EXAMINING PERFORMANCE OF IDINVERT GAN MODEL ON IMITATING REAL HUMAN FACES

PATTANADEJ CHAENGSRISUK

Graduate School Srinakharinwirot University

2022

การตรวจจับภาพใบหน้าปลอมที่สร้างจากโครงข่ายประสาทเทียมก่อกำเนิดแบบมีคู่ปรปักษ์

พัฒนเดช แจ้งศรีสุข

สารนิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการข้อมูล
คณะวิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ
ปีการศึกษา 2565

EXAMINING PERFORMANCE OF IDINVERT GAN MODEL ON IMITATING REAL HUMAN FACES

PATTANADEJ CHAENGSRISUK

A Master's Project Submitted in Partial Fulfillment of the Requirements
for the Degree of MASTER OF SCIENCE
(Data Science)
Faculty of Science, Srinakharinwirot University
2022

THE MASTER’S PROJECT TITLED

EXAMINING PERFORMANCE OF IDINVERT GAN MODEL ON IMITATING REAL HUMAN FACES

BY

PATTANADEJ CHAENGSRISUK

HAS BEEN APPROVED BY THE GRADUATE SCHOOL IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE MASTER OF SCIENCE
IN DATA SCIENCE AT SRINAKHARINWIROT UNIVERSITY

(Assoc. Prof. Dr. Chatchai Ekpanyaskul, MD.)
Dean of Graduate School

ORAL DEFENSE COMMITTEE

.... Major-advisor	.... Chair
(Dr.Napa Sae-bae)	(Dr.Nida Chatwattanasiri)
	.... Committee
	(Asst. Prof. Dr.Sirisup Laohakiat)

Title	EXAMINING PERFORMANCE OF IDINVERT GAN MODEL ON IMITATING REAL HUMAN FACES
Author	PATTANADEJ CHAENGSRISUK
Degree	MASTER OF SCIENCE
Academic Year	2022
Thesis Advisor	Dr. Napa Sae-bae

This study evaluates the performance of IDInvert, a variant of the generative adversarial network (GAN) models, in terms of ability to generate synthetic face images that resemble real ones, while also preserving personal identity. The main focus of the study is to investigate whether current techniques can detect the subtle differences between real and synthetic face images generated by the IDInvert model. The findings reveal that although the IDInvert model produces highly realistic facial images, they do not preserve personal identity, and they can be identified using feature extraction techniques and standard classification models. Overall, the study highlighted the potential risks of using GAN inversion models and emphasized the importance of developing more robust and secure algorithms to prevent the misuse of such technology.

Keyword : Generative model, GAN, GAN inversion, Facial recognition, Fake face detection

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor, Asst. Professor Dr. Napa Sae-bae, for priceless feedback and warm patience. I am also deeply grateful to my thesis committee for generously contributing knowledge and expertise.

I am also thankful to my classmates for their kind advice, inspiration, and cheerfulness. Our time spending is very memorable.

Lastly, I would like to recognize my family for deep motivation and support. My success is for you all.

PATTANADEJ CHAENGSRISUK

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background

Generative models can generate realistic non-existing content like images and videos. In particular, generative adversarial networks (GAN) (Goodfellow et al., 2014) that takes a random vector as an input and generates facial image that resemble samples in training data have been rapidly developed during recent years and still continually improved. The recent GAN variations can generate amazingly realistic face images with even higher resolution, such as PGGAN (Karras, Aila, Laine, & Lehtinen, 2017) and BigGAN (Brock, Donahue, & Simonyan, 2018), and some of the latter architectures even disentangle the manipulation on each attribute and let the generated facial images features, such as ages, pose, gender and expression, can be separately edited. Examples of this identity-preserving GAN model are style-based GAN family (Karras, Laine, & Aila, 2019), (Karras, Laine, et al., 2020) and (Karras et al., 2021).

These GAN technologies have raised the possibility of many useful visual applications. However, it also brings the concern on critical abuses and even severe criminals, as these fake contents plausibly deceive human eyes and recognition. In addition, with identity-preserving GAN model, it can be widely abused and harmfully impacts a target person.

Although fake detection techniques had been long developed for decades, those traditional techniques that were developed are ineffective when dealing with contents generated from deep learning networks (Marra, Gragnaniello, Verdoliva, & Poggi, 2019), (Neves et al., 2020). Novel techniques are therefore required to detect those GAN-generated contents. (Tolosana, Vera-Rodriguez, Fierrez, Morales, & Ortega-Garcia, 2020)

Considering the wrestling between efforts to improve GAN illusion ability and efforts to catch up it, This study is set to examine the model called IDInvert (J. Zhu, Shen, Zhao, & Zhou, 2020b) which is one of the GAN inversion (Xia et al., 2022). Given the face photos of the target persons, this GAN inversion model converts the photos into latent vectors that pretrained GANs can use to regenerate identity-preserving images.

1.2 Objectives of the Study

These GAN technologies have raised the possibility of many useful visual applications. However, it also brings the risk concern of critical abuses and even severe criminals, as these fake contents plausibly deceive human eyes and recognition. In addition, with identity-preserving GAN model, it can be widely abused and harmfully impacts a target person.

Although fake detection techniques had been long developed for decades, those traditional techniques that were developed are ineffective when dealing with contents generated from deep learning networks (Marra, Gragnaniello, et al., 2019), (Neves et al., 2020). Novel techniques are therefore required to detect those GAN-generated contents. (Tolosana et al., 2020)

The objective of this study is to investigate the performance of this GAN inversion model on synthesizing realistic face images and to develop the fake detection technique that can effectively classify the GAN inversion generated facial images from the real ones.

1.3 Scope of the Study

In this study, one of the state-of-the-art GAN inversion models, IDInvert (J. Zhu et al., 2020b) is used to regenerate real face photos from LFW dataset. Figure 1 shows examples of images regenerated by IDInvert.

LFW dataset contains 13,233 faces images of 5,749 individuals with rather diversified races and skin colors, and thus allows the test without bias on a narrow group of races or skin colors. It also contains face images with various gesture, including both straight poses, left poses, and right poses of the same persons, which enables the test on personal identity preservation. Figure 2 shows sample photos of LFW.

This LFW dataset will be used as real images and as reference images to generated synthetic images in the following two experiments.

To investigate the performance of this GAN inversion model on generating realistic face images, similarity test is designed to verify IDInvert capability to preserve the personal identity of target faces by comparing the distance between real photos with different poses and the distances between real photo and regenerated images with the same pose. The test result will demonstrate how close synthetic images are to real images.

To develop a fake detection technique that can effectively classify the GAN inversion generated facial images from the real ones. The set of the features of each face image will be extracted from pretrained CNN networks and combined with frequency domain analysis features (Durall, Keuper, Pfreundt, & Keuper, 2019). This feature vector will then be used as an input for a multilayer perceptron network classifier where it is trained to discriminate those regenerated images from the real ones.

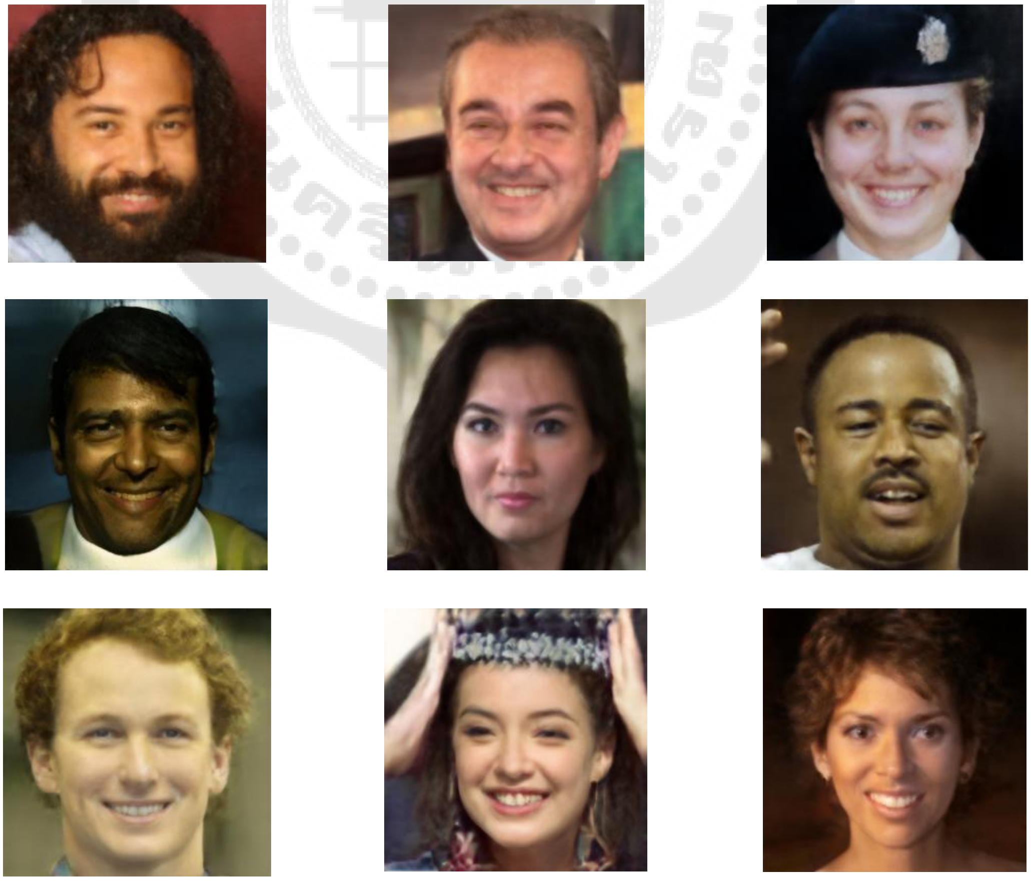


Figure 1 Samples of IDInvert-generated images

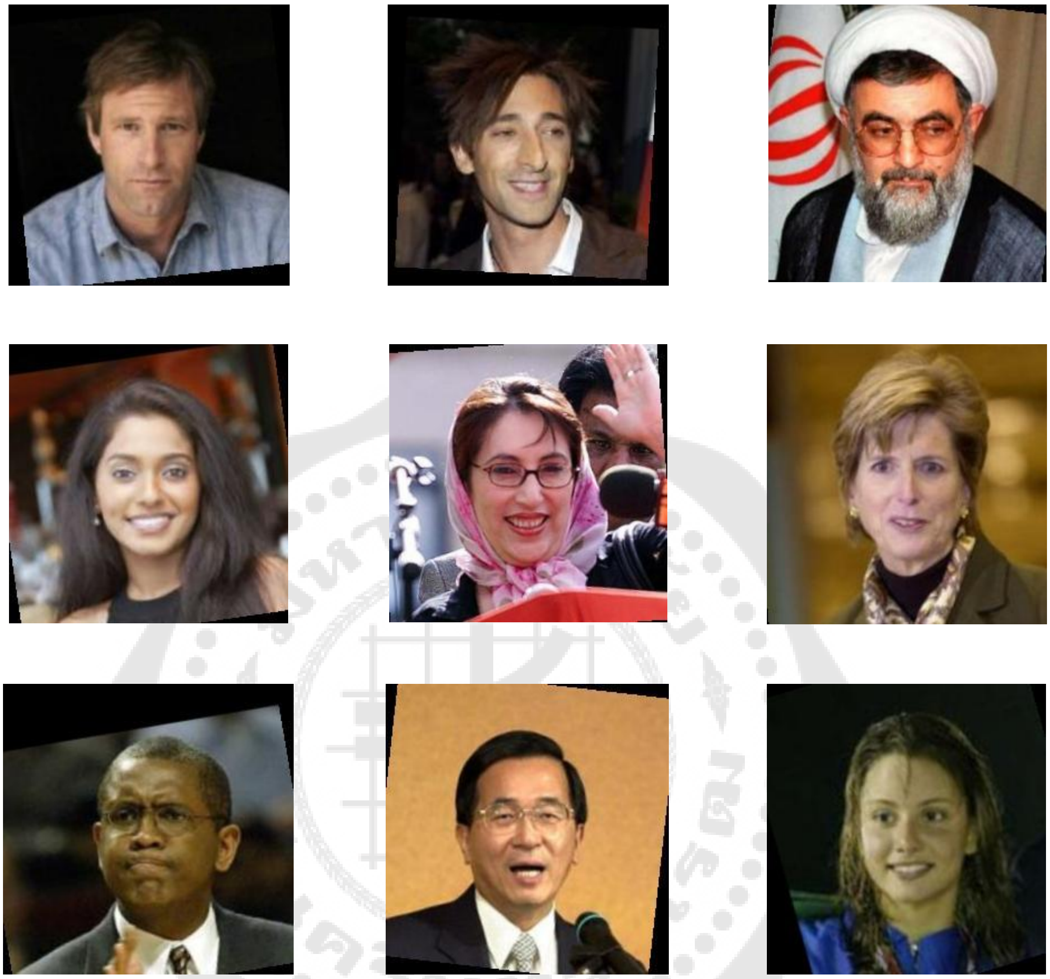


Figure 2 Samples of LFW face images

CHAPTER 2
LITERATURE REVIEW

In this chapter, previous works on three topics: generative adversarial networks (GAN), GAN inversion, and fake face detection techniques, are reviewed. In the first topic, the definition and the general components of GANs, including the issues that may arise in GAN model training phrase, are described. In addition, the interesting GAN architectures are sequentially described. These models became the foundation of style-based GANs, which enabled semantic editing on generated images.

In the second topic, GAN inversion techniques, which were developed to exploit capability of the recently developed StyleGAN (Karras, Aittala, et al., 2020) and enable realistic manipulation on real images are discussed. In particular, the properties, approaches, application tasks, and examples of GAN inversion models are described. The selected GAN inversion model for this study, the IDInvert, is also elaborated in detail.

In the third topic, the types of fake face manipulation techniques and the recent techniques which were developed to tackle GAN-generated fake faces are briefly described. Some examples of the fake face detection techniques against the entire face synthesis are discussed. Lastly, the generalizability limitation of the current fake face detection approaches is discussed. Details of these topics are follows.

2.1 Generative Adversarial Networks

Generative adversarial networks model (GAN) (Goodfellow et al., 2014) was first introduced as a subclass of generative models and widely obtained attention for achieving realistic image generation. A GAN basically consists of the two linked multi-layered neural networks, namely the generator (G) and the discriminator (D). G has a structure simply contrary to an encoder that converts images into array. It conversely generates images from random numbers, so called latent vectors. Latent vectors are embedded in the multi-dimensional space, which is called the latent space.

While D could be simply considered as a binary classifier which is responsible for checking whether an image is real or generated. Figure 3 shows the general components of GAN architectures.

