössler, C. Riess, M. Nießner, and L. Verdoliva. Forensictransfer: Weakly-supervised domain adaptation for forgery detection. *CoRR*, abs/1812.02510, 2018.

[9] T. Dai, J. Cai, Y. Zhang, S.-T. Xia, and L. Zhang. Second-order attention network for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11065–11074, 2019.

[10] H. Dang, F. Liu, J. Stehouwer, X. Liu, and A. Jain. On the detection of digital face manipulation. In *In Proceeding of IEEE Computer Vision and Pattern Recognition*, Seattle, WA, June 2020.

[11] T. Daniel, T. Kurutach, and A. Tamar. Deep variational semi-supervised novelty detection. *arXiv preprint arXiv:1911.04971*, 2019.

[12] R. Durall, M. Keuper, and J. Keuper. Watch your up-convolution: Cnn based generative deep neural networks are failing to reproduce spectral distributions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[13] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[14] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[15] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020.

doi:10.6342/NTU202003518

[16] D. E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.

[17] K. Li, T. Zhang, and J. Malik. Diverse image synthesis from semantic layouts via conditional imle. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4220–4229, 2019.

[18] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu. Celeb-df: A large-scale challenging dataset for deepfake forensics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3207–3216, 2020.

[19] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[20] P. Oza and V. M. Patel. One-class convolutional neural network. *CoRR*, abs/1901.08688, 2019.

[21] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019.

[22] P. Perera and V. M. Patel. Learning deep features for one-class classification. *CoRR*, abs/1801.05365, 2018.

[23] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1–11, 2019.

[24] L. Ruff, R. A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K.-R. Müller, and M. Kloft. Deep semi-supervised anomaly detection. In *International Conference on Learning Representations*, 2020.

doi:10.6342/NTU202003518

[25] L. Ruff, R. A. Vandermeulen, N. Görnitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft. Deep one-class classification. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 4393–4402, 2018.

[26] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. C. Platt. Support vector method for novelty detection. In *Advances in neural information processing systems*, pages 582–588, 2000.

[27] D. M. Tax and R. P. Duin. Support vector data description. *Machine learning*, 54(1):45–66, 2004.

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

[29] X. Zhang, S. Karaman, and S.-F. Chang. Detecting and simulating artifacts in gan fake images. In *WIFS*, 2019.

[30] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

doi:10.6342/NTU202003518