Personal Photo Privacy Protection by Attacking Face Detectors

by

Zhang Chongyang

Tokyo University of Technology

Graduate School of Bionics, Computer and Media Sciences

January, 2023

2023

Abstract

As artificial intelligence becomes increasingly integrated into people's daily lives, researchers are increasingly concerned with how to use these technologies safely and responsibly. Malicious face swapping is caused by the abuse of AI technology and seriously endangers individuals' privacy security. Based on previous research and the workflow of the popular face-swapping program, Faceswap, this study argues that attacking the facial detection stage is an effective solution to this issue. However, there is currently no attack method that can simultaneously make MTCNN, SSD, and S3FD provided by Faceswap ineffective. This study presents a solution, with the specific work content summarized as follows.

In Part One, based on previous research methods, the image style transfer technique is employed to enhance the probability of detecting the background portion in an image, and the change of MTCNN facial detection result is observed. It is confirmed that the addition of perturbations to the background under these conditions does not impact the MTCNN facial detection result. The second part, a method of attacking the MTCNN facial detector with invisible perturbations is proposed for the first time, and the reasons for its difficulty in being attacked are explained. Based on the feature of using image pyramids to process input images, an attack method that fuses perturbations of multiple scales is constructed. This is a white-box attack method, with a success rate of 76.3% in quantitative testing on CelebA data. Also, for the first time in the research of attacking MTCNN with invisible perturbations, a baseline for image quality assessment based on CelebA data is proposed. The baseline scores for PSNR, SSIM, and LPIPS

image quality assessment algorithms are 30.25, 0.9, and 0.06, respectively.

In the third part, a method of disrupting feature extraction continuity by adding black lines to the face is proposed. The attack ability is enhanced as the width of the line increases. Quantitative experiments were conducted using MTCNN, SSD, and S3FD, when the line width was 10 pixels, the detection rates in CelebA data dropped to 6.15%, 9.47%, and 32.1%, respectively; when the line width was 8 pixels, the detection rates in FFHQ data dropped to 7.2%, 4.3%, and 9%, respectively. Due to the reduced usability caused by the coverage of facial features by black lines, this study conducted optimization experiments on the structure and image quality of this method. According to the results of the structure optimization experiment, it is difficult to optimize the black line structure manually. Short line structures can expose the eyes and corners of the mouth, which can be optimized to some extent, but the usability is still affected. In the image quality optimization experiment, random perturbations were added to the coordinates between the two points of the line, and the coverage range of the perturbations was reduced to below 45%. This experiment takes images where the face is completely covered by black lines as baseline images for image quality evaluation. Based on the PSNR, SSIM, and LPIPS image quality assessment methods, the scores on the CelebA data are 9.27, 0.18, and 0.21 higher than the baseline of this experiment, and the scores on the FFHQ data are 8.56, 0.3, and 0.33 higher than the baseline. Combining the questionnaire survey on user usage requirements, the generated adversarial examples using this method have a 34.2% higher acceptance rate compared to the baseline images. Based on the results of the aforementioned studies, it can be concluded that our method is capable of providing assistance to users who have the need for facial privacy security.

Contents

Abstract i

List of Figures vi

List of Tables x

1 Introduction 1

1.1 Research Background . . . 1

1.1.1 Face Detection . . . 3

1.1.2 Generative Models . . . 6

1.1.3 Adversarial Attack . . . 9

1.2 Research Objectives . . . 11

1.3 The Structure of This Thesis . . . 13

2 Related Work 15

2.1 Face Detection . . . 15

2.1.1 Convolutional Neural Network . . . 15

2.1.2 Multi-Task Convolutional Neural Network . . . 18

2.1.2.1 Image Pyramid . . . 18

2.1.2.2 Network Structure . . . 18

2.1.3 Single Shot MultiBox Detector . . . 22

2.1.4 Single Shot Scale-invariant Face Detector . . . 23
2.2 Adversarial Attack . . . 25
2.2.1 Fast Gradient Sign Method . . . 25
2.2.1.1 Pipeline . . . 25
2.2.1.2 Linear Validity Analysis . . . 26
2.2.2 Projected Gradient Descent . . . 27

3 Correlation between Background and Face 28
3.1 Introduction . . . 28
3.2 Neural style transfer . . . 30
3.3 Method and Experiment . . . 31
3.4 Discussion and Conclusions . . . 33

4 Multi-scale Perturbation Fusion Adversarial Attack of MTCNN Face Detection System 34
4.1 Introduction . . . 34
4.2 Image Interpolation . . . 36
4.3 Method . . . 38
4.4 Experiment . . . 41
4.4.1 Effectiveness Experiment . . . 41
4.4.2 Structural Similarity . . . 43
4.4.3 Effective Scales . . . 44
4.5 Discussion . . . 45
4.6 Conclusions . . . 48

5 A New Method of Disabling Face Detection by Drawing Lines between Eyes and Mouth 49

5.1 Introduction . . . 49
5.2 Black Line Structure Experiment . . . 52
5.2.1 Dataset . . . 52
5.2.2 Method . . . 52
5.2.3 Result . . . 54
5.2.4 Discussion . . . 56
5.3 Structure Optimization Experiment . . . 61
5.3.1 Method . . . 61
5.3.2 Result . . . 62
5.3.3 Discussion . . . 73
5.4 Image Quality Optimization Experiment . . . 74
5.4.1 Method and Experiment . . . 75
5.4.2 Discussion . . . 78
5.5 Conclusions . . . 80

6 Conclusions **82**
6.1 Chapters Summary . . . 83
6.2 Future Work . . . 85

Acknowledgments **86**

References **88**

List of Publications Related to This Thesis **95**

List of Publications **96**

List of Figures

1.1 CNN structures of the 12-net, 24-net and 48-net 5

1.2 The structure of an autoencoder. 8

1.3 The Faceswap program performs face swap training and converting pipelines. . . 8

1.4 An adversarial input overlaid on an original image can cause a classifier to misclassify a panda as a gibbon. 10

2.1 The network structure of LeNet-5. 16

2.2 The blue curve is the shape of the function itself, and the yellow curve is the shape of the function after taking the derivative. 17

2.3 Image pyramids deal with the structure of an image. 18

2.4 P-Net network structure pipeline. 19

2.5 R-Net network structure pipeline. 20

2.6 O-Net network structure pipeline. 21

2.7 SSD network structure pipeline. 22

2.8 S3FD network structure pipeline. 24

3.1 Provide non-facial training data for AI facial synthesis models. 29

3.2 The neural style transfer process. 30

3.3 MTCNN's P-Net network for rough detection and extraction of the bounding box content. 31

3.4 The images of the face region and the images of the three non-face regions are style transferred and restored to their original positions. 32
3.5 Before and after the style transfer, the facial detection results are compared. . . . 32

4.1 In the digital domain, attacks against whole images and faces can disable SSD-based face detectors. 35
4.2 Training data and adversarial samples for attacking MTCNN in the physical domain. 35
4.3 In bilinear interpolation, the associated four-pixel values are calculated. 36
4.4 From left to right, the original sizes of the images are 128x128, 178x218, 300x232, and 3840x2160. 42
4.5 The original size of the two images is 1300x867, 1024x820. 43
4.6 Taken from an image of size 128x128, downscaled to 0.1. 43
4.7 Effective size comparison image. 45
4.8 The original pixel image and the interpolated image. 47

5.1 Blocking part of the facial area and performing face detection through face detectors SSD, S3FD, and MTCNN. 50
5.2 Image (a) is a manually added black line. Image (b) is a presentation image of the black line structure. Both images are from the FFHQ dataset. 53
5.3 Different unit pixel widths are used due to the other face sizes in the CelebA dataset and the FFHQ dataset. The table shows the probability of the black line image being detected by MTCNN, S3FD, and SSD. 55
5.4 The image above shows some images with black line structure added but which can be detected by the face detector. 56
5.5 We selected two images from the CelebA dataset and added a black line structure with widths of 6 pixels, 8 pixels, and 10 pixels. MTCNN cannot detect the first three pictures, and the last three pictures are pictures that MTCNN can detect. . . 58

5.6 We selected two images from the FFHQ dataset and added a black line structure with widths of 4 pixels, 6 pixels, and 8 pixels. mtcnn cannot detect the first three pictures, and the last three pictures are pictures that mtcnn can detect. 58
5.7 We selected two images from the CelebA dataset and added a black line structure with widths of 6 pixels, 8 pixels, and 10 pixels. S3FD cannot detect the first three pictures, and the last three pictures are pictures that S3FD can detect. 59
5.8 We selected two images from the FFHQ dataset and added a black line structure with widths of 4 pixels, 6 pixels, and 8 pixels. S3FD cannot detect the first three pictures, and the last three pictures are pictures that S3FD can detect. 59
5.9 We selected two images from the CelebA dataset and added a black line structure with widths of 6 pixels, 8 pixels, and 10 pixels. SSD cannot detect the first three pictures, and the last three pictures are pictures that SSD can detect. 60
5.10 We selected two images from the FFHQ dataset and added a black line structure with widths of 4 pixels, 6 pixels, and 8 pixels. SSD cannot detect the first three pictures, and the last three pictures are pictures that SSD can detect. 60
5.11 An example diagram of an optimized structure. 61
5.12 The figure shows the resulting curve that the MTCNN face detector can detect the face. 65
5.13 The figure shows the resulting curve that the SSD face detector can detect the face. 66
5.14 The figure shows the resulting curve that the S3FD face detector can detect the face. 67
5.15 The figure shows the detection results of MTCNN, SSD, and S3FD face detectors at various widths of line segments. 68
5.16 Comparison of the detection results of the short line structure and the complete black line structure detected by MTCNN. 70

5.17 Comparison of the detection results of the short line structure and the complete black line structure detected by SSD. 71
5.18 Comparison of the detection results of the short line structure and the complete black line structure detected by S3FD. 72
5.19 The (a) represents the images with the faces fully covered and the (b) represents the images with added random perturbation. 76
5.20 The images are from the FFHQ dataset, with the first row being randomly perturbed images and the second row being genetically perturbed images. 78
5.21 A comparative chart of different professions' acceptance levels of different adversarial samples was included in the user privacy requirements survey. The 'yes' represents the number of participants who chose the adversarial sample, while 'no' represents the number of participants who did not choose it, both displayed as a percentage. 80

List of Tables

4.1 Use the original image size as the data for the structural similarity comparison results. . . . 44

4.2 The "Original" row shows the evaluation score of image quality when the image has not been modified. The "CelebA" row shows the adversarial sample image quality scores in the CelebA dataset. . . . 44

4.3 Before and after interpolation, the value changes of the adversarial samples are compared. . . . 46

5.1 The table shows the detection capabilities of the three face detectors on the original images in the CelebA dataset and on images with black line structures of 6-pixel, 8-pixel, and 10-pixel widths added. . . . 54

5.2 The table shows the detection capabilities of the three face detectors on the original images in the FFHQ dataset and on images with black line structures of 4-pixel, 6-pixel, and 8-pixel widths added. . . . 54

5.3 The values in the table are the detection results that the MTCNN face detector can detect the face. . . . 63

5.4 The values in the table are the detection results that the SSD face detector can detect the face. . . . 64

5.5 The values in the table are the detection results that the S3FD face detector can detect the face. 64
5.6 The values in the table are the detection results that MTCNN, SSD, and S3FD can detect faces when the line segment width is 2 pixels. 68
5.7 The table below shows the detection capabilities of three face detectors for short line structures. 69
5.8 The "Original" row displays the image quality evaluation score for the unaltered image. "Baseline" show the image quality benchmarks for the CelebA and FFHQ datasets when the faces are fully obscured by black. Each column represents the score obtained from the relevant evaluation method. The "Black line structure" row, "Random perturbation" row, and "Genetic Algorithm" row respectively show their image quality scores in the corresponding dataset. 75

1 Introduction

1.1 Research Background

Artificial intelligence has existed for more than 60 years since its proposal in 1956. Researchers have overcome one challenge after another to bring artificial intelligence into everyday life. At present, artificial intelligence research mainly focuses on three aspects: natural language processing, computer vision, and robotics. However, with the advancement of technology comes new challenges. Despite the convenience it brings, the risk of malicious use must be addressed, including the need for prevention and detection methods. Computer vision technology, due to its widespread use in daily life, is particularly vulnerable to malicious exploitation. Among these challenges, the negative social impact of the malicious use of face-swapping technology is particularly noticeable.