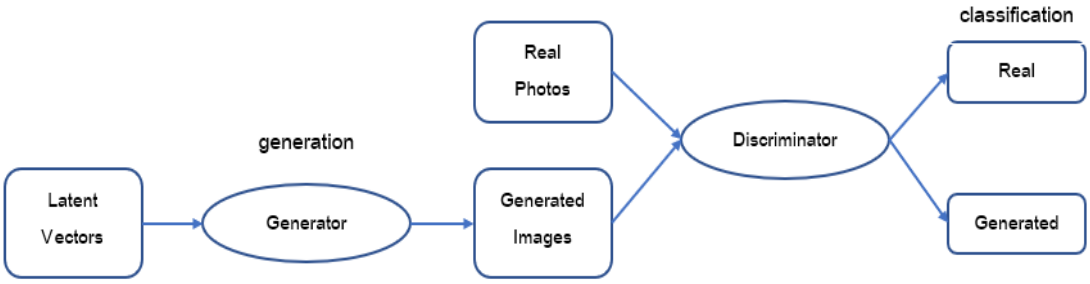


Figure 3 Overview of GAN architectures

To train a GAN, the G and D network are trained in an adversarial manner. At first, the weights of G are frozen, and it could not yet generate realistic images. D is trained to classify which images are real or not, by using a labeled set of real images and generated images from the unlearned G. The loss function for training D was designed to minimize errors that D wrongly identifies the real images and the generated ones, so that weights of D are adjusted to learn how real images would look like.

Then, the trained D is then fixed, and G is trained to generate more realistic images and deceive D. G loss function is set to maximize the chances that D wrongly classifies generated images as real images. The training process is repeated for some numbers of epochs until the stop criteria is met. This could be until the generator produces output that is indistinguishable from real data, or until the performance on the validation set stops improving.

2.1.1 Problems of GAN

The original GAN is considered hard to train, since there are serious problems that can result in training failure (Goodfellow, 2016).

1) **Non-convergence** It is the situation where trained parameters could not find the stable equilibrium between optimizing G and D loss function, or the weights of G and D are not converging.
2) **Vanishing Gradients** This problem is that D is highly effective in discriminating samples generated by G and would not be fooled by G. In this case, D would not give useful feedback to G for adjusting weights and optimize the generation. (Arjovsky & Bottou, 2017)
3) **Mode Collapse** In this problem, G pays too much attempts to overcome D by generating only a small set of realistic images. Generated images are therefore alike and loss diversification comparingly to the train images.

2.1.2 Variations of GAN architectures

Successive works proposed the changes in network architectures, loss functions and training steps for developing in training stability, image diversification. The latter GANs can generate more realistic and larger images. Some of the improved GAN versions are as follows: -

1) **cGAN** (Mirza & Osindero, 2014) The conditional GAN was proposed as an early revision of the original GAN. Conditional components were added, and labels were used in training the GAN. So that the images could be conditionally generated by desired condition.

2) **AC-GAN** (Odena, Olah, & Shlens, 2017) The auxiliary-classifier GAN (AC-GAN) was an extension of conditional GAN. The module that also outputs the labels of generated images was added.

3) **DCGAN** (Radford, Metz, & Chintala, 2015) As the network structures of the original GAN was similar to common image encoders and decoders, G and D initially consisted of max pooling layers and fully-connected layers. The Deep

convolutional GAN (DCGAN) was introduced with changes in convolutional layers. This resulted in increasing quality of generated images and training stability. DCGAN was widely accepted as successful GAN architecture and became the basis for the latter models. The proposed changes were as follows.

(1) Max pooling layers in G and D were replaced by the transpose convolution layers and the convolution layers consecutively.

(2) Fully connected layers were removed both in G and D

(3) Batch normalizing layers were added both in G and D

(4) Regarding the activate function, ReLU was used in G, and LeakyReLU was use in D.

4) **Improved Techniques for GAN training** (Salimans et al., 2016) This paper suggested the following techniques to encounter non-convergence and mode collapse problem. The techniques were mostly dealing with the loss function of G and D as follows.

(1) **Feature matching** The objective of training G was adapted to prevent from overtraining on the same current D. Instead, the new loss function required G to generate images that matched the statistics of real images, which were identified by D.

(2) **Minibatch discrimination** This technique was designed to resolve mode collapse and prevent the model generating images that resemble each other. Training images are packed into minibatches for training D collectively, instead of processing each images individually. during. This is to avoid the lower distribution of generated images.

(3) **Historical averaging** This technique was to add historical averaging terms to the loss function of both G and D with the objective to help the training achieved convergence better.

(4) **One-sided label smoothing** In this technique, smoothing on the positive labels in D loss function was deployed to improve the stability of the adversarial model during the training process.

(5) **Virtual batch normalization** This can be regarded as the extension of batch normalization proposed by DCGAN paper. Particularly, statistics of the reference batch was used to normalize input data instead of the individual batch's statistics.

5) **EBGAN** (Zhao, Mathieu, & LeCun, 2016) The energy-based GAN (EBGAN) was developed based on the concept of LeCun's energy-based model (LeCun & Huang, 2005). The discriminator in the adversarial model was modified to calculate the energy function and use as the loss function. To achieve better stability, the architecture of an auto-encoder was used in the discriminator. This method can be seen as an early version of the loss function, implemented to enhance stability of the training process.

6) **Unrolled GAN** (Metz, Poole, Pfau, & Sohl-Dickstein, 2016) To stabilize GAN training and decrease opportunity of mode collapse, this paper proposed modification on the training procedures with the unrolled optimization of the discriminator. G's parameters would be updated more often than D's parameters by the defined numbers of unrolled step. This was to prevent G overfitting any current version of D.

7) **WGAN** (Arjovsky, Chintala, & Bottou, 2017) This paper also encouraged major change in GAN modeling. Wasserstein GAN (WGAN) adopted the concept of the Earth Mover distance (EM distance) and used the so-called Wasserstein distance as the loss function. WGAN was trained to make the distribution of generated images close to the distribution of real images. The network called 'critic' was used in place of D. The critic's architecture was alike D, but it

returned a scalar value instead of the probability that an image was real or not. This scalar value takes roll as a score measuring quality of generated images. WGAN was reported to improve training stability and get rid of mode collapse and became another foundation for latter works.

8) **LSGAN** (Mao et al., 2017) The Least Squares GAN (LSGAN) paper proposed to use the least squares in place of the sigmoid cross entropy for D's loss function. The architectures of G and D consisted of convolutional layers like DCGAN. The model was reported to generate images with better quality and achieve more training stability.

9) **BEGAN** (Berthelot, Schumm, & Metz, 2017) The architecture of Boundary Equilibrium GAN (BEGAN) was much alike the mentioned EBGAN (Zhao et al., 2016) in the way that an auto-encoder was used as a discriminator. Reconstruction losses of the autoencoder were calculated for both real images and generated images. The distance between reconstruction losses is then measured by the Wasserstein distance, which is used as a loss function for GAN training. The model was argued to better provide fast and stable training and able to produce high image quality.

10) **WGAN-GP** (Gulrajani, Ahmed, Arjovsky, Dumoulin, & Courville, 2017) The Wasserstein GAN with gradient penalty (WGAN-GP) was an improved version of original WGAN (Arjovsky et al., 2017). The improvement focused on the loss function. The authors proposed to use the term 'gradient penalty' in the loss function to enforce the Lipschitz constraint, in place of the weight clipping. This was because the weight clipping formular potentially led to either non-convergence or vanishing gradients problems.

11) **SN-GAN** (Miyato, Kataoka, Koyama, & Yoshida, 2018) The spectrally normalized GAN (SNGAN) could be considered as a revision of WGAN (Arjovsky et al.,

2017). The paper focused on improving the loss function of the critic network. They argued that the proposed spectral normalization term could outperform the previous training stabilization techniques, e.g., weight normalization (Salimans & Kingma, 2016), weight clipping used in the original WGAN, and gradient penalty in WGAN-GP (Gulrajani et al., 2017), for controlling the Lipschitz constraint.

12) **PGGAN** (Karras et al., 2017) The progressive growing GAN (PGGAN) paper seemed to make another milestone on GAN modeling. This paper proposed the progressive growing technique for architecture and the training procedures. The convolution layers were designed to have increasingly higher resolution from 4x4, 8x8, 16x16, 32x32 up to 1024x1024, and were added and trained both in G and D networks gradually. The PGGAN could generate images with high resolution up to 1024x1024, which was the largest size ever.

    The authors deployed the minibatch discrimination technique (Salimans et al., 2016) to encounter mode collapse and the WGAN-GP loss function (Gulrajani et al., 2017) for training stability. Their new techniques, the equalized learning rate and the pixelwise feature vector normalization are also introduced for preventing unhealthy competition between G and D also.

13) **SAGAN** (H. Zhang, Goodfellow, Metaxas, & Odena, 2019) The Self-Attention GAN (SAGAN) was developed mainly to solve weakness of the prior GANs regarding generating the images with structural complication, e.g. four-legged animals. The self-attention module, which was previously used in NLP models (Parikh, Täckström, Das, & Uszkoreit, 2016), (Cheng, Dong, & Lapata, 2016), was added to the G network. This was to deal with the relationships between widely separated spatial regions better, while typical convolutional GANs did not perform well on this. SAGAN was highly based on SN-GAN (Miyato et al., 2018) and used the spectral normalization term in loss function too. The two-timescale update rule (Heusel, Ramsauer, Unterthiner, Nessler, & Hochreiter, 2017), which

was proposed to use different learning rates in training G and D, was also deployed for the more stabilized and faster training.

14) **BigGAN** (Brock et al., 2018) The large-scale GAN, BigGAN, was another phenomenon. It could generate broad range of image classes with high quality and high resolution up to 512 x 512. (While other high-resolution GANs, e.g., PGGAN, focused only on face images.) The BigGAN architecture was based on SAGAN (H. Zhang et al., 2019). It also employed the self-attention modules in G's networks and the spectral normalization terms in the loss function for stabilization purpose.

The major modification was in the G layout. The blocks of batch normalization layers, upsampling layers, and convolutional layers (so called residual block, ResBlock, collectively) were in place of the regular convolutional layers. Each ResBlock was related to a step of resolution growing. Therefore, the more numbers of ResBlocks were added, the generated images got the higher resolution.

It is also interesting that the latent vector z was first split into many chunks and before feeding to each ResBlock. This is different from typical GANs where the whole latent vectors were fed to models with no partition.

15) **AEGAN** (Guo et al., 2019) This model is particularly designed to generate high resolution images. The autoencoder was deployed to extract the global structure features of real images and combine them into the latent embedding for training the generator. The adversarial denoiser was also used to refine generated images by removing visual artifacts. AEGAN could generate high resolution images up to 512x512 and was claimed to consume significantly less training time than PGGAN (Karras et al., 2017) while having quite close performance.

16) **MSG-GAN** (Karnewar & Wang, 2020) The multi-scale gradient GAN (MSG-GAN) was developed mainly to improve training stability for high resolution image generation. There were two architectures which were simply the variation of PGGAN (or ProGAN) (Karras et al., 2017) and StyleGAN (Karras et al., 2019), namely MSG-ProGAN and MSG-StyleGAN respectively. The basic components, including G, D, and loss function. Each version was like either PGGAN or StyleGAN. This design enables the model to generate up to 1024x1024 images. However, the MSG-ProGAN was not trained in the progressive growing scheme like the original PGGAN, as all the layers were trained at the same time.

The core features of MSG-GAN were the direct links between all the counterpart layers of G and D. The 8x8 deconvolution layer in G was connected to the 8x8 convolution layer in D and so on. This direct connection was inspired by architecture of the U-net (Ronneberger, Fischer, & Brox, 2015), which was initiated for biomedical image segmentation works. The connection allowed the transfers of gradient descents to train G layers at every scale simultaneously, and therefore stabilized the learning of higher scale layers better.

17) **InterFaceGAN** (Shen, Gu, Tang, & Zhou, 2020) The InterFaceGAN was not a GAN itself. Literally, it was an extension module which helped a GAN better pick up latent codes. It enabled disentanglement of face semantics and allowed the face attributes i.e., pose, age, gender, eyeglasses to be edited independently. With the InterFaceGAN, image manipulation can be performed on the face images generated by conventional GANs in the same way as images generated by StyleGAN (Karras et al., 2019). It could also operate with GAN inversion models to reconstruct and semantically edit the real face images.

18) **StyleGAN** (Karras et al., 2019) The first version of StyleGAN might be the biggest game changer in GAN works. As prior works just focused on improving stability, convergence, image diversification and resolution, StyleGAN emphasized on

how to make attributes of generated images could be further edited independently. It was thus mainly designed to disentangle the latent codes and enable the sperate controls over style generation. The introduction of StyleGAN sparked the findings of techniques to utilize the ability to perform realistic manipulation on real images, which later became GAN inversion.

The StyleGAN architecture was based on PGGAN (Karras et al., 2017). The discriminator network (so called the critic) and the loss function were adopted from the WGAN-GP paper (Gulrajani et al., 2017). The progressive structure of convolution layers was also inherited and enabled the generation of high-resolution images up to 1024x1024 pixels.

A major modification was added to the generator. Inspired by studies on style transfer, the random latent code z was fed to the network, called the mapping network, instead of the generator network like other GANs. The mapping network then converted the input codes z to the intermediate code w. The intermediate code w was not directly fed to the generator also, but was passed to the affine transformation algorithms to be further converted into the ‘styles’ (y). The style y variables were then distributed to the adaptive instance normalization modules (AdaIN) (X. Huang & Belongie, 2017) that were embedded in every convolution block of each resolution level.

One of the most prominent features of StyleGAN is the operation called ‘style mixing’ which enables generating new face images by mixing attributes of two face images. This mechanism allows mixing of the coarse styles (including pose, gender, age, races, and skin color), the middle styles (general hair styles, face shape and eyeglasses, and background color), and the fine styles (eye color, hair color, detailed hair styles) from the original images. This separable style control is then encoded in the style (y) variables fed to AdaIN.

19) **StyleGAN2** (Karras, Laine, et al., 2020) A year after the first StyleGAN paper, another paper on StyleGAN proposing the improved architecture and training

processes. This was motivated by the observed blob-like artifacts in background areas of generated images, and the problem that face details, like teeth, did not move correspondingly to the whole face when changing pose.

The first problem urged modification in detailed design of the generator's blocks, especially splitting AdaIN components, refining normalization and modulation algorithms, and adding weight demodulation operation into every block.

The latter problem was solved by redesigning the connection between counterpart layers of G and D to transfer gradient descents during the training process. The original StyleGAN applied progressive growing method derived from PGGAN, but it was pointed out that causing the said problem. StyleGAN2 was therefore switched to the skip connection scheme which was inspired by MSG-GAN (Karnewar & Wang, 2020) .

20) **StyleGAN2-ADA** (Karras, Aittala, et al., 2020) Training the StyleGAN families is not that cheap. It requires both the large enough dataset of high-resolution images and the incredibly powerful processing units. Therefore, the special version of StyleGAN2 with the adaptive discriminator augmentation mechanism, so called StyleGAN2-ADA, was proposed. Its objectives were to stabilize the training with limited dataset and to make StyleGAN better for generating images in other domains (besides human face). The concept of this technique was roughly the same as general image augmentation techniques to increase additional samples by adapting the existing images.

21) **StyleGAN3** (Karras et al., 2021) This latest version of StyleGAN was just introduced in 2021 and seems yet to be well known. The problem that motivated this work could be observed when interpolating a set of the same face with different poses, angles, or positions. In other words, it is a problem of transition animation rather than a single image, so called texture sticking. It is the situation

that small attributes, especially hair, moustache, or textile patterns are unintentionally fixed to some certain pixels and do not move to other pixels correspondingly to shift of the whole image. This problem makes the image transition look unrealistic.

The problem might not be seen as critical but required complicated solutions. To eliminate texture sticking problem, it required the concepts of positional references and the ‘equivariance’ property of GAN networks. The generator architecture was almost reengineered. The main improvement objectives were to remove all causes of positional refences and to enhance either translation equivariance or rotation equivariance property of the networks.

2.2 GAN Inversion

GAN inversion, or GAN embedding, is currently one of the emerging AI fields. Its purpose is to enable realistic manipulation on real images. GAN inversion models utilize the capability of recently developed GANs, especially StyleGAN (Karras, Aittala, et al., 2020), which allows independent editing attributes of generated images e.g., pose, age, gender, eyeglasses of generated faces.

To edit real images using GANs, many techniques were designed to invert real images into codes that GANs can further use to regenerate like those generated images. However, the inversion is not the same as the feature extraction used in image classification works. Finding of appropriate area in latent space and to enable semantic reconstruction and effectively maintain attribute disentanglement is still challenging.

2.2.1 Properties of GAN inversion methods

The important properties of GAN inversion methods were proposed by (Xia et al., 2022) as follows.

1) **Resolution of reconstructed images** The Capacity to reconstruct high resolution images with satisfied quality depends both on the employed GAN and inversion algorithms. Recently developed GANs, especially PGGAN (Karras et al., 2017) and StyleGAN, could generate high resolution images up to 1024x1024 pixels. However, the utmost utilization of ability to generate that high resolution requires appropriate mapping the target images to the latent spaces. This results in currently ongoing search of inversion approaches and architectures.
2) **Semantic awareness** Sematic seems to be the most important thing for the realisticness. It may be considered as the attempt to maintain meaningfulness of the whole image while the model generates a single pixel. The well-designed GAN inversion methods will reconstruct real images as if they know what objects are lying on and how the whole objects should look like after editing. In other words, GAN inversion models with the sematic awareness should let generated faces smile, turn left-right, and get younger-older while still looking like real human faces.

3) **Being layerwise** This means how easy the model is designed to be tracked which layer was processing the signal data. Indeed, this concept has attracted research attention for a long time because deep learning networks was perceived as a magic box. Many studies were also conducted for clear understanding how GANs worked and how to make them better. Though the continuous performance improvement led to more and more complicated architectures, GANs and GAN inverters are also expected to be layerwise and tractable which network layers potentially cause problems.
4) **Out-of-distribution generalizability** This requires GAN inverters could invert wide ranges of image categories even though the inverter and the employed GAN were trained by image sets of specific domains. For example, a StyleGAN-based inverter, which was trained mainly on human face datasets, can also effectively invert animal images or geographic scenes, like sky and ocean, and even allow semantic manipulation. This is challenging but also unjustifiable. The model is expected to semantically generate objects that it has never known nor specialized. Generalizability would be very useful in situations with limited data and training resources.

2.2.2 Types of GAN inversion methods

During the last few years, lots of GAN inversion models have been proposed. They could be categorized into 3 approaches, namely learning-based models, optimization-based models, and hybrid models (Xia et al., 2022).

1) **Learning-based GAN Inversion models** This approach is generally based on training the encoder networks to map real images into latent codes. Many architectures and training procedures in this framework have been proposed, including pSp (Richardson et al., 2021), e4e (Tov, Alaluf, Nitzan, Patashnik, & Cohen-Or, 2021), ReStyle (Alaluf, Patashnik, & Cohen-Or, 2021), E2Style (Wei et al., 2022), High-fidelity GAN inversion (T. Wang, Zhang, Fan, Wang, & Chen, 2022), HyperInverter (Dinh, Tran, Nguyen, & Hua, 2022) etc.

2) **Optimization-based GAN Inversion models** The optimization approach does not train the encoder's weights but directly optimizes the latent codes to catch up with the target images. This method normally begins with initializing latent codes and then performs an optimization process to achieve the best result. It thus consumes higher processing power and more computers' memory. However, it is accepted to return better quality than the encoder approach. The optimization-based models include Image2StyleGAN (Abdal, Qin, & Wonka, 2019), Image2StyleGAN++ (Abdal, Qin, & Wonka, 2020), mGANPrior (Gu, Shen, & Zhou, 2020), Editing in Style (Collins, Bala, Price, & Susstrunk, 2020), StyleGAN2 Distillation (Viazovetskyi, Ivashkin, & Kashin, 2020) and MimicGAN (Anirudh, Thiagarajan, Kailkhura, & Bremer, 2020), StyleSpace (Wu, Lischinski, & Shechtman, 2021), BDInvert (Kang, Kim, & Cho, 2021), and Improved StyleGAN Embedding (P. Zhu, Abdal, Qin, Femiani, & Wonka, 2020) etc.

3) **Hybrid GAN Inversion models** This approach blends the advantages of the above two approaches by using the encoder to initialize the latent vector for further optimization. The encoder helps facilitate the optimization process to begin with the more potential latent vectors. It thus accelerates the inversion while maintaining satisfied image quality. The proposed hybrid models include GANSeeing (Bau et al., 2019), GANPaint (Bau et al., 2020), IDInvert (J. Zhu et al., 2020b), GANEnsembling (Chai, Zhu, Shechtman, Isola, & Zhang, 2021), PTI (Roich, Mokady, Bermano, & Cohen-Or, 2022), and HyperStyle (Alaluf, Tov, Mokady, Gal, & Bermano, 2022)

2.2.3 Application of GAN inversion

Ability of GAN inversion to reconstruct target images and allow realistic editing can be applied to many tasks e.g., image manipulation, image interpolation, image restoration etc. (Xia et al., 2022).

1) **Image manipulation** The specific attributes of regenerated images, such as pose, expression, age, gender, hair style and eyeglasses of face images, can be edited via linear algebra operation over the certain latent codes. This benefits from the StyleGAN capability to disentangle image attributes and let a single attribute being edited independently from others. The target face can smile wider, turn to be male, or get younger while keeping other features the same and still looking realistic.
2) **Style transfer** This is like changing a woman's hair style to another style or changing a car's color to another color while keeping other features the same. It also benefits from the style-mixing technique of StyleGAN, which enables selective lending either low-level, middle-level, or high-level styles from one image to another images. Also, it is based on mathematical operations over the corresponding latent codes.
3) **Image restoration** The GAN inversion is also applicable to repair damaged photo. This exploits the model's ability to normalize defect pixels by learning semantics of the entire images and replace them realistically. The colorization techniques for adding colors to pale or black & while photos for refreshment and the super resolution techniques for increasing resolution for better clearness are also based on GAN inversion.
4) **Image interpolation** The gradual morphing between two given images can be performed by interpolating their corresponding latent vectors. The latent codes of in-between images are somewhat the weighted average value of the two target images. The weight of the first image gradually moves from 0.0 to 1.0, and vice versa for another image.

2.2.4 Examples of GAN inversion models

During the last few years, there were many GAN inversion papers published. Some of the interesting works are described as follows:

1) **Image2StyleGAN** (Abdal et al., 2019) The Image2StyleGAN was one of the early GAN inversion models and was published shortly after the introduction of StyleGAN (Karras et al., 2019). It proposed a pure optimization method that the initial latent vectors were directly optimized. It used a pre-trained generator and combination of the VGG-16 perpetual loss and the pixel-wise MSE loss against the target images. Regarding choices of initial vector, the authors tried using both a random vector and vector of the average face and found the latter was the better choice.

   The choices to embed optimized codes into the StyleGAN generator as the initial codes or as the disentangled intermediate codes were also experimented. The authors decided to embed as intermediate codes, which would be distributed to AdaIN modules in eighteen convolution layer blocks of StyleGAN generator.

   The improved technique, the Image2StyleGAN++ (Abdal et al., 2020), was later published in 2020. Still keeping pure optimization scheme, the authors proposed 3 ways of enhancement. The first was to employ noise optimization as a complement to latent code optimization. The second way was to extend the global latent space embedding to enable local embeddings. Finally, it was to use the activation tensor manipulation for high-quality editing.

2) **IDInvert** (J. Zhu et al., 2020b) The in-domain GAN inverter (IDInvert) is one of hybrid GAN inverters. It uses an encoder network to convert input images into initial codes and pass to the optimizer for improvement. (So, a single image is processed at a time.) The authors mainly focused on how to blend the semantic domain of the train images into the generation, instead of just redefining the value of individual pixel. The authors designed a GAN inverter that recognizes

what kind of objects that the images were presenting. Their proposed 'in-domain' codes were described as semantically meaningful and subject to the semantic domain of training data.

To regenerate input images at both pixel level and semantic level, the domain-guided encoder and the domain-regularized optimizer were deployed. The domain-guided encoder's structure was not different from typical encoders used in previous GAN inverters. The main difference was that it was trained using both the pretrained generator and the pretrained discriminator (while other encoders were trained just by a pretrained generator on its generated images). Therefore, the domain-guided encoder learned from real images i.e., was guided by the domain of real images, and could pass on the domain information to inverted codes.

And to further refine the inverted codes, the learned domain-guided encoder also participated in the optimization process through the additional term of the objective function. Therefore, the codes would be optimized in the way that semantic domain was still maintained.

3) **E2Style** (Wei et al., 2022) This learning-based GAN Inverter proposed the encoder with well-designed architecture that did not require further optimizing inverted codes. It refined the codes with training iterations, called multi-stage refinement. The encoder applied the modules of average pooling layers and fully connected layers, called Efficient Head. The Efficient Heads separately processed the code from each layer of different resolution and inverted every feature levels equally well.

Regarding the loss functions, the two new loss terms, namely multi-layer identity loss and multi-layer face parsing loss, were combined with normal loss function to improve quality of the inversion.

The following Table 1 summarizes the mentioned GAN inversion techniques.

Table 1 Examples of GAN inversion models and their performance

Model	Reference	Inversion Approach	Image dataset	Best metric result
Image2StyleGAN	(Abdal et al., 2019)	optimization-based	FFHQ (Karras et al., 2019)	distance between the inverted code and the code of average faces = 30.67
Image2StyleGAN++	(Abdal et al., 2020)	optimization-based	FFHQ	PNSR* = 45 dB
IDInvert	(J. Zhu et al., 2020b)	hybrid	FFHQ, LSUN	FID** = 42.6
E2Style	(Wei et al., 2022)	learning-based	CelebA (Ziwei Liu, Luo, Wang, & Tang, 2015), FFHQ	FID = 49.4

** Peak signal-to-noise ratio (Hore & Ziou, 2010)*

***Fréchet Inception Distance (Heusel et al., 2017)*

2.3 Fake Face Detection

There has been the need for detection techniques to distinguish fake contents for long, especially fake face images and videos. Fake contents can be abused harmfully and even used to commit crimes like fake news and deceiving biometric systems. Studies on detecting fake contents were conducted long before the deep learning era. And they successfully dealt with fake faces that were created by traditional techniques like photomontage.

However, the rise of GANs widely sparked social anxiety with the capability to create even more realistic fake contents. The latter variations of GANs can generate non-existing human faces with high resolution and can be edited while keeping realisticness. These GAN-generated contents can fool human eyes, and the traditional detection techniques would not effectively cope with.

Recently, there consequently seems to be continual competition between the improved GANs and searches of techniques to detect fake contents. Research on detecting GAN-generated faces was highly active. Many new detection techniques against GAN images were proposed during the last few years. Some of them highly achieved to classify real and fake face images reportedly.

2.3.1 Types of fake face manipulation techniques

In the context of fake face detection, fake face manipulation techniques on using GAN models could be categorized in 4 types, namely entire face synthesis, identity swap, attribute manipulation, and expression swap. The detection approaches for these techniques are proposed differently. (Tolosana et al., 2020)

1) **Entire face synthesis** This technique creates the whole face images of non-existing people by using GANs. The recent GAN architectures e.g., PGGAN (Karras et al., 2017), BigGAN (Brock et al., 2018), StyleGAN (Karras et al., 2019) and MSG-GAN (Karnewar & Wang, 2020) can generate highly realistic faces. These generated faces can easily pretend to be real people and can be used as profile pictures in any social network platforms.

2) **Identity swap** This manipulation aims to replace certain persons' faces in videos with others' faces. It was made by traditional computer graphic techniques formerly. But the invention of deep learning networks led to the new technique, so called DeepFakes, that has presented amazing performance on the fake videos of celebrities saying and doing what they had never really done.
3) **Attribution manipulation** Also known as face editing or face retouching, this is to modify certain face attributes e.g., hair style, skin color, gender, age and eyeglasses, by using GAN variations such as StarGAN (Choi, Uh, Yoo, & Ha, 2020). It can be applied for selling cosmetics or hair style products, and the well-known application, FaceApp (Warzel, 2019), also deploys this manipulation.
4) **Expression swap** Also called face reenactment; this is to switch face expression of certain persons in motion pictures with face expression of the others. Face2Face is one of the most popular techniques in this category. It is different from identity swap that the expression swap just modifies the facial expression of a person in that video, while identity swap is like dropping the target person into the venue that he does not really appear.

Since this study focuses on capability of GAN inversion technique to reconstruct real faces, the fake detection techniques against the entire face synthesis, which are considered fit for measuring its performance, are explored.

As mentioned above, several novel techniques mainly to detect GAN-generated images, introduced just in the past few years, illustrate the ongoing competition against works on improving GANs and increasing realisticness of generated images. As many fake detection techniques utilize the artifacts, so called fingerprint, attached to synthesize images to distinguish real and fake faces, one of improvement direction of GANs is to remove these artifacts, which are originated from image generation processes.

2.3.2 Examples of works on fake face detection

The interesting fake face detection techniques, which were recently proposed to detect GAN-generated faces are briefly described as follows.

1) **Color Cues** (McCloskey & Albright, 2019) This paper argued that GANs formed the images' colors differently from those of the photos produced by cameras. This was due to unique statistical relation among pixel values of synthesized images, which was caused by generation architectures. Based on this concept, the authors employed two techniques, namely color image forensics and saturation-based forensics to classify camera photos and GAN-generated images.

2) **Attribution Networks** (Yu, Davis, & Fritz, 2019) This paper took advantages of the prior study (Marra, Saltori, Boato, & Verdoliva, 2019) that the GAN-generated images also got specific patterns, called fingerprint. The fingerprint could be used to distinguish them from the camera-generated photos. Attribution networks therefore were introduced to extract those fingerprints from image sets, which were used as the features for the classifiers.