In 2017, a user known as "DeepFakes" posted a manipulated video on a social media platform. Using face-swapping technology, the user replaced the face of an actor in the video with that of a pornographic film star, giving the impression that the video was authentic [45]. Subsequently, fake news and manipulated speeches featuring politicians became prevalent on online platforms, and even the faces of ordinary individuals were maliciously used in such videos. This malicious use of face-swapping technology not only harms the reputation of companies and famous individuals, but also leads to the dissemination of incorrect information to the public. It will also pose a great threat to the life, work and even mental health of ordinary people. For example, malicious

users may use face-swapping technology to superimpose the face of a victim onto the body of a pornographic actress, thereby insulting, slandering, and even blackmailing the victim. Additionally, the technology may be used to spread political propaganda. This face-swapping technology is also known as "DeepFakes".

With the emergence of fake photos or fake videos, detection algorithms for fake content have also been proposed. For example, LRNet [48] detects fakes videos by temporal modeling of geometric features; A multi-attention deepfakes detection algorithm that uses multi-spatial attention heads to prompt the network to notice different local parts and enhances texture feature blocks to amplify subtle artifacts in shallow features [58]; Some researchers even constructed a new large-scale fake face dataset "FFIW-10K" in order to improve face forgery detection to a new level. FFIW-10K includes 10,000 high-quality fake videos with an average of three faces per frame [59]. However, even if the fake photos are detected, the social harm it caused and the psychological damage to the victims is irreversible. Therefore, research on preventing malicious face swapping should receive more attention from researchers. Leaving the decision on whether to allow face-swapping in the hands of the photo owner can bring long-term stability and harmony to society.

At present, many face swap programs incorporate deepfakes, such Faceswap [8], FakeApp [9], OpenFaceSwap [37], DeepFaceLab [38], etc. Most face-swapping programs are built on the Faceswap platform, including Faceswap, OpenFaceSwap, FakeApp, and DeepFaceLab. Faceswap is known for its high efficiency, reliability, and ease of debugging. It offers a variety of parameters for the face-swapping process, which can be divided into three main steps.

Extraction. Faceswap provides three face detectors, the face detector in the OpenCV [36] module based on the Single Shot multibox Detector (SSD) [29], the MultiTask cascaded Convolutional Neural Network (MTCNN) [55], and the Single Shot Scale-invariant Face Detector (S3FD) [57]. Optional parameters are face alignment and mask generation. Finally, it is also possible to manually filter the detected images to remove the images detected incorrectly or not

needed.

Traning. Although some face swapping algorithms use the generative adversarial networks (GANs) [13], the training model provided in the Faceswap is based on the architecture of autoencoders [22], which is also a type of generative network.

Convert. After the optional steps of personalization, such as color adjustments and the selection of mask types, the face-swapping process is complete.

From the process of Faceswap, it is clear that if the face detection step or the training step does not function correctly, the face swapping will fail. In this way, malicious face-swapping can be prevented. Before discussing the specific objectives of this study, let me provide information about face detection, generative models, and adversarial attacks.

1.1.1 Face Detection

Face detection has evolved through three stages: early methods, AdaBoost [10] framework, and deep learning.

In the early stages of face detection development, algorithms used template matching to match a face template with each position in the image to identify if a face is present. This approach involves performing binary classification on whether a certain area of an image contains a face or not. A representative study is a method proposed by Rowley et al. [40, 41]. They used a 20x20 image size to train a multi-layer perceptron model for face detection. The method outlined in [40] addresses the task of approximate front face detection. Meanwhile, the approach described in [41] addresses the challenge of multi-angle facial detection. The detection system consists of two neural networks, where the first network is used to estimate the perspective of a human face, while the second network determines whether the image depicts a face. The term "multi-angle" in this context refers to the presence of rotations in the image, rather than rotations of the face in a three-dimensional space. The method proposed by Rowley et al. first employs the angle estimator to output a rotation angle, which rotates the detection window, effectively straightening

the image. Then, the second network is used to judge whether the straightened image is a face. While the method has high accuracy, it has the drawback of being slow in detecting faces due to the need for comparing features with many benchmark faces and using a dense sliding window for sampling and classification.

The Boost [44] algorithm is a collection of ensemble learning algorithms that are founded on the theory of Probably Approximately Correct (PAC) [15] learning. The fundamental idea behind Boost is to construct a highly accurate robust classifier through the combination of multiple simple weak classifiers. This approach formed the basis for the later AdaBoost framework. In 2001, Viola and Jones developed a face detection algorithm [51]. They jointly developed a face detector that utilized the Haar-like feature extraction technique and cascaded AdaBoost classifiers. The improved detection speed while retaining high accuracy is referred to as the VJ framework. The VJ framework is a significant milestone in the history of face detection and serves as the foundation for the AdaBoost-based target detection framework. Before the rise of deep learning, the VJ framework was the basis for face detection solutions in the industry. However, the VJ algorithm still had some limitations. The features extracted by Haar-like are simple and lack stability. The weak classifiers, which use basic decision trees, are susceptible to overfitting. As a result, the algorithm performs well on frontal face detection but falls short in handling special and complex scenarios such as occlusion, posture, and expression.

In 2012, AlexNet [23] won the Imagenet [6] competition, which led to a resurgence of interest in Convolutional Neural Network (CNN) [25] and fueled rapid advancements in the field of computer vision. Following their success in image classification, CNN quickly surpassed the AdaBoost framework in accuracy for face detection tasks. Cascade CNN [27], as a representative study combining traditional techniques and deep networks, has a similar cascade structure to the VJ face detector, with multiple classifiers organized in a cascade. Cascade CNN differs in that it uses a CNN as its classifier at each cascade stage. It consists of six networks, named based on the input image size, as shown in Figure 1.1. The model is trained on images reduced using an image

pyramid [5] to handle facial images of varying sizes. The images are first proportionally reduced and then uniformly resized to 12x12 for training.

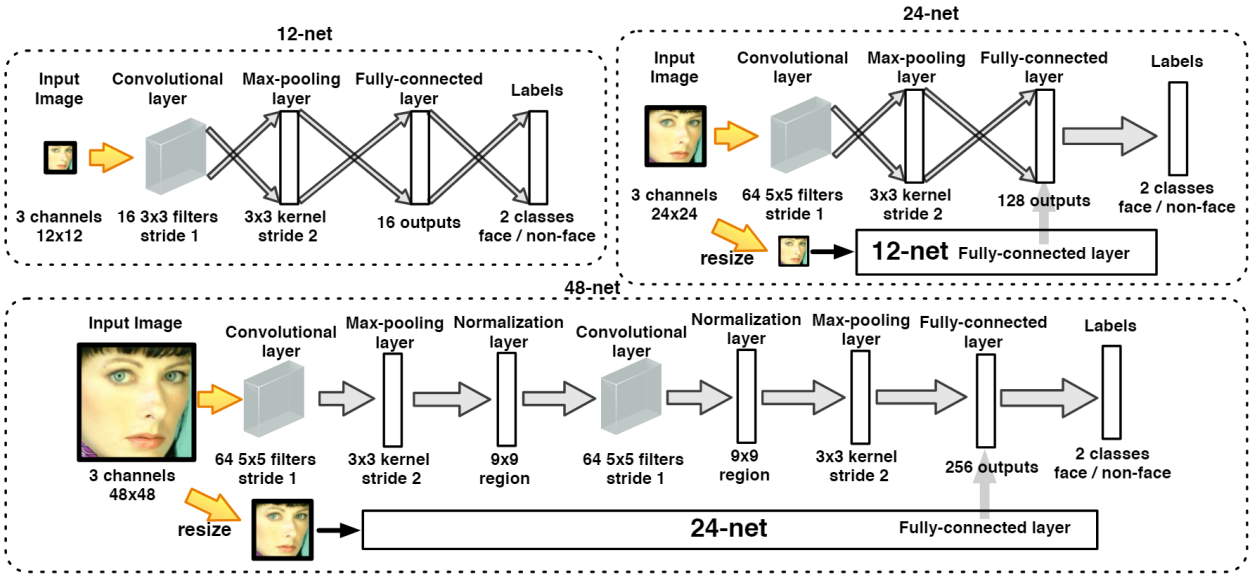

Figure 1.1: CNN structures of the 12-net, 24-net and 48-net

Cascade CNN effectively addresses the sensitivity to illumination and angle in open scenes, a common issue with traditional methods. However, the first-level network of the framework still uses window filtering with dense sliding windows, which limits its performance in high-resolution images with many tiny faces.

MTCNN is a multi-task face detection algorithm that combines face region and keypoint detection. Like Cascade CNN, MTCNN also employs a cascade framework, but with a more ingenious and efficient design. In contrast to Cascade CNN, the first layer of MTCNN uses a Fully Convolutional Network (FCN) [31] that can process images of multiple sizes, eliminating the need to limit input images to a single size.

The other two face detectors provided by Faceswap, SSD and S3FD, are different from MTCNN. SSD, a lightweight object detection algorithm, was added to the OpenCV deep neural network module (DNN) in version 3.3. The characteristic of the SSD algorithm is that the algorithm uses feature maps of different scales when detecting objects of different sizes, and the size of the object

is inversely proportional to the size of the feature map used for detection. Moreover, the size of the priors anchor used by the SSD algorithm is not fixed, and the scale and aspect ratio of the priors anchor can be adjusted according to the needs of the target detection work.

Although the SSD algorithm considers the detection of both large and small targets, it doesn't have a clear advantage over other detection algorithms when it comes to detecting small targets. Therefore, many algorithms aimed at improving the detection capabilities of small targets in SSD have emerged, and S3FD is one of them.

Inspired by the multi-scale mechanism in SSD, S3FD proposes a real-time face detection method that is single-shot and scale-invariant, based on the SSD network structure. An important change in this method is the introduction of a detection convolutional layer with wide anchor correlation, with a gradually increasing stride from 4 to 128. This allows S3FD to perform effectively in face detection across multiple scales.

1.1.2 Generative Models

A model that is generative is designed to produce random output that is both observable and predictive. In short, the generated samples are as similar as possible to the real samples. The primary objectives of generative models are to understand the underlying probability distribution of real samples in terms of their features, and to generate new data based on this understanding.

Generative Adversarial Networks. Generative Adversarial Networks (GANs) is a type of generative model that learns the data distribution through a confrontation between two networks. The network architecture consists of two opposing networks: a generation network, which generates samples that are as realistic as possible, and a discriminator network, which tries to identify whether the samples are real or fake. There are several advanced fake face generation algorithms that perform exceptionally well, such as StyleGAN [19]. It can manipulate facial attributes and generate high-resolution facial images with precise control over specific features or characteristics. For example, StyleGAN can control the visual features of each layer by modifying the inputs

to each layer individually, without affecting other layers. These features can include coarse features, such as pose and face shape, and detailed features, such as eye color and hair color.

Although the generative model can generate a fake face that the human eye cannot easily recognize, more auxiliary structures are required to perform face-swapping. For example, Face Swapping GAN (FSGAN) [35] proposed a face-swapping technique based on generative adversarial networks. This method can process the face image pairs that have not been seen in training well and can complete the face area of the source image according to the mask of the face area of the target image, which can better solve the face occlusion question. The architecture of FSGAN can be divided into three parts: replay and segmentation module, completion module, and fusion module. Of course, the premise is that the facial area is accurately extracted.

AutoEncoder. In 1985, Hinton et al. solved the problem of "backpropagation without guidance" by using input data as a guide and proposed the earliest form of the autoencoder [42]. Today, autoencoders are mainly used for data denoising and visual dimensionality reduction. They can also generate data due to the unique structure of autoencoders.

The general structure of an autoencoder is shown in Figure 1.2 and consists of an encoder and a decoder, which can be an arbitrary model. The encoder reduces the input data into a lower-dimensional encoded code. The decoder then generates data that is highly similar to the original input. Autoencoders train the parameters of the encoder and decoder by minimizing the difference between the input data and the generated data. Once the training is complete, passing unknown encoded codes to the decoder can also generate data similar to the original data.

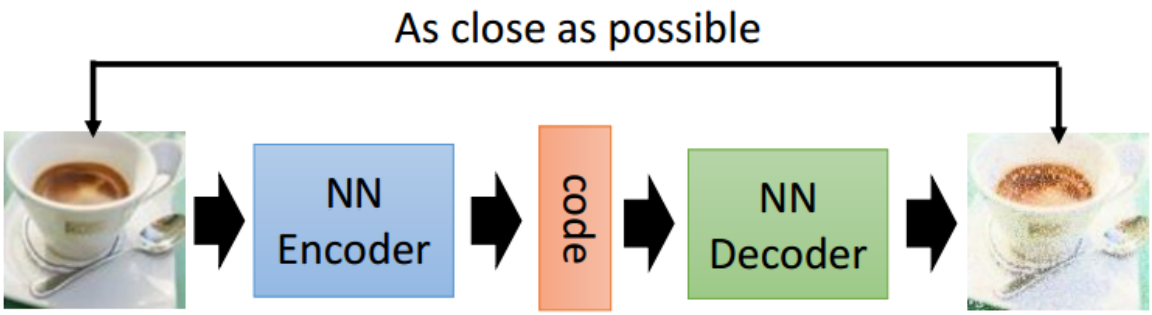

Figure 1.2: The structure of an autoencoder.

Figure 1.3 shows the training process of the Faceswap autoencoder generation model. During training, faces A and faces B use the same encoder but different decoders. After the training process is complete, the latent features generated from faces A can be passed through the encoder to decoder B, which will attempt to reconstruct the face from the encoded features of faces A. This results in an image that combines the body from face A and the face from face B.

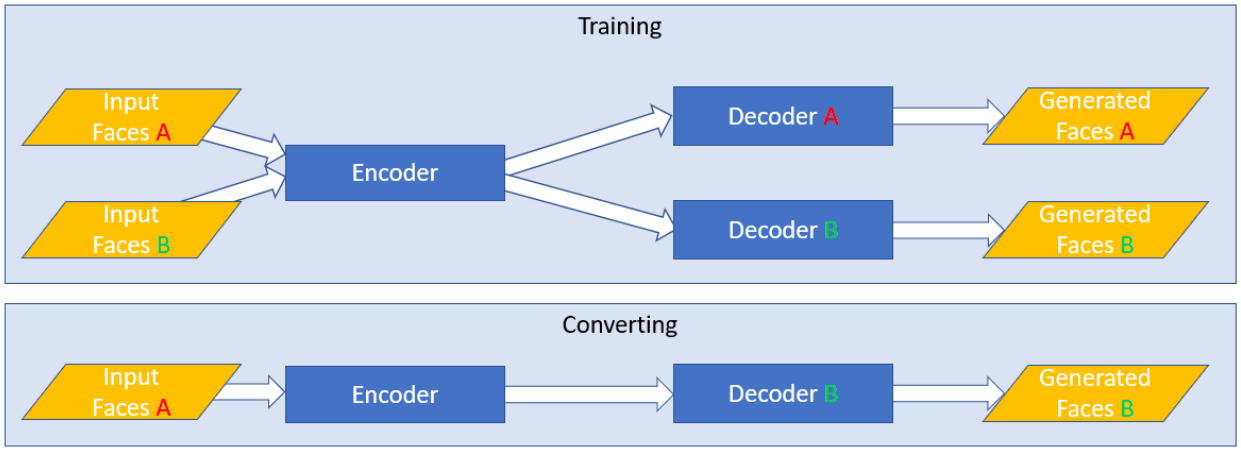

Figure 1.3: The Faceswap program performs face swap training and converting pipelines.

1.1.3 Adversarial Attack

In 2013, Szegedy et al. discovered the vulnerability of deep neural networks, giving rise to the field of adversarial attacks [49]. They found that the input-to-output mapping learned by deep neural networks is highly non-linear and discontinuous. We can make the network misclassify the image by applying some imperceptible perturbation, which is found by maximizing the prediction error of the network. Furthermore, these perturbations are not random, and the same perturbations can lead to wrong judgments in networks trained on the same dataset.

In 2014, Goodfellow et al. [14] proposed the concept of adversarial samples, and the specific process is shown in the figure 1.4. The article argues that the reason for the effectiveness of adversarial attacks lies in the linearity of deep neural networks in high-dimensional space rather than in the complexity of the networks. Based on this conclusion, Goodfellow et al.propose a more efficient method for producing adversarial examples, the Fast Gradient Sign Method (FGSM). It has been shown that even a classifier with a good test performance doesn't understand the underlying meaning of the classified samples in the same way that a human does. It just so happened that the model was built and worked reasonably well on the training data, giving the illusion of grandeur. However, when it encounters unusual points in space, the limitations of the model's capabilities become evident. As a result, more and more researchers have shifted their focus to the field of adversarial attacks, categorizing the different types of attack algorithms based on the methods used and the targets being attacked [1].

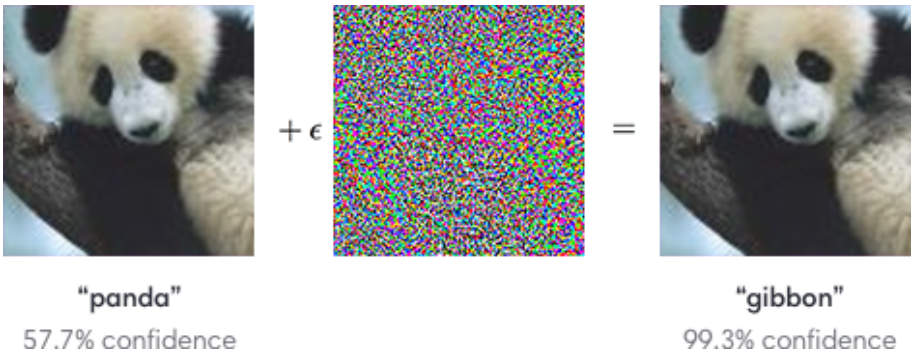

Figure 1.4: An adversarial input overlaid on an original image can cause a classifier to miscategorize a panda as a gibbon.

The classification of attack methods can be divided into the following three types.

White-box attacks: Attackers can obtain information such as the target model's network architecture, parameters, and gradients and then use this information to design adversarial examples. White-box attacks are widely studied because the disclosure of model structures and parameters helps people to understand the weaknesses of deep neural network models clearly and enables mathematical analysis.

Semi-black-box attacks: A semi-black-box attack means that the attacker has limited knowledge of the target model and uses generative methods to construct adversarial examples.

Black-box attacks: In black-box attacks, the attacker has no information about the target model. During this process, the attacker generates samples and based on the output results of the model, they can make further actions, making the attack more representative of practical scenarios

Classification by attack target refers to whether the attacker has control over the result of the attack on the neural network model. Assigning adversarial samples to a specific label is called a targeted attack, while attacks that do not require feedback from the model to determine the label are referred to as non-targeted attacks.

Among the adversarial attack methods for face detectors, it can be divided into digital attacks and physical attacks. Digital attacks objective to interfere with the model's detection by adding

imperceptible perturbations to the image's pixels, while physical attacks target the real world by using occlusions or creating attack patches.

In the daily application of artificial intelligence, adversarial attacks have certain harmfulness. For example, if a patch is attached to a stop sign on the side of the road, the unmanned detection system may make a mistake and continue driving, resulting in a risk of an accident [7]. Criminals can attach a patch to their bodies to deceive the target detection model, allowing them to commit theft and other crimes [50]. However, the nature of adversarial attacks to invalidate the model happens to be a tool for protecting the privacy of images.

With the continuous optimization of the convolutional neural network structure and the deepening of the number of layers, the face detector can more accurately detect the faces in the image, and the generative model can also generate clearer fake faces. It has brought a lot of challenges for researchers in the work of protecting facial privacy. Fortunately, adversarial attacks have also made a difference in face detection and generative models.