3) **Image Spectrum** (X. Zhang, Karaman, & Chang, 2019) This technique was based on the concept that GANs normally deployed upsampling operation of transposed convolution layers to increase resolution of generated images, which would result in different patterns of the frequency spectrum from those of real images. The authors adopted Discrete Fourier Transform algorithms to compute the normalized spectrum of real and fake images, which then passed to the image classifier networks as input features.

4) **Convolutional Traces** (Guarnera, Giudice, & Battiato, 2020) This technique was also based on the upsampling operation to generate images from latent

codes in GANs, and the hidden correlation pattern between pixels of generated images, so called the convolutional traces, thus would reflect the generator’s architecture. The Expectation Maximization (EM) algorithms were deployed to extract those convolutional traces and convert them into feature vectors for classification.

5) FakeSpotter (R. Wang et al., 2019) This technique proposed to use a simple neural network for classifying real and generated images. However, the feature for classification was not just the vectors extracted from images. The authors of FakeSpotter argued that the behavior of the feature extractor networks would be different when converting real and fake images into vectors. This difference was thus used to distinguish images. The module called mean neuron coverage (MNC) was thus introduced for capturing the layer-by-layer neuron activation behaviors of the CNNs in VGG-Face, to use as feature vectors for the classifier.

6) Frequency Domain Analysis (Durall et al., 2019) This paper relied on the concept of frequency domain analysis in the signal processing theory and proposed to deploy Discrete Fourier Transform (DFT) algorithm and Azimuthal Average to extract DFT power spectrum from each image. As the power spectrum of real images and fake images was found to be significantly different at the higher level of spatial frequency, it could be therefore used as features for supervised classifiers, like SVM and logistic regression models, and unsupervised K-Means clustering also.

Table 2 summarizes the mentioned works on fake face detection. Most techniques utilize some hidden properties of generated images, of which the differences from those of real images could be observed, as input features for the classifiers, rather than directly classify raw images.

2.3.3 Limited generalization over different GANs

There was some evidence showing that a single fake detector did not achieve the same performance over fake images generated by the different models of GANs. The work of (Zhengzhe Liu, Qi, & Torr, 2020) conducted the experiments on their proposed classifier using fake images generated by PGGAN and StyleGAN. Their experiments were conducted both in the 'in-domain' setting (Training images and testing images were generated by the same PGGAN or by the same StyleGAN.) and the 'cross-GAN' setting (Training images were generated by PGGAN and testing images were generated by StyleGAN and vice versa.). The results obviously showed the lower performance in case of the 'cross-GAN' setting.

Although it was not clearly concluded whether this was because of the specific fingerprint patterns of each GAN architecture, it might be considered that the different design of the generator network and image generation processes of GAN models would result in different signals and information hidden in generated images. And it should be aware that the fake detection model which is trained using images generated by a GAN will not always achieve the same performance when used to test images generated by a different kind of GAN.

However, since the scope of this study focuses on performance of the IDInvert GAN inversion model and does not aim to propose the generalizable fake detector, the experiments here are conducted in the 'in-domain' setting of the IDInvert.

Table 2 Related work on fake face detection.

Model	Technique	Dataset	GAN model	Best result
Color Cues (McCloskey & Albright, 2019)	Color image forensics (INH), Saturation-based forensic (SVM)	LSUN[1], ImageNet[2]	PGGAN[3]	AUC = 0.70
Attribution Networks (Yu et al., 2019)	Attribution Networks to extract GAN fingerprint from images to use as feature vectors for the classifier	LSUN, CelebA[4]	PGGAN, SNGAN[5], CramerGAN[6], MMDGAN[7]	accuracy = 99.50%
Image Spectrum (X. Zhang et al., 2019)	Discrete Fourier Transform algorithms to compute the normalized spectrum of real and fake images and use as feature vectors for the classifier	CycleGAN dataset[8]	CycleGAN[8]	accuracy = 98.70%
Convolution Traces (Guarnera et al., 2020)	EM algorithms to extract Convolutional Traces from images to use as feature vectors for the classifier	CelebA	StarGAN[9], StyleGAN[10]	accuracy = 99.81%
FakeSpotter (R. Wang et al., 2019)	MNC module to extract neuron activation behaviors from CNN extractor to use as feature vectors for the classifier	CelebA, FFHQ[10]	PGGAN, StyleGAN	accuracy = 98.60%
FDA classifier (Durall et al., 2019)	Discrete Fourier Transform (DFT) algorithm and Azimuthal Average to extract DFT power spectrum from each image and use as feature vectors for the classifier	CelebA, FFHQ	DCGAN[11]	accuracy = 100.00%

[1](L. Wang, Guo, Huang, Xiong, & Qiao, 2017), [2](Russakovsky et al., 2015), [3](Karras et al., 2017),[4](Ziwei Liu et al., 2015), [5] (Miyato et al., 2018), [5](Miyato et al., 2018), [6](Bellemare et al., 2017), [7](Li, Chang, Cheng, Yang, & Póczos, 2017), [8](J.-Y. Zhu, Park, Isola, & Efros, 2017), [9](Choi et al., 2020), [10](Karras et al., 2019), [11](Radford et al., 2015)

CHAPTER 3
RESEARCH METHODOLOGY

The main purpose of this study is to examine how well GANs currently perform on disguising human faces, which would further indicate the possibility to abuse GAN-generated contents harmfully. And since GAN inversion techniques currently present the most effective way to replicate real faces while allowing realistic manipulation on reconstructed faces, the study focus on how close these reconstructed faces get to original faces, and how well fake detection techniques can distinguish them. This leads to the two experiments on images regenerated by the GAN inverter from real image dataset, i.e., the similarity test and the fake face detection test.

3.1 Dataset

In this study, the experiments are conducted using the sets of original face photos ('real images') and 'fake' images regenerated by the GAN inverter from those real images. The original image dataset is from the University of Massachusetts's Labeled Faces in the Wild database (LFW dataset) (G. B. Huang, Mattar, Berg, & Learned-Miller, 2008), (G. B. Huang & Learned-Miller, 2014), which contains totally 13,233 faces images of 5,749 individuals with various gesture. The LFW dataset is considered suitable for the experiments since it stores images of each person separately. It also collected images of people with diversified races and skin colors, and thus allowed face replication test without bias on a narrow group of races or skin colors. And photo sets of many persons include both straight poses, left poses, and right poses of the same persons. This enables the test on personal identity preservation. Table 3 shows the distribution of the numbers of each person. Although most people have gotten only a few photos, there still be over one hundred persons with more than 10 photos.

Table 3 Numbers of photos per person in LFW dataset

No. of images per person	Head count
Not over 10	5,606
11 - 30	111
31 - 50	20
51 - 100	7
over 100	5
total people	5,749

3.2 GAN Inversion Model

A GAN inversion model used in this study is the in-domain GAN inverter (IDInvert) (J. Zhu et al., 2020b). It is one of hybrid GAN inverters which conceptually uses an encoder network to convert input images into initial codes and further pass to be refined by an optimizer (a single image therefore can be processed at a time), before passing to pretrained StyleGAN (Karras et al., 2019) for regeneration. With capability to disentangle attributes of a face image, the features like age, eyeglasses, gender, pose, and expression of regenerated can be adjusted with only slight impact on each other.

Although, IDInvert is not currently the latest GAN inversion model, since research on this field has been highly active and novel methods keep being proposed during recent years, it still be accepted as a state-of-the-art architecture, and latter works have not significantly raised the new standard yet. In addition, as the ready-to-use version is publicly accessed, it does not require deep knowledge nor skill to be abused and is therefore interesting for testing its performance on replicating real faces. And due to time constraint and limited processing resources, the pretrained model is deployed here. Figure 4 shows examples of IDInvert regenerated images and their reference photos from LFW dataset.

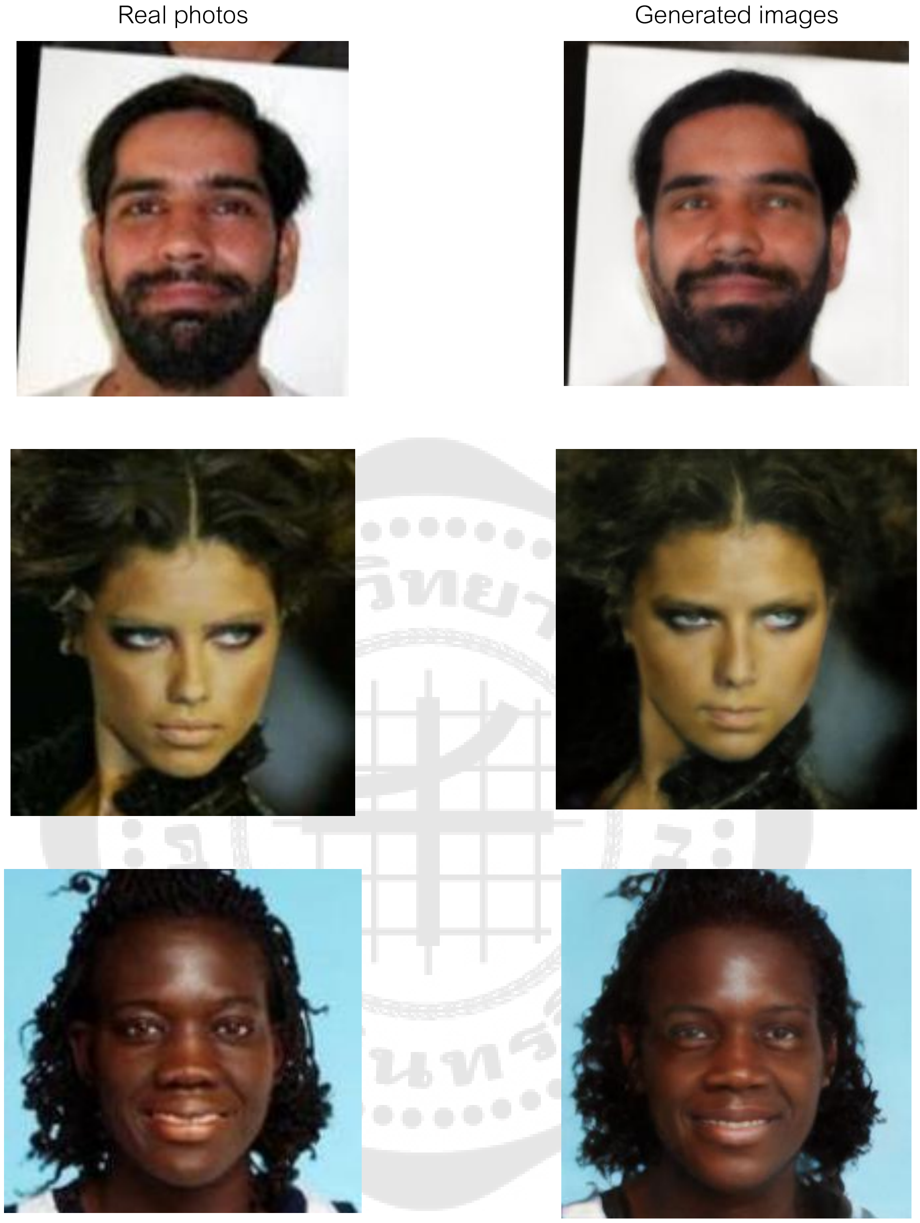


Figure 4 LFW real images versus fake images regenerated by IDInvert

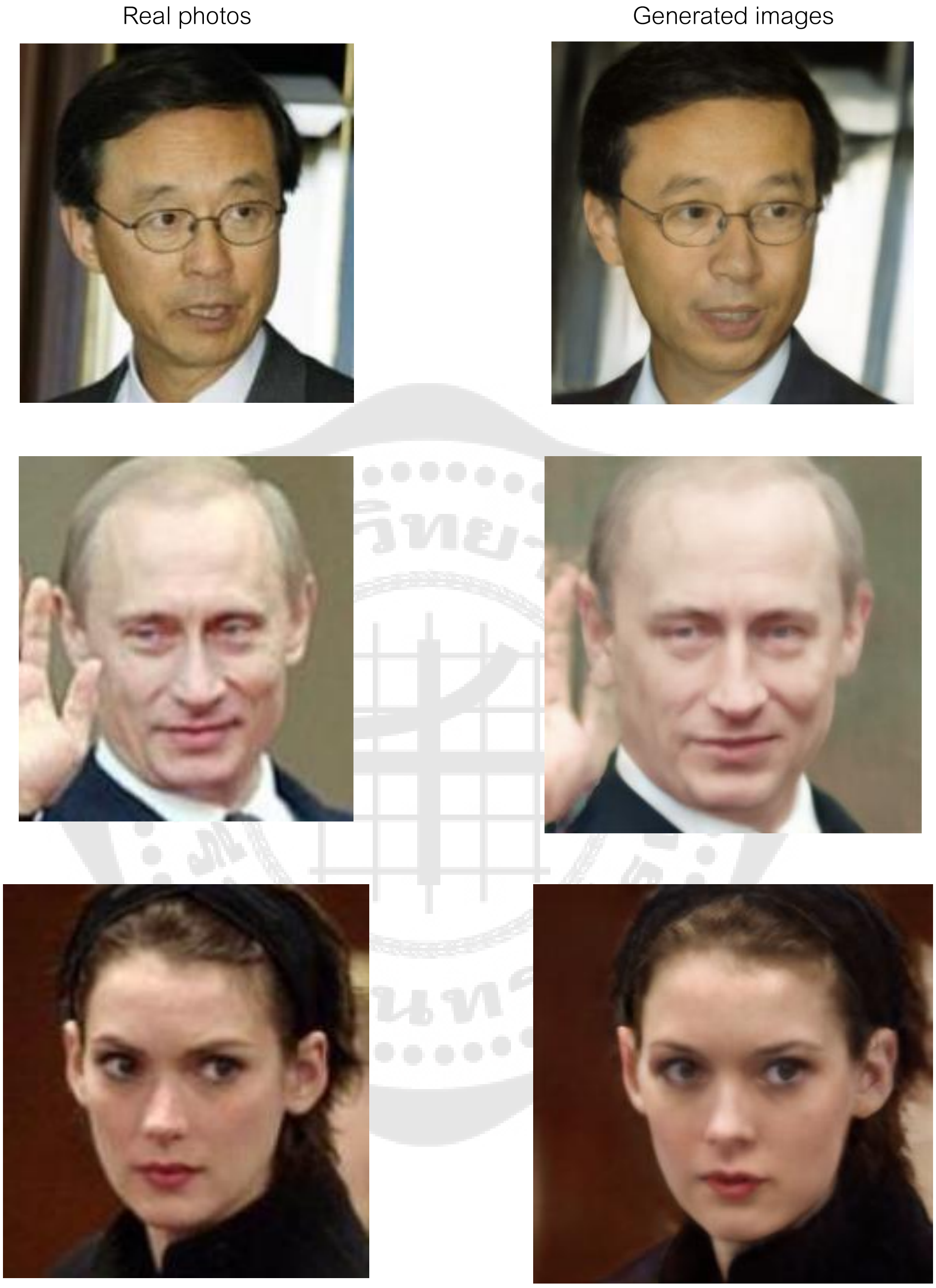


Figure 4 LFW real images versus fake images regenerated by IDInvert(cont’d)

3.3 Experiments

The questions of this study lead to the two experiments on images regenerated by the IDInvert from LFW dataset, namely the similarity test and the fake face detection test.