1.2 Research Objectives

The social significance of this study is to protect users' facial images privacy from malicious deepfake face swapping. This is a relatively broad goal, as the technology involved in deepfakes is quite widespread. By reviewing the previous research results that solve malicious face swapping using adversarial attacks, we believe that this problem can be solved by attacking the facial detection stage of the face-swapping software. The application of attacking face detectors is wider and not only prevents face swapping but also prevents users' facial images from being illegally searched and recognized. It can reduce the possibility of images being sold as datasets without authorization. We first lock the research object on Faceswap, which is fast, popular, and face-swapping software. Faceswap provides three different face detectors, MTCNN, SSD, and S3FD, but existing attack methods can not simultaneously make these three face detectors fail. There-

fore, the objective of this study is to propose a method that can simultaneously render these three face detection algorithms ineffective to help users who have privacy security needs for their facial images.

In order to achieve the objective of our study, we attempted to perform several experiments. Firstly, we needed to determine the effective attack range within the image, which helps us reduce ineffective perturbation and improve image clarity. Based on the attack techniques found in previous studies, within the scope of these attack techniques [28], we discussed the relationship between the background and the face and completed the confirmation experiment. Next, in order to better analyze the methods that cause MTCNN, SSD, and S3FD facial detectors to fail, we objective to first find the respective attack methods that make each of these three facial detectors fail. We found that the known white-box adversarial attack methods can cause SSD and S3FD facial detectors to fail. At present, there is no method for attacking MTCNN using invisible perturbations in the digital domain. We constructed a white-box multi-scale perturbation fusion attack algorithm to address this issue. In this study, we evaluate image quality using PSNR, SSIM, and LPIPS, and for the first time establish a baseline for image quality evaluation of this problem in the CelebA dataset [52] [56]. Our goal is to provide objective image quality scores as references for our subsequent work and other researchers. PSNR is an objective image quality evaluation algorithm, but it has certain limitations. To more comprehensively evaluate the quality of the adversarial samples, we also utilized SSIM and LPIPS as methods for evaluating the image quality. Although we have found a method for attacking MTCNN using invisible perturbations in the digital domain, it is challenging to attack the three facial detectors simultaneously due to the weak transferability of white-box adversarial attacks.

Lastly, We objective to limit the perturbation to the range of faces and render MTCNN, SSD, and S3FD face detectors ineffective. We proposed an attack method of adding black lines to the face to address this issue. Using this method, we conducted facial detection experiments on the CelebA and FFHQ datasets. The results show that our method can attack MTCNN, SSD, and

S3FD face detectors while restricting the perturbation to the range of faces. In this approach, we want to improve the quality of the images. We address this issue by optimizing the structure and attack method. Since completely covering the face with black lines is the minimum standard for image quality in this method, we used it as a baseline to determine if the image quality has improved. We conducted quantitative experiments on the image quality of the optimized images in CelebA and FFHQ datasets. The results showed that the image quality of the optimized images was superior to the baseline. Although this optimization method has significantly improved the quality of the images, our social goal is to help users in need. To this end, we conducted a user needs survey to confirm the practical application effectiveness of this method. The participants who select optimized images should be more than those who choose images with the entire face covered by black lines. The results of the survey showed that more users chose our optimized image than those who chose the image whose face was completely covered by black lines.

1.3 The Structure of This Thesis

Chapter 1 introduces the background and significance of this research, systematically sorts out the development and status of related technologies, and finally clarifies the purpose of the research.

Chapter 2 introduces the content of the related works. The techniques used in each experiment and the associated techniques are summarized and explained.

Chapter 3 confirmation the effective attack range of faces and backgrounds in images based on previous attack methods, using the MTCNN face detector and style transfer techniques.

Chapter 4 describes the first use of invisible perturbation in the digital domain to achieve an attack on the MTCNN face detector. In this chapter, a multi-scale fusion adversarial attack method based on perturbation interpolation is proposed for the unique properties of MTCNN. This method belongs to the white box attack. Moreover, a baseline for image quality evaluation

was proposed using the PSNR, SSIM, and LPIPS methods based on the CelebA dataset.

Chapter 5 presents an attack method that adds black lines to the face to disrupt the continuity of feature extraction. And the experiment of structure optimization and image quality optimization was conducted with the black line completely covering the face in the image as the baseline for image quality. Based on the experimental results, a questionnaire survey on the requirements for the secure use of image privacy was conducted, and the results were analyzed.

Finally, in Chapter 6, the research work of this paper is summarized, and the follow-up work direction of this research is discussed.

2 Related Work

2.1 Face Detection

This study consists of three experimental parts, with the ultimate goal of discovering a general method to attack face detectors. As a result, some related works are shared among each of the experiments. In this chapter, we will describe and summarize in detail the relevant research involved in all three experiments.

2.1.1 Convolutional Neural Network

Lecun [25] drew on the ideas of Fukushima [11] and proposed the original version of the convolutional neural network LeNet-5 [26] in 1998. The network structure of LeNet-5 is shown in Figure 2.1 and consists of two convolutional layers, two subsampling layers, two fully connected layers, and an output layer.

LeNet-5 mainly has three key characteristics to ensure that the image's feature information remains unchanged after operations such as panning and zooming.

Local receptive field. The receptive field refers to the size of the area where the pixels on the feature map output by each layer of the convolutional neural network are mapped back to the input image. In LeNet-5, the local receptive field of view can extract the basic features of the image. With the deepening of the network, the extracted features will also change from rough to detailed.

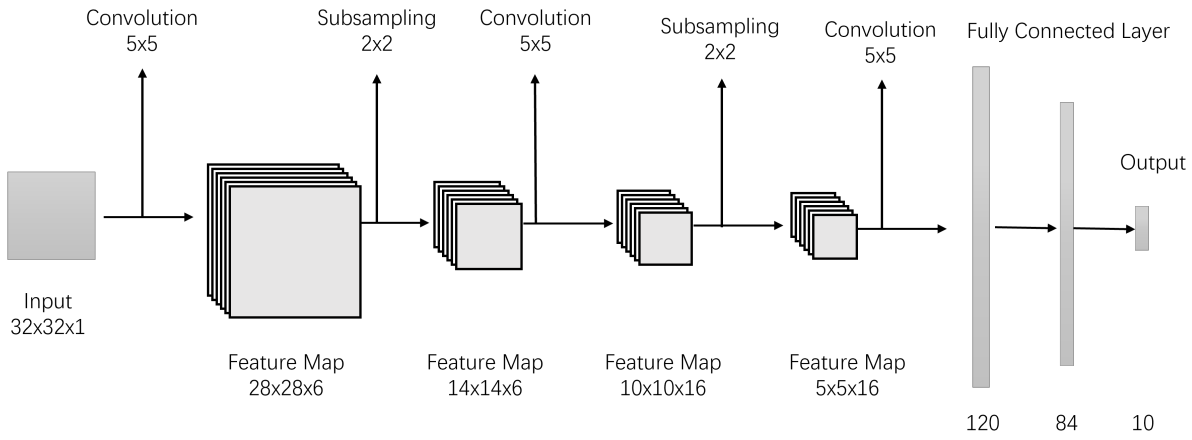

Figure 2.1: The network structure of LeNet-5.

Parameter sharing. Parameters in convolutional neural networks, also referred to as convolution kernels, consist of two-dimensional arrays of variable size. In computer vision, different convolution kernels with distinct values can be designed to perform specific recognition tasks. Parameter sharing can reduce the computational complexity of model learning and make similar judgments on the existence of multiple important regions on an image, enabling the model to have generalization capabilities.

Spatial or temporal subsampling.

If the content in the image moves or deforms, it will also change in the feature map. Therefore, the absolute position of the feature becomes less critical, and the relative position between the feature and the feature will contribute significantly to the model's judgment. Then the change of the feature position of the subsampling layer will not affect the result of each layer of the network. On the contrary, the subsampling layer's features simplify the positional structure, reduce the influence of absolute position on the calculation, and further decrease the weight of insignificant features.

LeNet-5 was the first model to train convolutional neural networks using gradient descent, and it still has some differences in settings from advanced convolutional neural networks such as AlexNet.

LeNet-5 does not employ zero padding in the convolutional layer, which leads to a reduction in the size of the feature map after the convolution operation. In order to keep the boundary information, or use a deeper network structure and other designs, researchers will fill zero around the feature map to increase the size of the feature map.

The receptive field of LeNet-5 does not scan already scanned regions, while in AlexNet, it does overlapping scans. That is, the stride size is smaller than the receptive field size.

LeNet-5 uses tanh instead of sigmoid as the activation function of the hidden layer. The output and input of tanh can maintain a nonlinear monotonous rising and falling relationship, which conforms to the solution principle of the backpropagation neural network and has better fault tolerance. In recent years, researchers often use the Rectified Linear Unit(ReLU) [34] as the activation function of the hidden layer. The non-saturation of ReLU can effectively solve the problem of vanishing gradient, provide a relatively wide activation boundary, and the convergence speed is also faster than sigmoid and tanh. The linear structure of the tanh, sigmoid, and ReLU are shown in Figure 2.2 from [46].

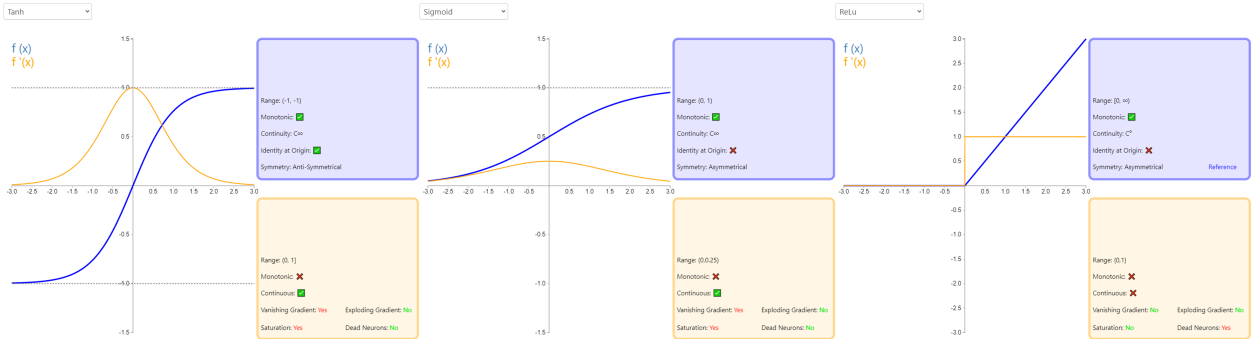

Figure 2.2: The blue curve is the shape of the function itself, and the yellow curve is the shape of the function after taking the derivative.

2.1.2 Multi-Task Convolutional Neural Network

2.1.2.1 Image Pyramid

Image pyramid is a multi-scale representation method of images. An image pyramid of an image is a bottom-up collection of images that gradually reduces the resolution of the original image, as shown in Figure 2.3. During the training phase, the image pyramid enhances the robustness of MTCNN to detect face images of different sizes. In the use stage, MTCNN continues to use image pyramids to process data, and different sizes of the same image can bring more reliable results for detection.

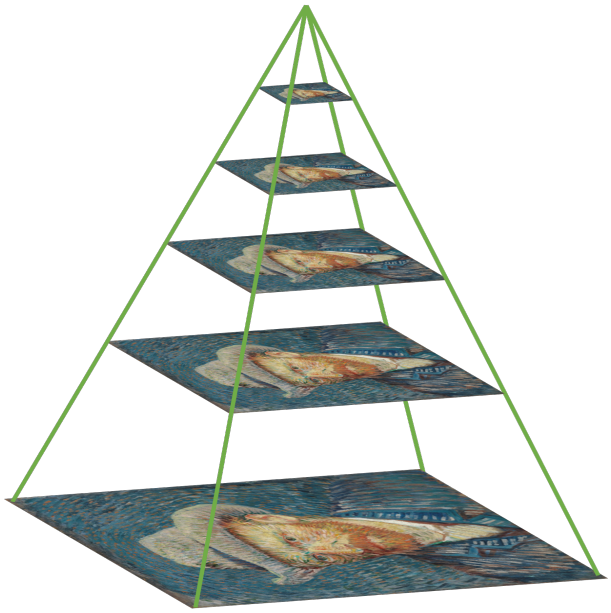

Figure 2.3: Image pyramids deal with the structure of an image.

2.1.2.2 Network Structure

Proposal Network(P-Net).

The reason why MTCNN is able to detect images of varying sizes is due to the use of fully convolutional networks in P-Net. Full convolution can accept inputs of any size, making it more computationally efficient than fully connected neural networks. Although fully convolutional

networks still have some limitations, such as not being precise enough and lacking sensitivity to the details of the input image, it is sufficient for the rough detection task in P-Net. This design is ingenious, as demonstrated in Figure 2.4 which shows the network structure.

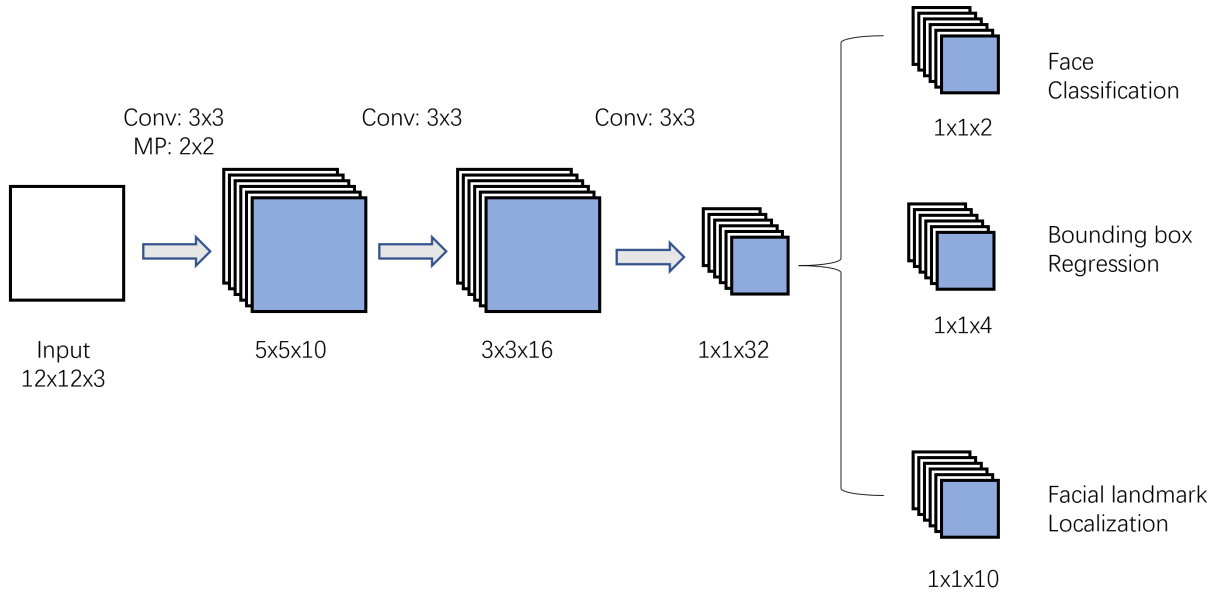

Figure 2.4: P-Net network structure pipeline.

Refine Network(R-Net). The basic structure of R-Net is similar to that of P-Net, except that a fully connected layer consisting of 128 neural units is added after the convolutional network, so that it can retain more effective features. The idea of R-Net is to use a more complex network structure to further select and adjust the rough bounding boxes provided by P-Net. To achieve the effect of high-precision filtering and face area optimization, the network structure is shown in Figure 2.5.

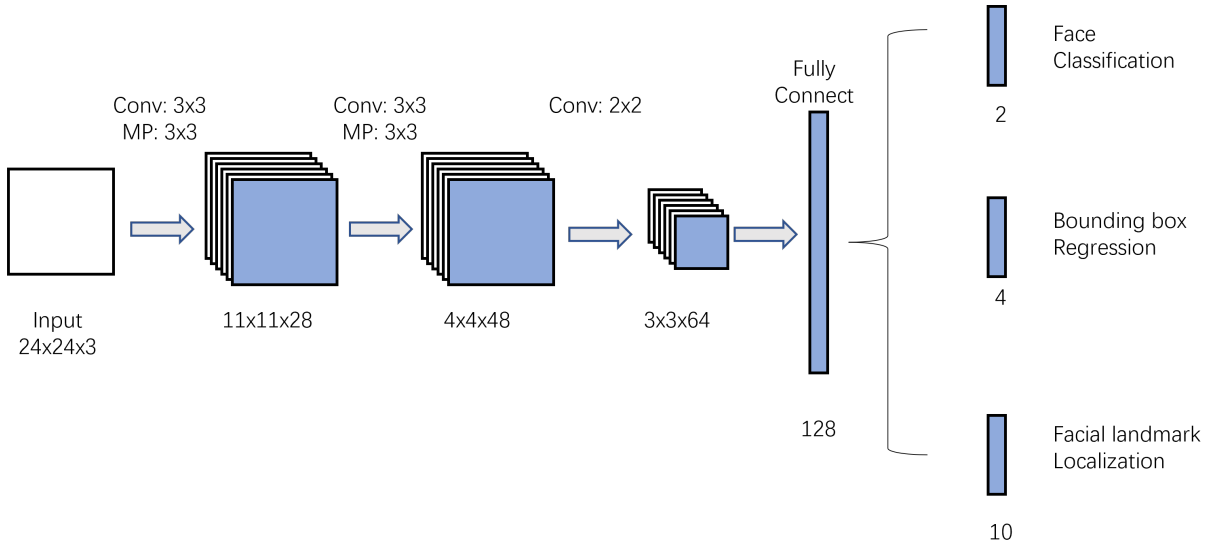

Figure 2.5: R-Net network structure pipeline.

Output Network(O-Net). Compared with R-Net, the network structure of O-Net has one more convolution layer. A fully connected layer with 256 neural units at the end of the network structure retains more image features. O-Net will perform face discrimination, face region bounding box regression, and face feature localization. The output results are the coordinates of the upper left corner of the face area, the lower right corner of the face area, and the eyes, nose, and mouth corners, a total of five keypoints. the network structure is shown in Figure 2.6.

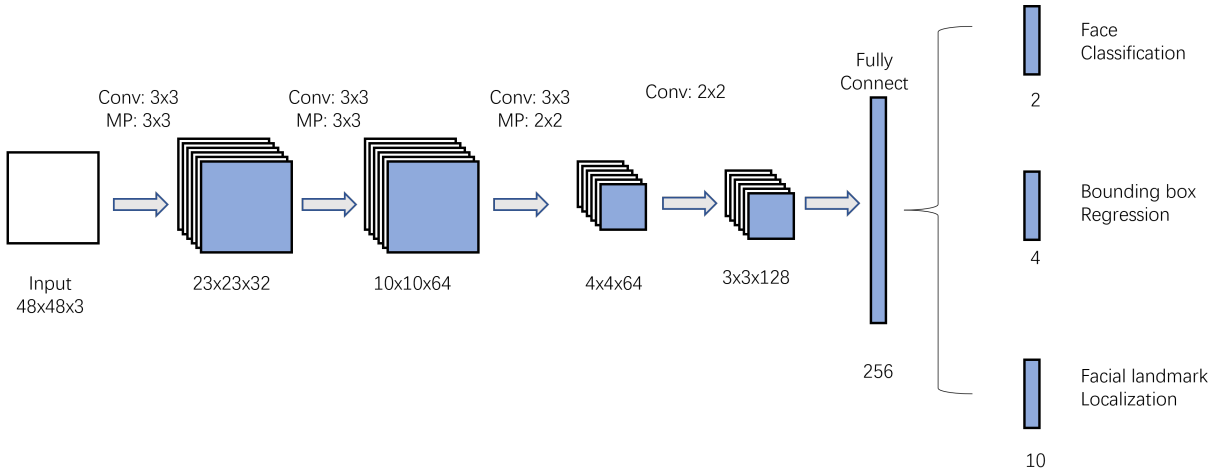

Figure 2.6: O-Net network structure pipeline.

In order to take into account performance and accuracy, MTCNN avoids the performance consumption caused by the traditional method of combining convolution kernel slides and classifiers. First, a simple model generates many bounding boxes with a certain probability. Then a more complex model is used for subdivision and higher-precision region bounding box regression to obtain the position of the facial bounding box and facial features. From the MTCNN detection process, it can be seen that the result transmission between the three network structures is absolute,. The absoluteness here means that if P-Net does not pass a rough bounding box to R-Net, then MTCNN judges that there is no face in the image. Therefore, this paper focuses on attacking P-Net, as it can reduce the number of perturbations and achieve the desired outcome.

2.1.3 Single Shot MultiBox Detector

In the name of SSD, Single shot indicates that it belongs to a one-stage object detection model [39], and MultiBox indicates that SSD is a multi-scale default box prediction.

The one-stage method of object detection is fast because it performs dense sampling of different positions and scales in the image and uses a convolutional neural network for feature extraction, classification, and regression in a single step. However, this uniform dense sampling approach is challenging to train, as the positive and negative samples are heavily imbalanced, which affects the accuracy of the model. SSD extracts detection results from feature maps of different scales and summarizes them into a loss function for training. Large-scale feature maps can be used to detect small objects, while small-scale feature maps are used to detect large objects, as shown in Figure 2.7.