3.3.1 Similarity test

This experiment is mainly to find out how well generated images replicate real faces, in other words how well a GAN inverter preserves personal identity. An output is comparison of the distances between real photos with actual right / left pose and real photos with straight pose, and the distances between real photos with actual right / left pose and regenerated images with manipulated right / left pose.

This starts from selecting real photos with various gestures, including straight pose, left pose and right pose of each person. Then, straight-pose photos are regenerated by IDInvert. The regenerated images with straight pose are adjusted into right-pose images and left-pose images, using the manipulation model. The 'fake' right-pose images and are paired with the real right-pose photos of the same person, and so as the left-pose ones, to calculate cosine distances. Figure 5 illustrates this test process.

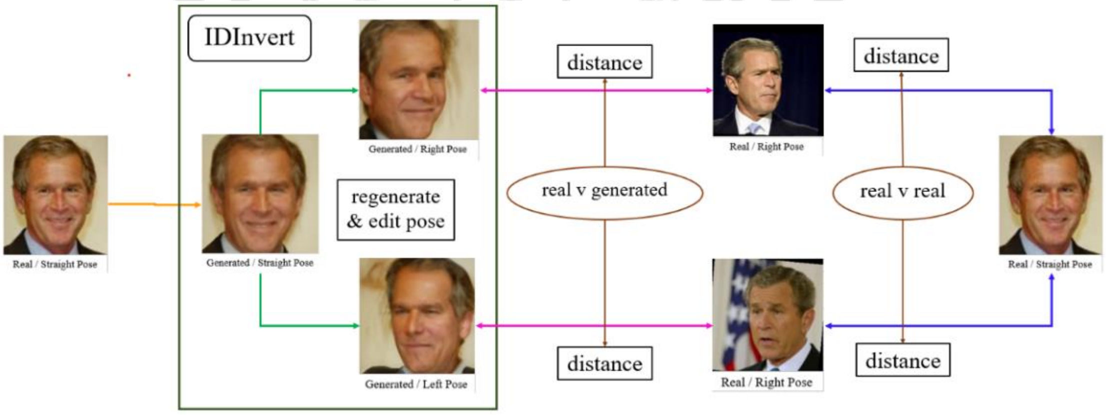


Figure 5 Illustration of the experimental procedure for the similarity test

To perform the experiment, photos of 35 persons with various gestures are selected from LFW face dataset, based on diversified races and skin colors. The set of totally 304 real photos, contains 70 straight-pose, 137 left-pose and 97 right-pose photos. Those real straight-pose photos are used to regenerate 167 fake images, containing 92 left-pose and 75 right-pose images for testing. Table 4 shows the number of image pairs to calculate cosine distance. Only the pairs of the same persons are considered, while those of different persons' images are excluded. Figure 6 shows the samples of paired images.

Table 4 Numbers of image pairs used in the similarity test

		No. of pairs
Pairs of Real Photos – Real Photos		
Real / Left-Pose	Real / Straight-Pose	331
Real / Right-Pose	Real / Straight-Pose	198
	total real – real	529
Pairs of Real Photos – Fake Images		
Real / Left-Pose	Fake / Left-Pose	460
Real / Right-Pose	Fake / Right-Pose	244
	total real - fake	704

In this test, the cosine distance between image vectors is used to represent distance between images. To calculate the cosine distance, the face area of all images are first detected and captured by MTCNN (K. Zhang, Zhang, Li, & Qiao, 2016) to exclude background and other objects and let the classifiers focus only on faces.

The cropped faces are then converted by the VGGFace module (Parkhi, Vedaldi, & Zisserman, 2015), (Cao, Shen, Xie, Parkhi, & Zisserman, 2018), with the Microsoft's Resnets50 (He, Zhang, Ren, & Sun, 2016) into vectors. The cosine distance between the image vectors are then calculated using numerical operations over NumPy arrays.

Cosine Distance

The cosine distance is derived from cosine similarity, which is used to measure similarity between the two vectors. Its invention was related to the domain of vector space model and information retrieval system. The formula of cosine distance and cosine similarity are as follows (Singhal, 2001).

$$cosine\ distance\ (D,Q) = 1 - cosine\ simialirty\ (D,Q)$$

$$cosine\ similarity\ (D,Q) = \frac{D\ \cdot Q}{\|D\|\|Q\|} = \frac{\sum_{i=1}^{n} D_i Q_i}{\sqrt{\sum_{i=1}^{n} D_i^2}\sqrt{\sum_{i=1}^{n} Q_i^2}}$$

where D_i and Q_i are the components of vector D and vector Q respectively.

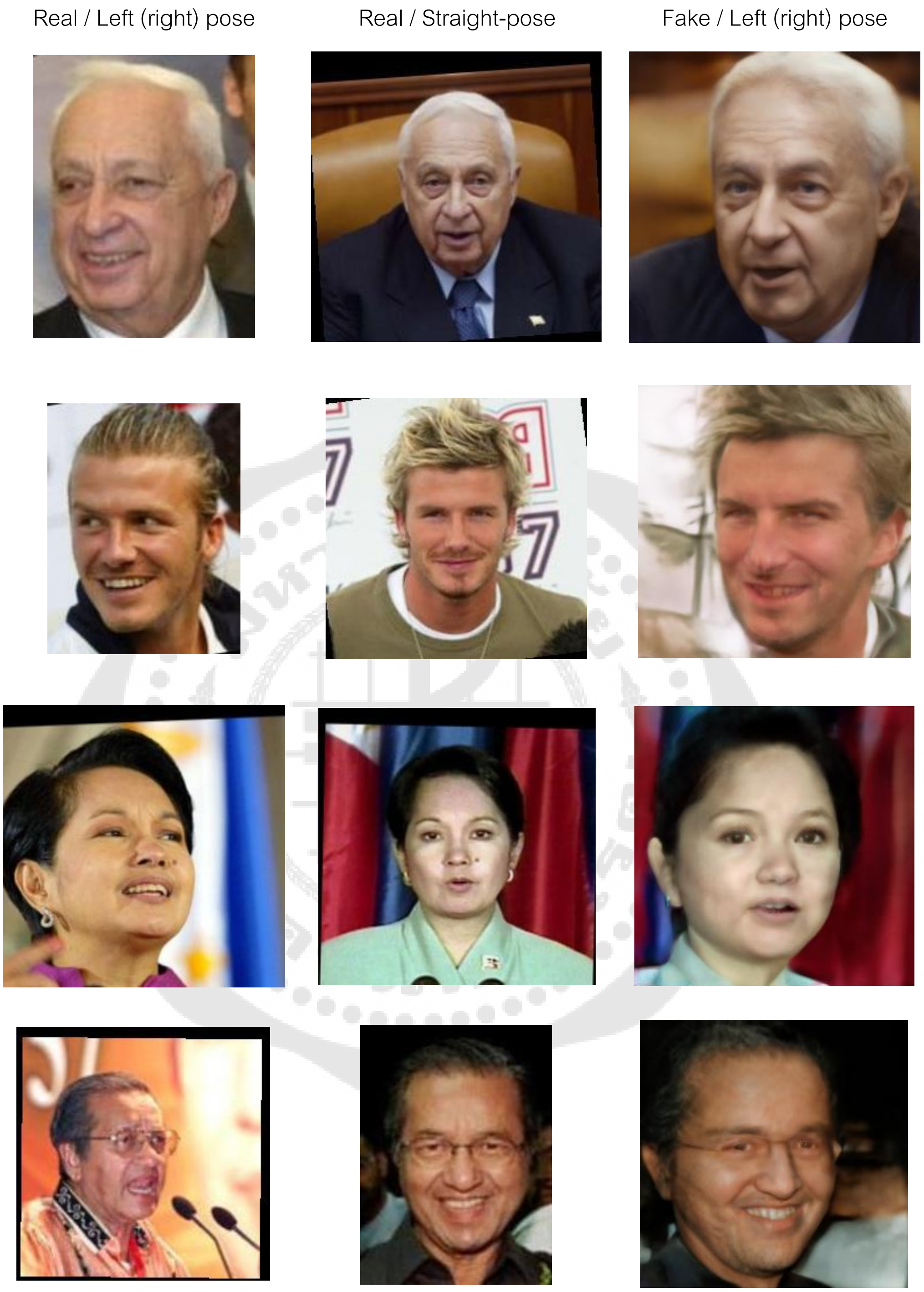


Figure 6 Samples of image pairs used to compare cosine distances.

3.3.2 Detectability test

This experiment aims to figure out how well the images regenerated by a GAN inverter can be distinguished from real photos. To answer the question, the combination of feature extraction techniques and classification models are adopted to classify the real photos from LFW dataset, and their counterpart images regenerated by IDInvert. In addition, image filtering techniques are also applied to generated images of the test set, to verify how the trained classification models would perform when facing the modified fake images.

This experiment is conducted through the following processes.

(1) Regenerate real photos using IDInvert model.

(2) Separate train images and test images

(3) Conduct image filtering on the test images.

(4) Detect and capture face area using MTCNN

(5) Extract features using CNNs and frequency domain analysis technique.

(6) Train classification models and test with non-filtered and filtered images.

(1) **Image regeneration** First, 5,749 real photos of every individual person in the LFW dataset are selected, one photo per person, to regenerate fake images by the IDInvert model. However, 595 real photos (approximately 10.3%) fail to be regenerated, and there are 5,154 images successfully regenerated. The failure real photos are excluded to make the numbers of samples in each class are equal. Therefore, there are 10,308 images (5,154 real photos and 5,154 generated images) in the dataset for this experiment.

It should be noticed that those 595 failure real photos consist of people with quite various characteristics, genders, age, poses, and races. It is thus not clear which specific attributes of face images would cause the regeneration failure. The error yet seems to be random. Figure 7 shows the samples of unsuccessful-regeneration of IDInvert.

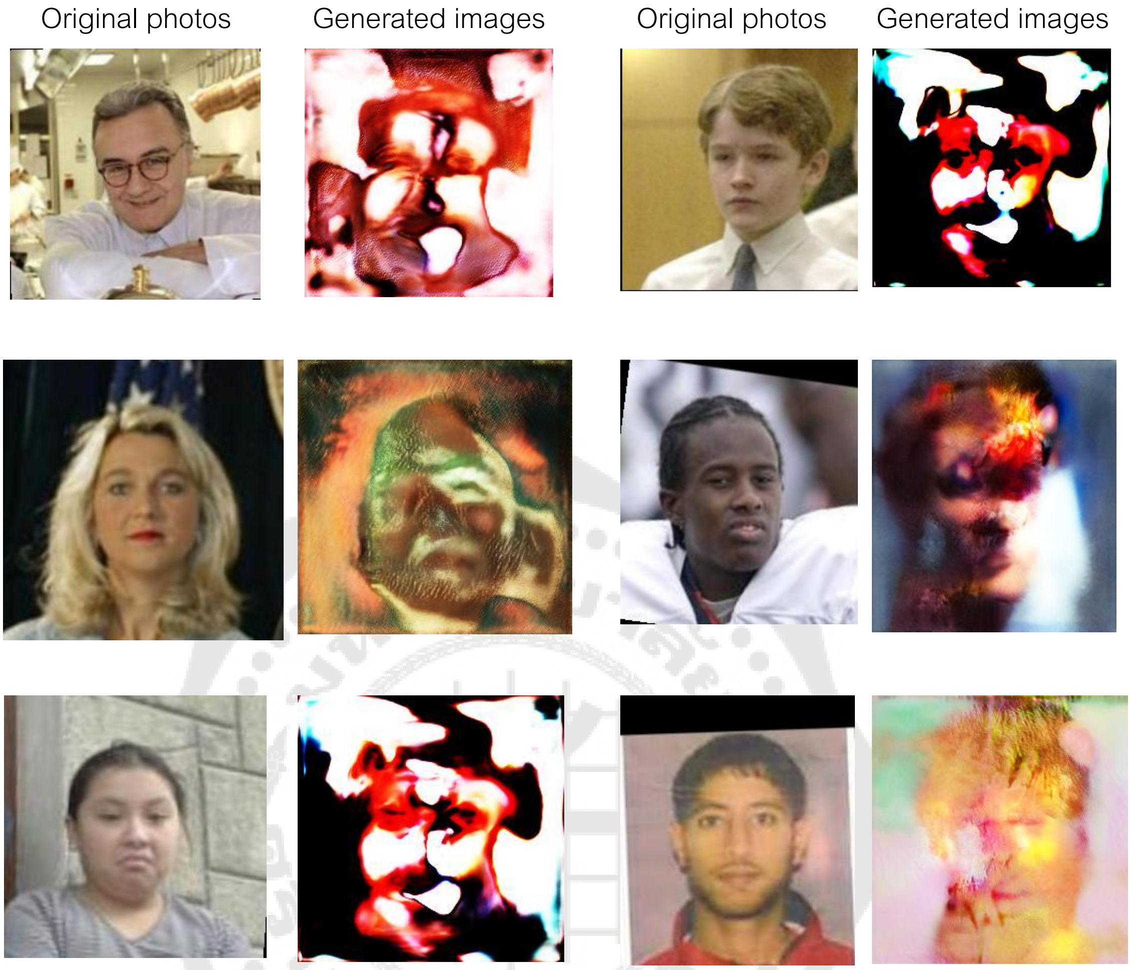


Figure 7 Samples of images unsuccessfully regenerated by IDInvert.

(2) **Train-test split** The dataset then is split, in portion of 70% to 30%, into the train set with 7,214 images and the test set with 3,094 images. The numbers of real photos and generated images are equal in each group. Therefore, there are 3,607 and 1,547 real photos in the train set and the test set consecutively, so as generated images. Real photos and generated images of the same persons are in the same group (train set or test set), in order not to let the test data leak into the training process. Table 5 summarizes the numbers of real and generated images in the train set and the test set.

Table 5 Numbers of images in the train set and the test set of the detectability test.

	Real Photos	Generated Images	Total
Train set	3,607	3,607	7,214
Test set	1,547	1,547	3,094
Total	5,154	5,154	10,308

(3) **Image filtering** This step is to prepare additional test sets to examine the effect of image filtering on the classification models' performance. These sets of original real photos and filtered generated images are further processed in the same steps as the unfiltered test sets and are also used to test the classification models, which are trained using the unfiltered train set. Variation of the test results will show how the models would perform when encountering a fake image that is modified by filtering techniques.

Those 1,547 generated images of the test set are blurred and sharpen by using image filtering techniques (Gonzalez, 2009), (Lukac & Plataniotis, 2018), while real photos are not changed. The test images are therefore extended into 5 sets, so called no-filter (original test set), blur4x4, blur8x8, sharpen, and sharpen2 according to the applied filters. Table 6 summarizes the 5 scenarios of model testing with the 5 test sets. And Figure 8 shows the samples of those unfiltered and filtered images.