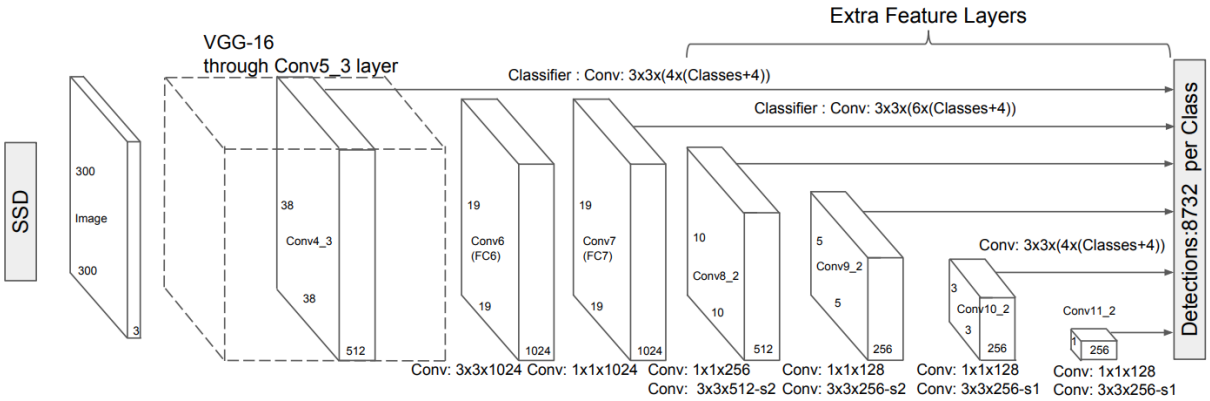

Figure 2.7: SSD network structure pipeline.

The multi-scale default box prediction of SSD may seem similar to the image pyramid of MTCNN, as both process features at different sizes to extract detection results. However, the two methods are actually quite different, which is why we conduct experiments to compare them in Chapter 3.

2.1.4 Single Shot Scale-invariant Face Detector

Although SSD is capable of detecting large and small objects at the same time, it is not particularly good at detecting small objects compared to other object detection algorithms. In response to this, there have been numerous efforts to improve the small object detection ability of SSD since its introduction. One such improvement is S3FD, a face detector based on SSD that addresses the issue of detecting tiny objects.

S3FD normalizes the feature row scale detected by each convolutional neural network. Aggregated and passed to a new predictive convolutional network. In this way, a scale fair structure is constructed, so that features at all scales can be given equal importance. The network structure is shown in Figure 2.8.

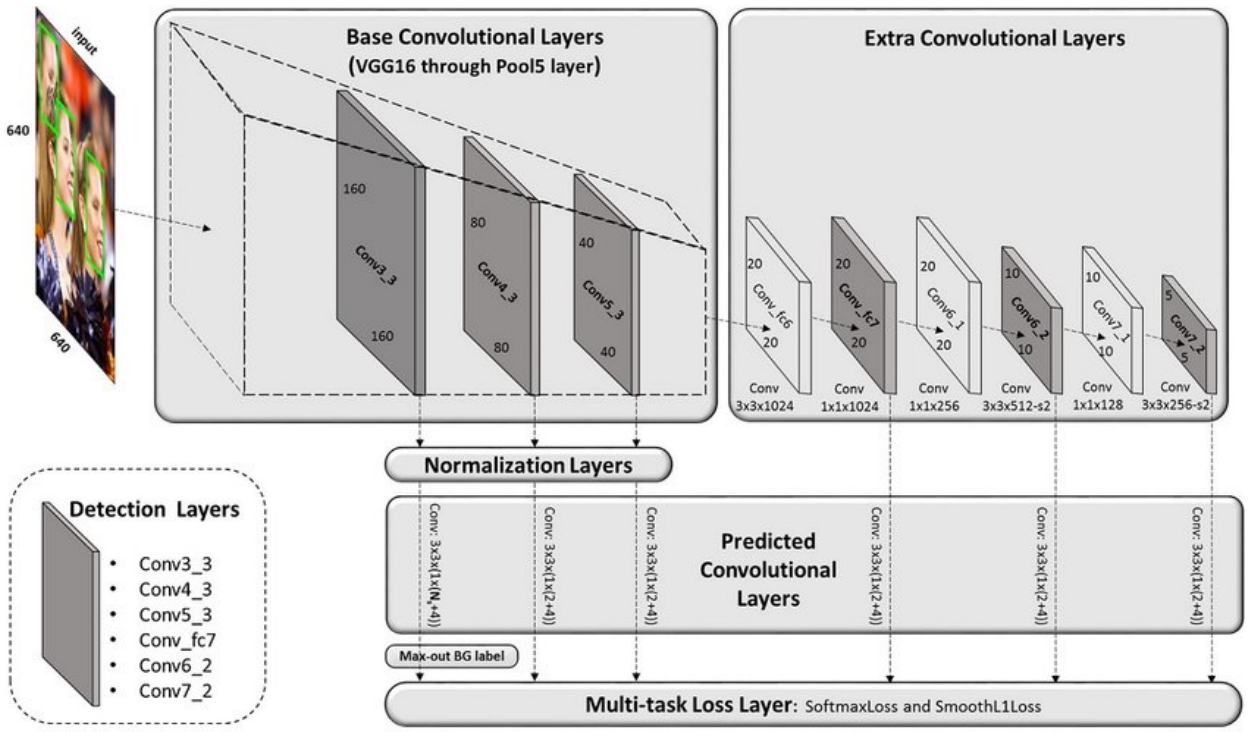

Figure 2.8: S3FD network structure pipeline.

Base Convolutional Layers. All layers from conv1_1 to pool5 of VGG16 [47] are used.

Extra Convolutional Layers. These layers can gradually decrease in size and form multi-scale feature maps.

Detection Convolutional Layers. The detection layers include conv3_3, conv4_3, conv5_3, conv_fc7, conv6_2 and conv7_2.

Normalization Layers. Scale normalization for conv3_3, con4_3, con5_3.

Predicted Convolutional Layers. Predict the features passed into the detection layer.

Multi-task Loss Layer. Summarize the prediction results and calculate the loss function.

S3FD and SSD operate the same on features, predicting features of various scales and using the aggregated prediction results to calculate the loss function. This is also the reason why they are different from MTCNN in their ability to resist adversarial attacks.

2.2 Adversarial Attack

Given a clear understanding of the network structure and parameters of the attack targets MTCNN, SSD, and S3FD, this article employs the gradient attack method of the original adversarial attack FGSM and the iterative approach and loss function design of Projected Gradient Descent (PGD) [33]. These adversarial attack methods are based on the model gradients, making them simple and powerful untargeted attacks, and they belong to the category of white-box attacks.

2.2.1 Fast Gradient Sign Method

The concept behind FGSM is based on the loss function used to train the model parameters. During training with stochastic gradient descent, the value of the loss function produced by the model decreases, resulting in the network making correct predictions. However, if the calculated loss value is added to the input image, causing the loss value produced by the model to increase, the model is more likely to produce incorrect predictions.

2.2.1.1 Pipeline

The meaning of each symbol in the loss function. The attacked original image is x_{ori}, whose label is y. A classification model M, and the parameters θ of the classification model. FGSM generates the adversarial attack perturbation η.

First, use the classification model M to perform a forward propagation on the input image x_{ori}. The loss function value $\nabla_{x_{ori}} J(\theta, x_{ori}, y)$ is calculated at this time. Due to the uneven size of the pixel values for calculating the loss function value, in order to control the maximum value of the loss function value, the sign function $sign()$ is used to extract the gradient direction instead of using the gradient value directly. And use ε to control the amplitude of the disturbance and satisfy $\|\eta\| < \varepsilon$.

The generation process of adversarial samples x_{adv} is Formula 2.1 and Formula 2.2.

$$\eta = \varepsilon sign(\nabla_{x_{ori}} J(\theta, x_{ori}, y)) \tag{2.1}$$

$$x_{adv} = x_{ori} + \eta \tag{2.2}$$

Although FGSM is a white-box attack method, the adversarial samples it generates have certain effects in the face of black-box models and are generally used as the algorithm baseline for comparison.

2.2.1.2 Linear Validity Analysis

This section analyzes the linear validity of FGSM using the trained model's computational approach. It is assumed that the weight vector is n dimensional and the length of each dimension is m. During forward propagation, as Formula 2.3.

$$W^T x_{adv} = W^T(x_{ori} + \eta) = W^T x_{ori} + W^T \eta \tag{2.3}$$

Not only η affects the calculation of forward propagation, but W^T also occupies a certain proportion. Since $W^T = mn$, as the weight vector dimension grows, the perturbation will have more and more influence on the model.

2.2.2 Projected Gradient Descent

PGD is not a straightforward improvement of FGSM, but instead is based on the Basic Iterative Method (BIM)) [24]. In PGD, random perturbations are added to the original input samples within their neighborhood, and adversarial samples are generated through multiple iterations. PGD has significantly improved performance, and boasts better migration and resistance to destruction compared to FGSM. By "resistance to destruction," it means that if a model can withstand an attack from PGD, it will also have strong resistance against other first-order adversarial attack methods.

FGSM is a first-order attack that only adds the perturbed amount of the gradient once to the image. But if the target of the attack is a complex nonlinear model, such an approach is likely to fail. The complex nonlinear model may change drastically in a very small range, so the large gradient span becomes the weakness of FGSM. PGD reduces the step size constraint and finds effective disturbances by frequently updating the perturbation. And will use $\prod_{x+S}$ after each iteration to clip the perturbation to a range.As Formula 2.4

$$x^{t+1} = \prod_{x+S}(x^t + \varepsilon sign(\nabla_x L(\theta, x, y))) \tag{2.4}$$

3 Correlation between Background and Face

3.1 Introduction

When performing adversarial attacks on object detection models, researchers typically focus on adding perturbations to only the area where the target is located in the image in order to increase efficiency and maintain the overall clarity of the image. It seems that ignoring the background in images is a common practice, and object detection models are no exception. However, it should be noted that this practice of focusing on the target area alone does not necessarily prove that attacking the background is ineffective or that attacking the target area is sufficient. For instance, Li et al.'s [28] research demonstrates the efficacy of hiding faces in the training data to prevent AI face synthesis by destroying the training data. The target model in their attack was a face detector based on Fast-RCNN [12] and SSD. Similar to other methods of attacking target detectors, they designed a formula for generating perturbations based on the loss function. Interestingly, their method was able to transfer the bounding boxes detected by the face detector to parts of the background outside the face, leading to an increase in false positives and false negatives, as shown in Figure 3.1. This caught our attention. Suppose that invisible perturbations are employed to enhance the detection probability of the background portion. In such a scenario, it can attack the face detector, which is undoubtedly the optimal strategy to preserve the clarity

of the face. Therefore, we conducted this confirmation experiment to examine whether the use of style transfer to enhance the detection probability of the image background can cause a facial detection system to malfunction.

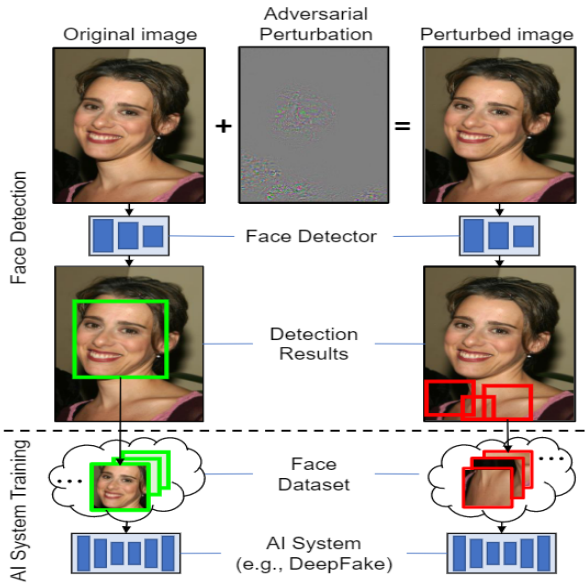

Figure 3.1: Provide non-facial training data for AI facial synthesis models.

This section analyzes Li et al.'s research in terms of attack goals and methods and does not cover the implementation details or feasibility of the research. The goal of Li et al.'s attack is to provide non-facial data for training AI face synthesis models, thereby destroying the model's training results and preventing AI face synthesis. For this study, their research had two questions:

Question 1. While training the model with non-face data may make it difficult for the model to generate realistic faces, in the Faceswap process there is a step of manually filtering bounding boxes, which can remove data that does not meet the training criteria.

Question 2. If only adding perturbations in the background area of an image causes the face detector to misidentify the background area as a face, does it contribute to lowering the probability of face detection?

Question 1 indicates that this research approach is limited, and Question 2 is the issue that we will discuss in this section. Under the conditions of Question 2, we discussed whether it is possible

to transfer facial features using style transfer techniques and only add them in the background area of the image, thereby achieving the goal of attacking the facial detection algorithm, MTCNN. This will help us clarify the focus of the follow-up research work.

3.2 Neural style transfer

Image style transfer is a technique that combines the style of one image with the content of another image. As shown in Figure 3.2, the Content Image provides the content while the Style Image provides the style. The target image is generated by taking the weighted sum of the content loss and the style loss values, as defined by the loss function.

$$J(G) = \alpha J_{content}(C, G) + \beta J_{style}(S, G) \tag{3.1}$$

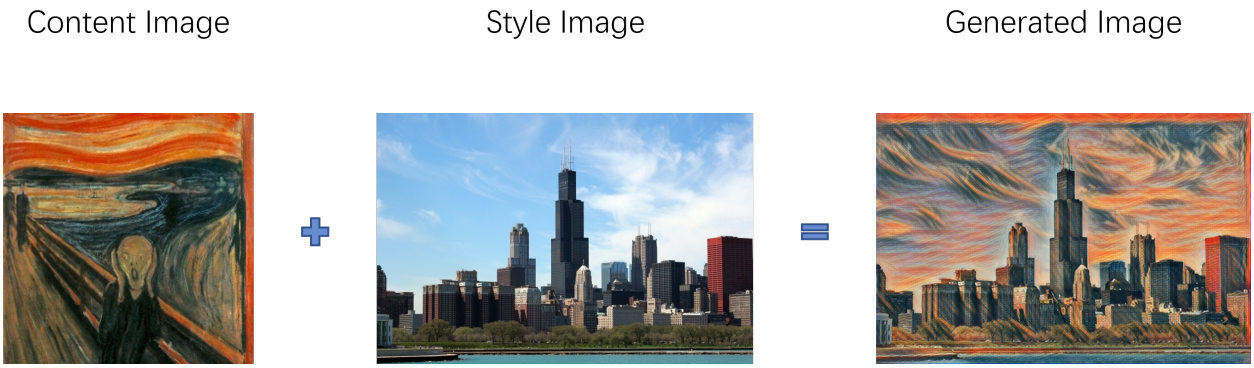

Figure 3.2: The neural style transfer process.

Typically, the pre-trained VGG16 network is utilized to extract both content and style information from images. As previously discussed in Chapter 2, a trained convolutional neural network (CNN) model has the ability to extract features from images. The information extracted by the network at different image locations can vary, with shallow networks tending to extract detailed textures and deep networks extracting more abstract information, such as outlines, shapes, and sizes. The correlation between the layers is used to calculate the style information of the image.

3.3 Method and Experiment

In the experiment, we use MTCNN's P-Net to perform rough facial detection on the image to utilize both the image background and facial information. To clarify the experimental results, the threshold of P-Net in MTCNN is set to 0.9 to reduce the candidate boxes, resulting in a small number of background borders and facial boundary boxes as illustrated in Figure 3.3. Three non-face regions, along with a face region, are extracted from the detected bounding boxes. These non-face regions include characters that are generated by the false detection of the detector. However, this experiment only regards these characters as part of the background, disregarding their actual meaning. Finally, these images are utilized for style transfer to enhance the facial features in the background area.

Figure 3.3: MTCNN's P-Net network for rough detection and extraction of the bounding box content.

In this experiment, the facial images provide the style information during the style transfer process, while the background images provide the content information. After the transfer is completed, the generated image is restored to its original position, and the facial detection is performed using the P-Net of MTCNN, as depicted in Figures 3.4 and 3.5.

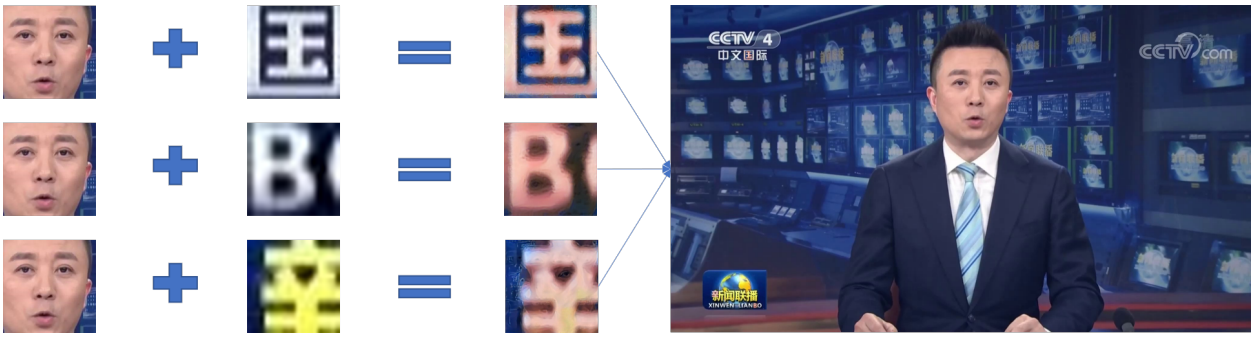

Figure 3.4: The images of the face region and the images of the three non-face regions are style transferred and restored to their original positions.

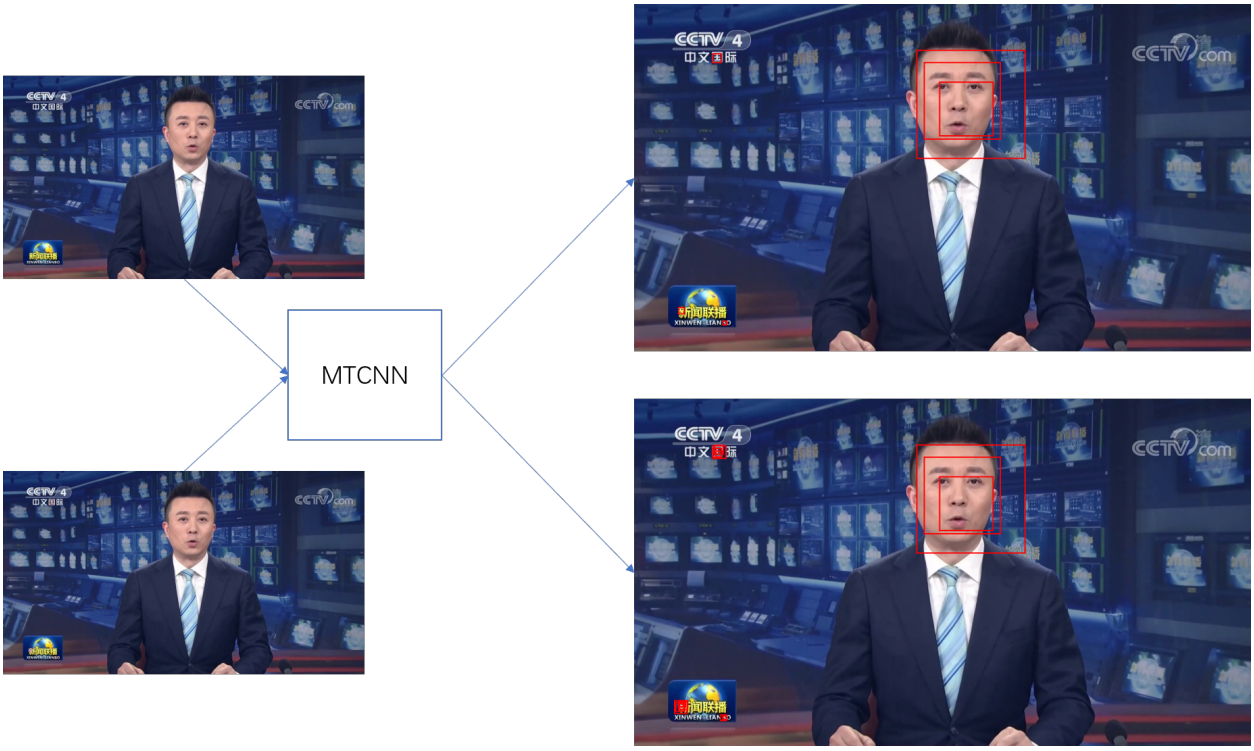

Figure 3.5: Before and after the style transfer, the facial detection results are compared.

3.4 Discussion and Conclusions

In this section, the technique of style transfer is applied to introduce facial features into the background of an image, without altering the face itself. Figure 3.5 presents a comparison between the original image and the stylized image on the left, as well as the result of P-Net detection by MTCNN on the right. An examination of the number of bounding boxes reveals that incorporating facial features into the background can mislead the face detection algorithm, causing it to generate more bounding boxes in the background region. Nonetheless, there is no significant change in the number of bounding boxes in the face region. The effectiveness of Li et al. is possibly due to the fact that they added perturbations not only to the background but also to the facial region. Thus, increasing the detection probability in the background using style transfer does not affect the detection results of MTCNN facial detector for faces in the image. In future studies, we will shift our focus to targeting the face area directly with adversarial attacks.

4 Multi-scale Perturbation Fusion Adversarial Attack of MTCNN Face Detection System

4.1 Introduction

In this section, we designated the face region in the image as the target for attack and applied adversarial attacks on MTCNN, SSD, and S3FD. However, after conducting further research and simple experiments, we discovered that existing adversarial attack methods could easily compromise SSD and S3FD, as depicted in Figure 4.1. Additionally, adversarial attacks have been shown to compromise face detectors that are based on the SSD structure [32]. Therefore, the experiments in this section aim to invalidate the MTCNN face detector.