Table 6 Image filtering techniques applied to a generated test set.

Scenario	Train set	Test set	Filtering techniques
1	No filter	No-filter	n/a
2	No filter	Blur4x4	blur with the kernel size (4, 4)
3	No filter	Blur8x8	blur with the kernel size (8, 8)
4	No filter	Sharpen	Sharpen with kernel [[0, -1, 0], [-1, 5, -1], [0, -1, 0]]
5	No filter	Sharpen2	Sharpen with kernel [[-1, -1, -1], [-1, 9, -1], [-1, -1, -1]]

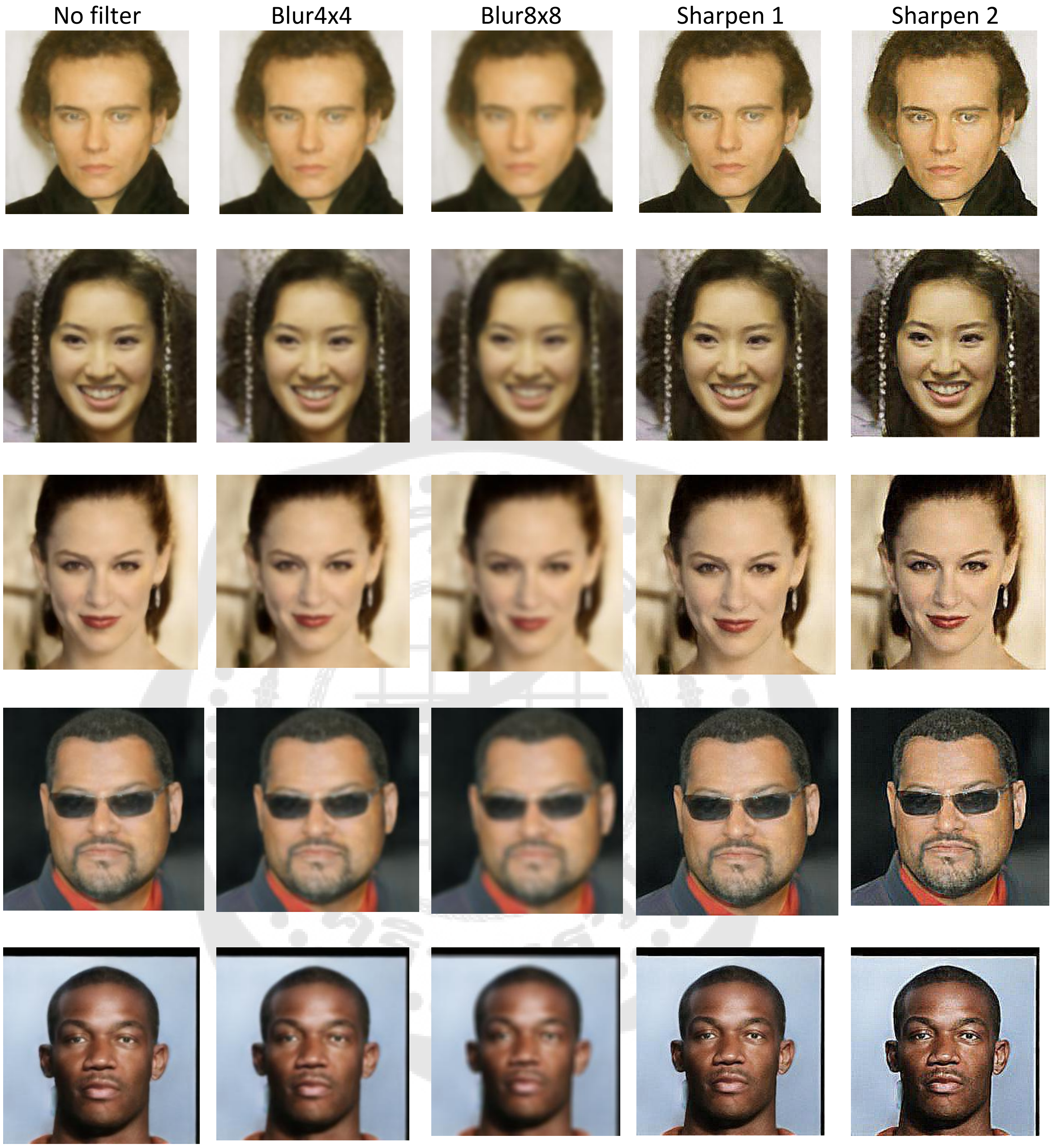


Figure 8 Samples of blurred and sharpen generated images.

(4) **Face detection** Second, the face area of each image is detected and captured by MTCNN (K. Zhang et al., 2016) to exclude background and other objects and let the classifiers focus only on the main face of each image.

However, MTCNN may also capture other faces in the same image or other stuff that it detects as human faces. Thus, the output of MTCNN must be cleaned up. Only the main face of each image is used in the further processes.

(5) **Feature extraction** Next, features of all detected faces are extracted before further classification. In this study, convolutional neural network (CNN) models and frequency domain analysis technique are used as feature extractors for comparison. Since this study focuses on face images, VGGFace module (Parkhi et al., 2015), (Cao et al., 2018) with the CNN models namely ResNet50 (He et al., 2016) and SeNet50 (Hu, Shen, & Sun, 2018) are deployed to convert images into feature vectors.

Regarding the frequency domain analysis technique, the use of Discrete Fourier Transform algorithm and Azimuthal Average proposed by (Durall et al., 2019) is used to extract power spectrum of each image. The power spectrum is reported to be significantly different between real photos and GAN-generated images, and the classification models which were trained by power spectrum presented superior performance on distinguishing real photos and GAN-generated images. Therefore, there are 3 sets of features i.e., ResNet50 vectors, SeNet50 vectors, and power spectrum.

(6) **Classification models** Each set of Feature vectors and power spectrum is then used to train and test the group of Classification models. In this study, 5 sets of Classification models including multi-layer perceptrons (MLP), Support Vector Machine (SVM) with polynomial kernel, SVM with linear kernel, SVM with and RBF kernels, and Logistic Regression (LR) models, are comparatively trained and tested to examine how each model will perform in cases of different image filtering and feature extraction techniques. Table 7 lists the models in use and their configuration. The parameters of SVM and LR are based on configuration of (Durall et al., 2019).

Table 7 Classification models in use and configuration

Model	Configuration
MLP	Layer: 2 dense layers with rectified linear unit function (ReLU) and logistic function (sigmoid) Optimizer: adaptive moment estimation (Adam) learning rate: 0.001 loss function: binary cross-entropy
SVM-poly	kernel: polynomial (poly)
SVM-linear	kernel: linear
SVM-rbf	kernel: radial basis function (rbf) C: 6.37 gamma: 0.86
LR	solver: liblinear max_iter: 1,000

All these 5 models are separately trained using 3 sets of image features, the ResNet50 vectors, the SeNet50 vectors, and the power spectrum. This combination therefore returns the different 15 Classification models as shown in Table 8. Each model is then tested by the sets of unfiltered and filtered test images as mentioned above.

Table 8 Combination of feature extraction techniques and Classification models

Model	Train Features	Classifier
1	ResNet50	MLP
2		SVM-linear
3		SVM-ploy
4		SVM-rbf
5		LR
6	SeNet50	MLP
7		SVM-linear
8		SVM-ploy
9		SVM-rbf
10		LR
11	Power spectrum	MLP
12		SVM-linear
13		SVM-ploy
14		SVM-rbf
15		LR

Finally, Figure 9 recapitulates the processes of the Detectability test.

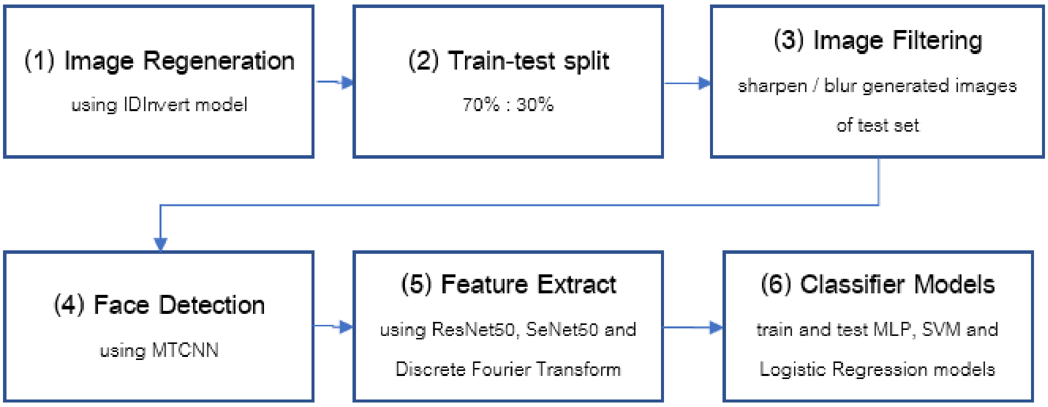


Figure 9 Summary of the Detectability test processes

CHAPTER 4
RESULT

In this chapter, experiment results of the similarity test and the fake face detection test are described as follows.

4.1 Results of Similarity test

Table 9 shows the average cosine distances between 529 pairs of real - real images and 704 pairs of real - generated images. The 529 pairs of real - real images consist of 331 pairs of real/left pose images - real/straight pose images and 198 pairs of real/right pose - real/straight pose images.

The 704 pairs of real - generated images consist of 460 pairs of real/left pose - generated/left pose images and 244 pairs of real/right pose - generated/right pose images. As the dataset contains more left pose images than right pose images, the number of image pairs of left pose images (real vs generated) is higher than number of image pairs of right pose images (real vs generated).

Table 9 Average distances and standard deviation of the similarity test

Pairs of Images			Numbers of Pairs	Avg. Cosine Distance	Standard Deviation
Real vs Real					
Real/Left Pose	vs	Real/Straight Pose	331	0.267	0.076
Real/Right Pose	vs	Real/Straight Pose	198	0.264	0.080
Total Real vs Real			529	0.266	0.077
Real vs Generated					
Real/Left Pose	vs	Generated/Left Pose	460	0.458	0.097
Real/Right Pose	vs	Generated/Right Pose	244	0.446	0.087
Total Real vs Generated			704	0.453	0.094

The average cosine distance of the 529 pairs of real - real images is 0.266, and the average cosine distances of the 331 pairs of real/left pose - real/straight pose images and of the 198 pairs of real/right pose - real/straight pose images are 0.267 and

0.264 respectively. While the average cosine distance of 704 pairs of real - real images is 0.453, and the average cosine distance of 460 pairs of real/left pose – generated/left pose images and of the 244 pairs of real/right pose – generated/right pose images are 0.458 and 0.446 respectively.

4.2 Results of Detectability test Result

As described in the previous chapter, this experiment totally obtains 15 Classification models by using 3 techniques (ResNets50, SeNet50 and power spectrum) to extract image features for training the 5 Classification models (MLP, SVM-Poly, SVM-linear, SVM-rbf and LR). The 15 classification models are then tested by the 5 sets of test images (no filter, blur4x4, blur8x8, sharpen and sharpen2) to examine the effect of image filtering. The accuracy, precision, recall, F1 score, and the area under ROC curve (AUC) results of each model are presented from Table 10 to Table 12. And Figure 10 shows ROC curves off the top performing models.

Regarding the ResNet50 models, the accuracy results are mostly in the range of 0.79 – 0.87 (except for SVM-rbf). There is not much difference between precision, recall and F1 score of the same models since the predicted results do not fall heavily into one class (except for SVM-rbf). And the AUC results are in the range of 0.85 – 0.94 for most models (except for SVM-rbf).

In the case of SeNet50 models, the MLP performance results (including accuracy, precision, recall, F1 score and AUC) are not much different. But in cases of SVM and LR models, accuracy drops to 0.49 – 0.50 for all filtered test sets. The larger differences between precision and recalls are because the test images are almost classified as real photos and fewer generated images are correctly detected. That results in the decrease of AUC too.

For the power spectrum models, the results are quite inconsistent. In the case of the no-filter test sets, the MLP performs poorly while SVM and LR perform quite well. However, all the models perform poorly on the blur test set, but perform very well on the blur8x8, sharpen and sharpen2 test sets (except for the SVM-poly model on sharpen2 test set that does not perform quite well).

Table 10 Performance of classifiers trained by the ResNet50 vectors

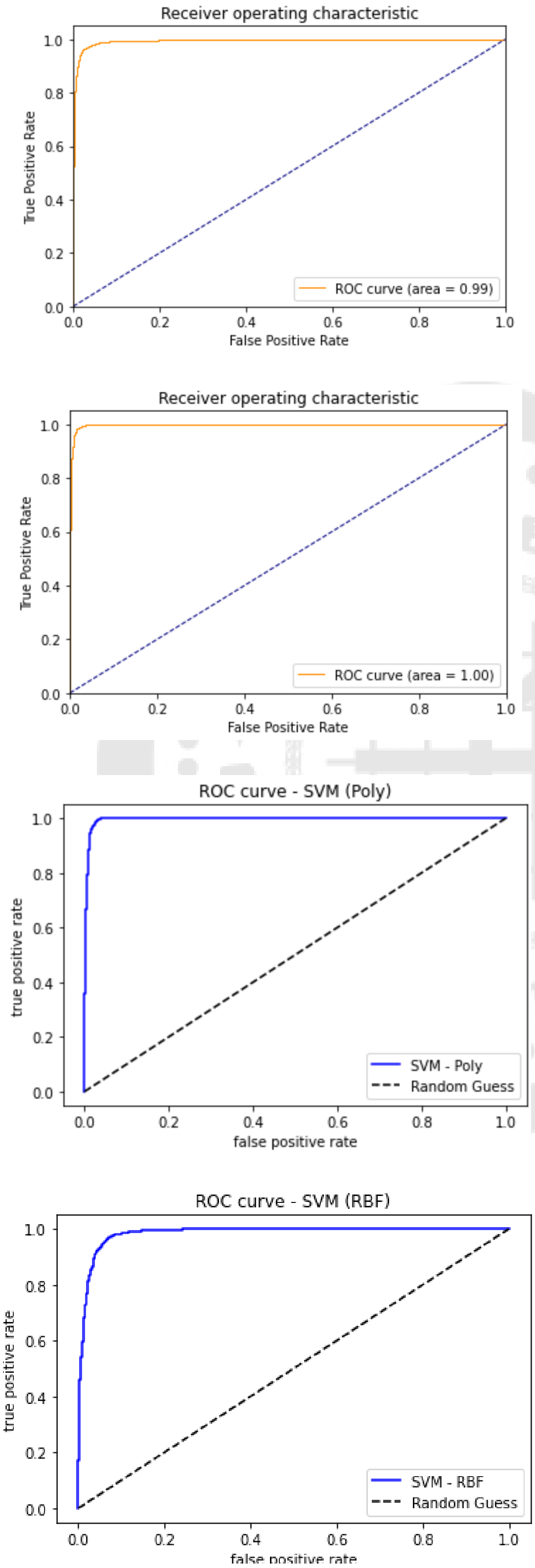
Classifier	Filter	Accuracy	Precision	Recall	F1	AUC
MLP	No Filter	0.87	0.87	0.85	0.86	0.94
	Blur4x4	0.84	0.84	0.85	0.84	0.92
	Blur8x8	0.80	0.78	0.84	0.81	0.89
	Sharpen	0.87	0.88	0.85	0.87	0.94
	Sharpen2	0.83	0.81	0.86	0.84	0.92
SVM-linear	No Filter	0.81	0.81	0.82	0.81	0.90
	Blur4x4	0.79	0.77	0.82	0.79	0.88
	Blur8x8	0.76	0.73	0.82	0.77	0.85
	Sharpen	0.83	0.85	0.82	0.83	0.92
	Sharpen2	0.81	0.80	0.82	0.81	0.89
SVM-poly	No Filter	0.87	0.87	0.86	0.87	0.94
	Blur4x4	0.86	0.85	0.86	0.86	0.93
	Blur8x8	0.83	0.81	0.86	0.84	0.91
	Sharpen	0.88	0.89	0.86	0.88	0.95
	Sharpen2	0.86	0.86	0.86	0.86	0.93
SVM-rbf	No Filter	0.50	1.00	0.00	0.00	0.50
	Blur4x4	0.50	1.00	0.00	0.00	0.50
	Blur8x8	0.50	1.00	0.00	0.00	0.50
	Sharpen	0.50	1.00	0.00	0.00	0.50
	Sharpen2	0.50	1.00	0.00	0.00	0.50
LR	No Filter	0.82	0.82	0.82	0.82	0.91
	Blur4x4	0.79	0.78	0.82	0.80	0.88
	Blur8x8	0.75	0.72	0.82	0.77	0.85
	Sharpen	0.84	0.85	0.82	0.84	0.92
	Sharpen2	0.81	0.80	0.82	0.81	0.89

Table 11 Performance of classifiers trained by the SeNet50 vectors

Classifier	Filter	Accuracy	Precision	Recall	F1	AUC
MLP	No Filter	0.83	0.83	0.82	0.83	0.91
	Blur4x4	0.80	0.79	0.81	0.80	0.89
	Blur8x8	0.76	0.74	0.82	0.77	0.85
	Sharpen	0.82	0.82	0.82	0.82	0.90
	Sharpen2	0.79	0.77	0.83	0.80	0.88
SVM-linear	No Filter	0.78	0.77	0.79	0.78	0.85
	Blur4x4	0.50	0.50	0.93	0.65	0.47
	Blur8x8	0.49	0.49	0.93	0.65	0.41
	Sharpen	0.49	0.50	0.93	0.65	0.49
	Sharpen2	0.49	0.50	0.93	0.65	0.47
SVM-poly	No Filter	0.84	0.84	0.83	0.84	0.92
	Blur4x4	0.50	0.50	1.00	0.67	0.48
	Blur8x8	0.50	0.50	1.00	0.67	0.34
	Sharpen	0.50	0.50	1.00	0.67	0.55
	Sharpen2	0.50	0.50	1.00	0.67	0.51
SVM-rbf	No Filter	0.50	0.00	0.00	0.00	0.50
	Blur4x4	0.50	0.00	0.00	0.00	0.50
	Blur8x8	0.50	0.00	0.00	0.00	0.50
	Sharpen	0.50	0.00	0.00	0.00	0.50
	Sharpen2	0.50	0.00	0.00	0.00	0.50
LR	No Filter	0.78	0.77	0.79	0.78	0.85
	Blur4x4	0.49	0.50	0.88	0.63	0.47
	Blur8x8	0.48	0.49	0.88	0.63	0.42
	Sharpen	0.49	0.50	0.88	0.63	0.49
	Sharpen2	0.48	0.49	0.88	0.63	0.47

Table 12 Performance of classifiers trained by the power spectrum features

Classifier	Filter	Accuracy	Precision	Recall	F1	AUC
MLP	No Filter	0.52	0.51	1.00	0.67	0.77
	Blur4x4	0.50	0.50	1.00	0.67	0.46
	Blur8x8	0.91	1.00	0.82	0.90	1.00
	Sharpen	0.96	0.99	0.93	0.96	0.99
	Sharpen2	0.98	1.00	0.95	0.98	1.00
SVM-linear	No Filter	0.86	0.91	0.80	0.85	0.93
	Blur4x4	0.70	0.66	0.80	0.73	0.79
	Blur8x8	0.90	1.00	0.80	0.89	1.00
	Sharpen	0.90	1.00	0.80	0.89	1.00
	Sharpen2	0.90	1.00	0.80	0.89	1.00
SVM-poly	No Filter	0.91	0.93	0.88	0.91	0.96
	Blur4x4	0.65	0.60	0.88	0.72	0.77
	Blur8x8	0.94	1.00	0.88	0.94	1.00
	Sharpen	0.92	0.95	0.88	0.92	0.98
	Sharpen2	0.72	0.66	0.88	0.76	0.74
SVM-rbf	No Filter	0.92	0.95	0.90	0.92	0.97
	Blur4x4	0.63	0.58	0.90	0.71	0.74
	Blur8x8	0.94	0.98	0.90	0.94	0.99
	Sharpen	0.93	0.97	0.90	0.93	0.99
	Sharpen2	0.86	0.84	0.90	0.87	0.92
LR	No Filter	0.85	0.87	0.81	0.84	0.91
	Blur4x4	0.67	0.63	0.81	0.71	0.77
	Blur8x8	0.90	1.00	0.81	0.89	0.99
	Sharpen	0.90	1.00	0.81	0.89	1.00
	Sharpen2	0.90	1.00	0.81	0.89	1.00



Feafure:	Power spectrum
Classsifier:	MLP
Test filter:	Sharpen
Accuracy:	0.96
AUC:	0.99

Feafure:	Power spectrum
Classsifier:	MLP
Test filter:	Sharpen 2
Accuracy:	0.98
AUC:	1.00

Feafure:	Power spectrum
Classsifier:	SVM-poly
Test filter:	Blur8x8
Accuracy:	0.94
AUC:	1.00

Feafure:	Power spectrum
Classsifier:	SVM-rbf
Test filter:	Blur8x8
Accuracy:	0.94
AUC:	0.99

Figure 10 ROC curves of the top performing models

CHAPTER 5
SUMMARY DISCUSSION AND SUGGESTION

5.1 Discussion

5.1.1 Similarity test

The similarity test results show that real photos are obviously closer to each other than real photos and generated images. The same direction of face pose (left, right and straight) cannot help making generated images get less cosine distance to real photos than the real photos with different direction of face pose. And since the average cosine distance between the right-pose pairs and the left pose pairs do not much deviate (0.267 / 0.264 for real-real pairs and 0.458 / 0.446 for real-generated pairs), it can be noted that the direction of face pose (right or left) does not significantly affect.

The cosine distance is generally used to represent similarity between a couple of images, especially human face images. The more cosine distance means the less similarity. This experiment result shows that image regeneration by the IDInvert model does not quite maintain similarity, since generated images obviously have more distances (less similarity) to real photos than the distances between real photos themselves. And as the generated images used in this experiment are not further modified by other techniques, it can be inferred that a generated image loses its similarity to its reference photo through regeneration process of the IDInvert model.

According to Figure 11, it can be seen that the generated image looks realistic. However, the cosine distance neither measure naturalness nor quality of generated images. But the similarity here rather focuses on personal identity of face images. The lower similarity scores suggest that the IDInvert model is not effectively preserving the personal identity of the reference individuals in the regenerated images. This could also be observed from the results that many generated images do not replicate all the specific attributes of reference faces, e.g., moles, eye shape and lip shape. This variation is also extracted when the images are encoded into the identity vector. As such, the face images with different levels of detail on these attributes would result in the identity vectors with the larger distance.

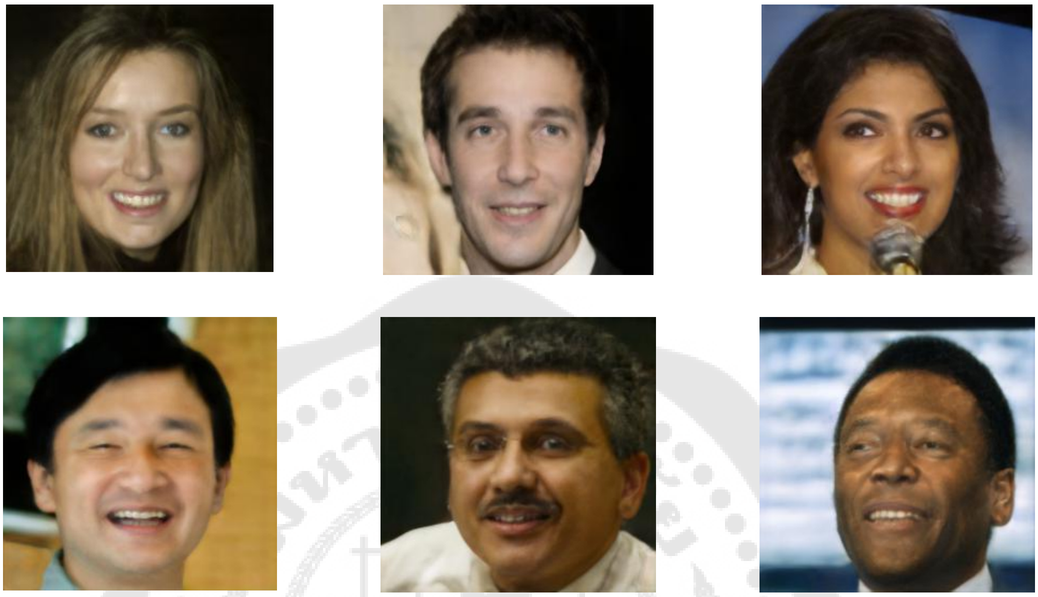


Figure 11 Samples of generated images

For the reason why the IDInvert model does not preserve personal identity of the reference face well, this study suggests that it might be because of its mechanism. The architecture of IDInvert model used in this study (J. Zhu, Shen, Zhao, & Zhou, 2020a), which is a ready-to-use edition, mainly consists of an encoder network and the pretrained StyleGAN model and is trained using the FFHQ dataset of human face images. The embedded StyleGAN, which is used to reconstruct an image, probably prioritizes the naturalness of generated images rather than preserving personal identity. Consequently, some personal attribute that quite deviates from the StyleGAN discriminator's view of 'realistic face', it might be excluded. The model seems to limit the output of generated faces within range of average faces that it has experienced and omits some personal attributes that are considered the outliers.

5.1.2 Detectability test

Regarding the detectability test, the results indicate that the face images the IDInvert generated images can be distinguished well by using existing feature extraction techniques like CNNs, frequency domain analysis technique coupled with standard ML classifiers.

In this experiment, using Discrete Fourier Transform algorithm and Azimuthal Average (Durall et al., 2019) to extract images' power spectrum as features for the SVM models with polynomial and rbf kernels present the best performance against generated images without filter. However, the performance of each classifier model is not consistent when facing blur and sharpen filters.

The power spectrum is used to detect differences in high frequency signal levels between real and generated face images, as demonstrated in Figure 12 Thus, modifying the frequency components of generated images with filters can significantly impact the model's detection performance. Our experimental results indicate that the SVM-rbf model and the SVM-poly model perform slightly better against generated images modified with the blur8x8 filter and the both sharpen filters (except the SVM-poly that also performed worse for the filter sharpen2) but performs much worse against images modified with the blur4x4 filter. This suggests that the power spectrum features are not robust to these types of image modification operators, and that these techniques can be applied to lower detection changes.

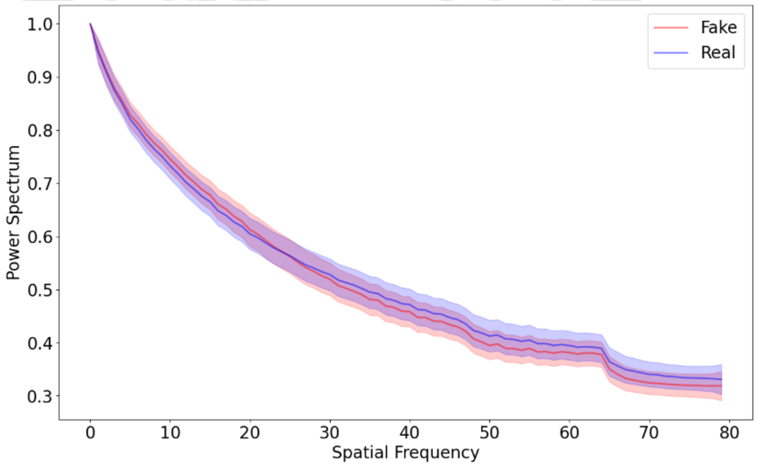


Figure 12 Frequency components of fake images and real photos

In addition, we also found that by using features extracted by ResNet and SeNet, the model can detect generated images with lower detection performance.

However, the models with ResNet50 vectors is more robust against image modification operators when using these features.

Regarding the models with ResNet50 vectors, the MLP and the SVM with polynomial kernel perform the best against unfiltered images. The performance of each classifier does not dramatically increase or decrease when tested on images that are filtered by the two levels of blurs and the two levels of sharpen. One exception is the SVM with rbf kernel which performs poorly for all cases. Compared to power spectrum, ResNet50 vectors perform slightly poorer but more consistently against image filtering.

Regarding the models with SeNet50 vectors, the classification performance of all the models is the lowest one among three types of vectors except for the MLP model. In addition, when tested on images that are filtered, the MLP's performance does not much change, but the performance of SVM with polynomial kernel, SVM with linear kernel and Logistic Regression significantly drops. One exception is again the SVM with rbf kernel which performs poorly for all cases.

Overall, our findings highlight the importance of carefully considering the impact of image modification techniques on detection performance and suggest that using alternative feature extraction methods can lead to improved robustness against image modification.

The objective of this study was to compare the performance of detection models that use power spectrum features with those that use ResNet and SeNet to extract features and investigate the impact of image filters on the detection of generated images. The results revealed that applying filters to generated images in the test set significantly affected the performance of detection models that used power spectrum features. To gain a deeper understanding of how image filters affect the frequency signal of generated images, future research should conduct an extended experiment using a wider range of image modification techniques.

On the other hand, it was found that the models that utilized ResNet and SeNet for feature extraction were more robust against these image modification operators. These findings suggest that the choice of feature extraction method can significantly affect the robustness of detection models to image filters. Overall, the study highlights the need to carefully consider the features used in detection models for generated images and to evaluate their performance under various image modification scenarios.

It should be noted that these feature extraction techniques are not initially designed for face image classification works, including the fake face detection here. The VGGFace module with ResNet50 and SeNet50 is generally used for biometric recognition task and maybe more suitable for identifying each individual person than classifying groups of face images collectively. And although frequency domain analysis techniques were found to perform very well for detecting fake contents, they were previously initiated for other fields, especially electronics and control system engineering. For the more reliable and consistent fake face detection performance, the suggested solution is to combine several techniques and models.

5.2 Conclusion

The IDInvert model is one of the most recent GAN inversion models and can regenerate face images of real people and easily deceive human' eyes. However, the experimental results of this study could be inferred that its generated images do not quite preserve the personal identity of reference persons, as the cosine similarity between those generated images and real photos is significantly different. The generated faces can be observed that they look different from the reference face as if they are not representing the same persons. Due to this reason, the risk that these fake images would imitate real people faces and cause serious damage is not well-awared currently.

In addition, the generated images can be distinguished from real photos by using the existing feature extraction techniques and classification models. Some of these models can achieve over 90% accuracy. Although these generated images may deceive humans' eyes, they have got hidden features which are different from the features of real photos. These differences could be captured by existing techniques and used to distinguish generated images from real photos.

Although, those fake images generated by the IDInvert model are distinguishable, all the applied techniques here require some related skills and computational resources that would not always be timely and widely accessible. In daily life, we may encounter these fake contents without tools for tackling them. On the other hand, harmful attacks using fake contents, or any technology misuses, might occur any chance, and cause serious losses in some context. Consciousness and prudence are strongly recommended as the first protection.

5.3 Future works

The IDInvert model and other recent GAN inversion models might be considered as the beginning phase since they were just invented during the last few years. Their ability to imitate real faces therefore is not yet perfect and hazardous currently. However, it should be aware that the GAN inversion and other generative models have been very actively progressed. With this regard, the proposed techniques of this study may be applied to further examine the potential threats of newly developed generative models.

Future works related to this study therefore should be to keep examining novel generative models' abilities to imitate and manipulate real contents and identify their possible harmful abuses. Regarding the similarity test, the methodology may be redesigned to enumerate how different GAN inversion models would perform on maintaining similarity to the reference photos and whether it is possible to keep both the realisticness and the similarity. And for the detectability test, the extended experiment using the wider range of image filters to examine why and how different filters differently affect the dectection performance would be interesting.

REFERENCES

Abdal, R., Qin, Y., & Wonka, P. (2019). *Image2stylegan: How to embed images into the stylegan latent space?* Paper presented at the Proceedings of the IEEE/CVF International Conference on Computer Vision.

Abdal, R., Qin, Y., & Wonka, P. (2020). *Image2stylegan++: How to edit the embedded images?* Paper presented at the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.

Alaluf, Y., Patashnik, O., & Cohen-Or, D. (2021). *Restyle: A residual-based stylegan encoder via iterative refinement.* Paper presented at the Proceedings of the IEEE/CVF International Conference on Computer Vision.

Alaluf, Y., Tov, O., Mokady, R., Gal, R., & Bermano, A. (2022). *Hyperstyle: Stylegan inversion with hypernetworks for real image editing.* Paper presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

Anirudh, R., Thiagarajan, J. J., Kailkhura, B., & Bremer, P.-T. (2020). Mimicgan: Robust projection onto image manifolds with corruption mimicking. *International Journal of Computer Vision, 128*(10), 2459-2477.

Arjovsky, M., & Bottou, L. (2017). Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*.

Arjovsky, M., Chintala, S., & Bottou, L. (2017). *Wasserstein generative adversarial networks.* Paper presented at the International conference on machine learning.

Bau, D., Strobelt, H., Peebles, W., Wulff, J., Zhou, B., Zhu, J.-Y., & Torralba, A. (2020). Semantic photo manipulation with a generative image prior. *arXiv preprint arXiv:2005.07727*.

Bau, D., Zhu, J.-Y., Wulff, J., Peebles, W., Strobelt, H., Zhou, B., & Torralba, A. (2019). *Seeing what a gan cannot generate*. Paper presented at the Proceedings of the IEEE/CVF International Conference on Computer Vision.

Bellemare, M. G., Danihelka, I., Dabney, W., Mohamed, S., Lakshminarayanan, B., Hoyer, S., & Munos, R. (2017). The cramer distance as a solution to biased wasserstein gradients. *arXiv preprint arXiv:1705.10743*.

Berthelot, D., Schumm, T., & Metz, L. (2017). Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*.

Brock, A., Donahue, J., & Simonyan, K. (2018). Large scale GAN training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*.

Cao, Q., Shen, L., Xie, W., Parkhi, O. M., & Zisserman, A. (2018). *Vggface2: A dataset for recognising faces across pose and age.* Paper presented at the 2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018).

Chai, L., Zhu, J.-Y., Shechtman, E., Isola, P., & Zhang, R. (2021). *Ensembling with deep generative views.* Paper presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

Cheng, J., Dong, L., & Lapata, M. (2016). Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*.

Choi, Y., Uh, Y., Yoo, J., & Ha, J.-W. (2020). *Stargan v2: Diverse image synthesis for multiple domains.* Paper presented at the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.

Collins, E., Bala, R., Price, B., & Susstrunk, S. (2020). *Editing in style: Uncovering the local semantics of gans.* Paper presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

Dinh, T. M., Tran, A. T., Nguyen, R., & Hua, B.-S. (2022). *Hyperinverter: Improving stylegan inversion via hypernetwork.* Paper presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

Durall, R., Keuper, M., Pfreundt, F.-J., & Keuper, J. (2019). Unmasking deepfakes with simple features. *arXiv preprint arXiv:1911.00686*.

Gonzalez, R. C. (2009). *Digital image processing*: Pearson education india.

Goodfellow, I. (2016). Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., . . .

Bengio, Y. (2014). Generative adversarial networks. *Communications of the ACM, 63*(11), 139-144.

Gu, J., Shen, Y., & Zhou, B. (2020). *Image processing using multi-code gan prior.* Paper presented at the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.

Guarnera, L., Giudice, O., & Battiato, S. (2020). *Deepfake detection by analyzing convolutional traces.* Paper presented at the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. C. (2017). Improved training of wasserstein gans. *Advances in neural information processing systems, 30*.

Guo, Y., Chen, Q., Chen, J., Wu, Q., Shi, Q., & Tan, M. (2019). Auto-embedding generative adversarial networks for high resolution image synthesis. *IEEE Transactions on Multimedia, 21*(11), 2726-2737.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Deep residual learning for image recognition.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems, 30*.

Hore, A., & Ziou, D. (2010). *Image quality metrics: PSNR vs. SSIM.* Paper presented at the 2010 20th international conference on pattern recognition.

Hu, J., Shen, L., & Sun, G. (2018). *Squeeze-and-excitation networks.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

Huang, G. B., & Learned-Miller, E. (2014). Labeled faces in the wild: Updates and new reporting procedures. *Dept. Comput. Sci., Univ. Massachusetts Amherst, Amherst, MA, USA, Tech. Rep, 14*(003).

Huang, G. B., Mattar, M., Berg, T., & Learned-Miller, E. (2008). *Labeled faces in the wild:*

A database forstudying face recognition in unconstrained environments. Paper presented at the Workshop on faces in'Real-Life'Images: detection, alignment, and recognition.

Huang, X., & Belongie, S. (2017). *Arbitrary style transfer in real-time with adaptive instance normalization.* Paper presented at the Proceedings of the IEEE international conference on computer vision.

Kang, K., Kim, S., & Cho, S. (2021). *Gan inversion for out-of-range images with geometric transformations.* Paper presented at the Proceedings of the IEEE/CVF International Conference on Computer Vision.

Karnewar, A., & Wang, O. (2020). *Msg-gan: Multi-scale gradients for generative adversarial networks.* Paper presented at the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.

Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2017). Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.

Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., & Aila, T. (2020). Training generative adversarial networks with limited data. *Advances in neural information processing systems, 33*, 12104-12114.

Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., & Aila, T. (2021). Alias-free generative adversarial networks. *Advances in neural information processing systems, 34*, 852-863.

Karras, T., Laine, S., & Aila, T. (2019). *A style-based generator architecture for generative adversarial networks.* Paper presented at the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., & Aila, T. (2020). *Analyzing and improving the image quality of stylegan.* Paper presented at the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.

LeCun, Y., & Huang, F. J. (2005). *Loss functions for discriminative training of energy-based models.* Paper presented at the International workshop on artificial intelligence and statistics.

Li, C.-L., Chang, W.-C., Cheng, Y., Yang, Y., & Póczos, B. (2017). Mmd gan: Towards deeper understanding of moment matching network. *Advances in neural information processing systems, 30*.

Liu, Z., Luo, P., Wang, X., & Tang, X. (2015). *Deep learning face attributes in the wild.* Paper presented at the Proceedings of the IEEE international conference on computer vision.

Liu, Z., Qi, X., & Torr, P. H. (2020). *Global texture enhancement for fake face detection in the wild.* Paper presented at the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.

Lukac, R., & Plataniotis, K. N. (2018). *Color image processing: methods and applications*: CRC press.

Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., & Paul Smolley, S. (2017). *Least squares generative adversarial networks.* Paper presented at the Proceedings of the IEEE international conference on computer vision.

Marra, F., Gragnaniello, D., Verdoliva, L., & Poggi, G. (2019). *Do gans leave artificial fingerprints?* Paper presented at the 2019 IEEE conference on multimedia information processing and retrieval (MIPR).

Marra, F., Saltori, C., Boato, G., & Verdoliva, L. (2019). *Incremental learning for the detection and classification of gan-generated images.* Paper presented at the 2019 IEEE international workshop on information forensics and security (WIFS).

McCloskey, S., & Albright, M. (2019). *Detecting GAN-generated imagery using saturation cues.* Paper presented at the 2019 IEEE international conference on image processing (ICIP).

Metz, L., Poole, B., Pfau, D., & Sohl-Dickstein, J. (2016). Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*.

Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Miyato, T., Kataoka, T., Koyama, M., & Yoshida, Y. (2018). Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*.

Neves, J. C., Tolosana, R., Vera-Rodriguez, R., Lopes, V., Proença, H., & Fierrez, J. (2020). Ganprintr: Improved fakes and evaluation of the state of the art in face manipulation detection. *IEEE Journal of Selected Topics in Signal Processing, 14*(5), 1038-1048.

Odena, A., Olah, C., & Shlens, J. (2017). *Conditional image synthesis with auxiliary classifier gans.* Paper presented at the International conference on machine learning.

Parikh, A. P., Täckström, O., Das, D., & Uszkoreit, J. (2016). A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933*.

Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015). Deep face recognition.

Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Richardson, E., Alaluf, Y., Patashnik, O., Nitzan, Y., Azar, Y., Shapiro, S., & Cohen-Or, D. (2021). *Encoding in style: a stylegan encoder for image-to-image translation.* Paper presented at the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.

Roich, D., Mokady, R., Bermano, A. H., & Cohen-Or, D. (2022). Pivotal tuning for latent-based editing of real images. *ACM Transactions on Graphics (TOG), 42*(1), 1-13.

Ronneberger, O., Fischer, P., & Brox, T. (2015). *U-net: Convolutional networks for biomedical image segmentation.* Paper presented at the International Conference on Medical image computing and computer-assisted intervention.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., . . . Bernstein, M. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision, 115*(3), 211-252.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training gans. *Advances in neural information processing systems, 29*.

Salimans, T., & Kingma, D. P. (2016). Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *Advances in neural information processing systems, 29*.

Shen, Y., Gu, J., Tang, X., & Zhou, B. (2020). *Interpreting the latent space of gans for semantic face editing*. Paper presented at the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.

Singhal, A. (2001). Modern information retrieval: A brief overview. *IEEE Data Eng. Bull., 24*(4), 35-43.

Tolosana, R., Vera-Rodriguez, R., Fierrez, J., Morales, A., & Ortega-Garcia, J. (2020). Deepfakes and beyond: A survey of face manipulation and fake detection. *Information Fusion, 64*, 131-148.

Tov, O., Alaluf, Y., Nitzan, Y., Patashnik, O., & Cohen-Or, D. (2021). Designing an encoder for stylegan image manipulation. *ACM Transactions on Graphics (TOG), 40*(4), 1-14.

Viazovetskyi, Y., Ivashkin, V., & Kashin, E. (2020). *Stylegan2 distillation for feed-forward image manipulation.* Paper presented at the European conference on computer vision.

Wang, L., Guo, S., Huang, W., Xiong, Y., & Qiao, Y. (2017). Knowledge guided disambiguation for large-scale scene classification with multi-resolution CNNs. *IEEE Transactions on Image Processing, 26*(4), 2055-2068.

Wang, R., Juefei-Xu, F., Ma, L., Xie, X., Huang, Y., Wang, J., & Liu, Y. (2019). Fakespotter: A simple yet robust baseline for spotting ai-synthesized fake faces. *arXiv preprint arXiv:1909.06122*.

Wang, T., Zhang, Y., Fan, Y., Wang, J., & Chen, Q. (2022). *High-fidelity gan inversion for image attribute editing.* Paper presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

Warzel, C. (2019). Faceapp shows we care about privacy but don’t understand it. *New York Times*.

Wei, T., Chen, D., Zhou, W., Liao, J., Zhang, W., Yuan, L., . . . Yu, N. (2022). E2Style: Improve the efficiency and effectiveness of StyleGAN inversion. *IEEE Transactions* *on Image Processing, 31*, 3267-3280.

Wu, Z., Lischinski, D., & Shechtman, E. (2021). *Stylespace analysis: Disentangled controls for stylegan image generation.* Paper presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

Xia, W., Zhang, Y., Yang, Y., Xue, J.-H., Zhou, B., & Yang, M.-H. (2022). Gan inversion: A survey. *IEEE transactions on pattern analysis and machine intelligence*.

Yu, N., Davis, L. S., & Fritz, M. (2019). *Attributing fake images to gans: Learning and analyzing gan fingerprints.* Paper presented at the Proceedings of the IEEE/CVF international conference on computer vision.

Zhang, H., Goodfellow, I., Metaxas, D., & Odena, A. (2019). *Self-attention generative adversarial networks.* Paper presented at the International conference on machine learning.

Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE signal processing letters, 23*(10), 1499-1503.

Zhang, X., Karaman, S., & Chang, S.-F. (2019). *Detecting and simulating artifacts in gan fake images.* Paper presented at the 2019 IEEE international workshop on information forensics and security (WIFS).

Zhao, J., Mathieu, M., & LeCun, Y. (2016). Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*.

Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). *Unpaired image-to-image translation using cycle-consistent adversarial networks.* Paper presented at the Proceedings of the IEEE international conference on computer vision.

Zhu, J., Shen, Y., Zhao, D., & Zhou, B. (2020a). *In-domain gan inversion for real image editing.* Paper presented at the European conference on computer vision.

Zhu, J., Shen, Y., Zhao, D., & Zhou, B. (2020b). In-Domain GAN Inversion for Real Image Editing Supplementary Material.

Zhu, P., Abdal, R., Qin, Y., Femiani, J., & Wonka, P. (2020). Improved stylegan embedding: Where are the good latents? *arXiv preprint arXiv:2012.09036*.

VITA