In the field of physical attack, it has been possible to attack the MTCNN face detector by generating perturbation patches to make it invalid. The method of Edgar et al. [20] uses black and white checkered patches on the left and right sides of the face as the initial data. The patches are continuously added or updated perturbations using an iterative gradient attack method. Then print out the generated patch and stick it on your face to achieve the attack effect, as shown in Figure 4.2. In their work and other researchers' reproduction experiments, they will wear decorations, such as hats, glasses, masks, etc. The impact of these accessories on the attack

Figure 4.1: In the digital domain, attacks against whole images and faces can disable SSD-based face detectors.

results is not specified in the experiment. Additionally, the perturbations added through the method proposed by Edgar et al. are visible, reducing the overall usability of the image.

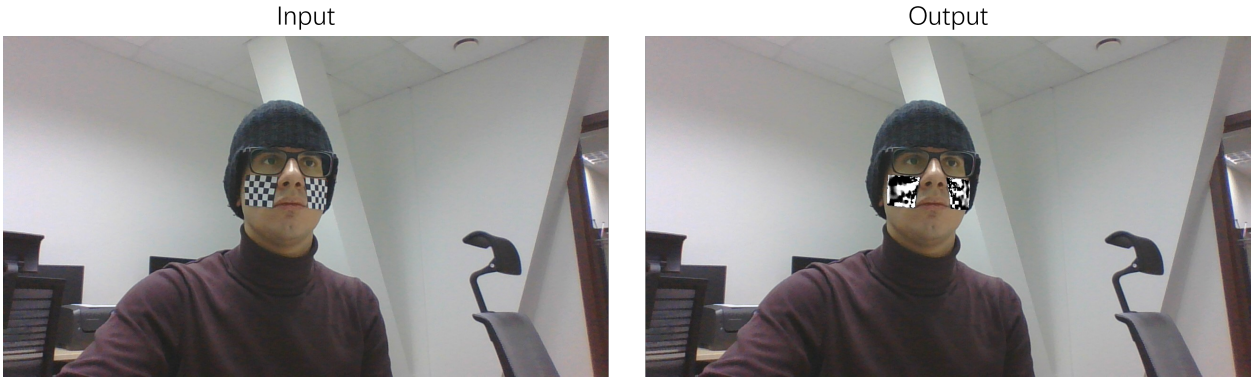

Figure 4.2: Training data and adversarial samples for attacking MTCNN in the physical domain.

However, in the digital domain, there is currently no method for using invisible perturbations to compromise the MTCNN detector. We believed it was due to MTCNN's use of image pyramids for processing and its aggregation of multi-scale detection results for face detection. When the adversarial sample is resized using the image pyramid, the number and intensity of the perturbations also change. As a result, the adversarial samples fed into the model are prone to losing their ability to perform the attack, resulting in the failure of the attack. In other words, adding perturbations to images of only one size is not sufficient. After iterating the gradient to find the perturbations, the process of searching for effective perturbations must be repeated for images

of different sizes, and then the perturbations should be fused to produce effective adversarial samples. The experiment in this section uses the idea of an iterative gradient attack to attack the MTCNN face detection system and interpolates and fuses disturbances in multiple sizes. The method is resistant to compression and changes in scale.

4.2 Image Interpolation

Image Interpolation is a resampling technique that calculates values for pixels that cannot be directly determined during an image's geometric transformation. The commonly used interpolation methods are Neighbor Interpolation, Bilinear Interpolation, and Bicubic Interpolation [17] [21]. In this study, we use Bilinear Interpolation and Bicubic Interpolation to amplify the perturbation. Bilinear Interpolation calculates new pixel values based on the weighted average of the four closest pixels and has fast speed and strong pixel continuity. When reducing the image size, we use the best performing method, Area Interpolation, as implemented in OpenCV. This method uses mean filtering and assigns equal weights to all values during the interpolation process. Figure 4.3 shows Bilinear Interpolation.

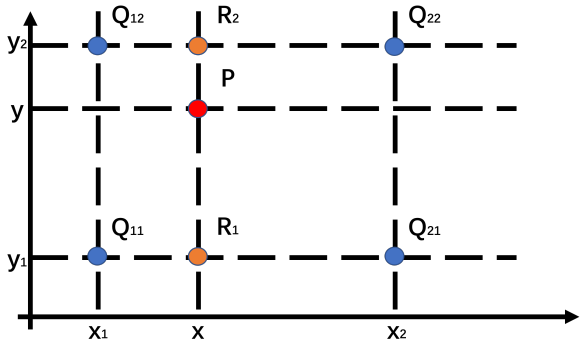

Figure 4.3: In bilinear interpolation, the associated four-pixel values are calculated.

First, we calculate the values of points R1 and R2, and then use those values to calculate the value of point P. This way, each new pixel is influenced by its surrounding four pixels, preserving the weight of the original pixel value. The specific calculation formula is as follows Formula 4.1 and Formula 4.2 and Formula 4.3.

$$f(R_1) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21}) \tag{4.1}$$

$$f(R_2) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}) \tag{4.2}$$

$$f(P) \approx \frac{y_2 - y}{y_2 - y_1} f(R_1) + \frac{y - y_1}{y_2 - y_1} f(R_2) \tag{4.3}$$

Bicubic Interpolation is a more sophisticated method that results in smoother image edges compared to Bilinear Interpolation. The algorithm uses the grayscale values of 16 points surrounding the sample point for cubic Interpolation. It takes into account the grayscale values of the four directly adjacent points and the rate of change of the grayscale values between adjacent points. Bicubic Interpolation provides a closer approximation to the high-resolution image enlargement effect, but the computational cost is greater than that of Bilinear Interpolation. The weight function of Bicubic Interpolation is as follows Formula 4.4,

$$W(x) = \begin{cases} (a+2)|x|^3 - (a+3)|x|^2 + 1 & \text{for } |x| \leq 1 \\ a|x|^3 - 5a|x|^2 + 8a|x| - 4a & \text{for } 1 < |x| < 2 \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

where a is typically set to -0.5 or -0.75. The value of x is the horizontal or vertical distance from the interpolation point to the source coordinates.

4.3 Method

This section details the scope of adversarial attacks, interpolating perturbations, and the design of the loss function.

Attack Area. In terms of the selection of image scale, our approach differs from that of Edgar et al. They selected three critical scales for P-net to target, while we adjust the scale to be targeted based on the number and size of faces in the image. We use the approximate detection area identified by P-net as the area to add perturbations in order to minimize contamination to the image. The position, number, and size of the detection frames may change after each attack, making our method more precise compared to other attack methods.

Attack Steps. We employ a double loop approach to find effective perturbations. The first loop iterates over images of different sizes, while the second loop adjusts the size of the perturbations until a valid perturbation is found.

Loss Function. We sum the confidences of the detection boxes detected in the second-layer loop in a particular proportion, and to continuously reduce the confidences, we take the negative sum as the loss value.

Search Rate. We normalize the image values in MTCNN to the range between -1 and 1 during image preprocessing. The gradient values obtained through the sign function are -1 and 1, so we chose to design the search rate as a ratio of the original image value, such as 1/255 or 0.1/255.

Perturbation Interpolation. The perturbation interpolation consists of three parts, and Algorithm 1 outlines the detailed process of the algorithm.

a) In the second loop, we maintain a temporary array of the same size as the image in the current loop. After each iteration, we superimpose the perturbation on this temporary array. If the loss value is below the threshold or no detection box is returned, we upsample the temporary array to the same size as the original image using bilinear Interpolation, and then add it to the clean original image.

b) Taking the original image with added perturbation as input data, we reduce it to the same scale as the current loop using interpolation and continue searching for perturbations until the reduced image no longer requires any additional perturbations.

c) We accumulate the effective perturbations from each scale and enlarge them using interpolation. Finally, we add these perturbations to the clean original image to obtain the final effective adversarial samples.

Algorithm 1 By scaling the perturbation to find adversarial examples that are resistant to size changes.

Input: Original image, I_o;
The size by which the image will be scaled, S;
P-net of MTCNN face detection system, N_p;
Control the threshold of the detection box, t.
Output: Valid attack samples for each size.

Initialize An array with the same size as I_o and a value of zero, used to store the final perturbation, PI_o.
for each $s \in S$ **do**
  I_s = I_o.resize(s, bilinear interpolation)
  PI_s = I_s.data(value=0) // store temporary perturbations
  while $True$ **do**
    prob $\leftarrow$ $N_p(I_s)$
    loss $\leftarrow$ createLoss(prob) // loss is less than or equal to zero
    boxes $\leftarrow$ findBox(prob, s, t)
    sr $\leftarrow$ searchRate()
    pertur $\leftarrow$ sr $\times$ I_s.grad.sign()
    if loss $\geq -t$ or boxes is None **then**
      temp = PI_s.resize(I_o.size, bilinear interpolation)
      I_s= (temp + I_o).resize(I_s.size, bilinear interpolation)
      if loss $\geq -t$ or boxes is None in the next loop **then**
        PI_o + = temp
        break
      else
        continue
      end if
    end if
    I_s[boxes] + = pertur[boxes] // modify the data within the bounding boxes
    PI_s[boxes] + = pertur[boxes]
  end while
end for
return $I_o + PI_o$

4.4 Experiment

We will conduct three experiments using the same MTCNN parameters as Edgar et al. The minimum size in the image pyramid is 21 pixels, the thresholds for the three subnetworks are 0.6, 0.7, and 0.7, and the scale step factor is 0.709.

Experiment 4.4.1 will demonstrate that the noise generated by our algorithm can make MTCNN unable to detect faces in various photos. In experiment 4.4.2, we will compare the structural similarity between adversarial examples and the original images. Experiment 4.4.3 aims to determine the most effective scales for attacking different images.

4.4.1 Effectiveness Experiment

Single-Person Photo. We attack four single-person photos of different sizes and styles, and the results are shown in Figure 4.4. Each column in the image is a comparison group to compare the image changes before and after the attack. Each row represents a different state of the image. The first line is the original image, the second line is the adversarial samples, and the third line is the perturbed image, which is the difference between the original image and the adversarial sample. We randomly selected 5000 images in the CelebA [30] dataset whose faces can be detected by MTCNN, all of which are single-person images of size 178x218. After adding perturbation to these 5000 images using Algorithm 1, 3815 images cannot be detected by MTCNN, and the detection success rate reduce to 23.7%. The interpolation method used is a bilinear interpolation.

Figure 4.4: From left to right, the original sizes of the images are 128x128, 178x218, 300x232, and 3840x2160.

Multi-Person Photo. In a similar manner as the experiment with single-person photos, we performed an attack on two multi-person photos to generate adversarial samples and perturbed images, as shown in Figure 4.5. Each line in the image represents a comparison group to compare the changes in the image before and after the attack. Each column represents a different image state. The first column is the original image, the second column is the adversarial sample, and the third column is the perturbed image, which is the difference between the original image and the adversarial sample.

Changes in Perturbation and Loss. Figure 4.6 shows the numerical distribution before and after the perturbation interpolation, as well as the loss value change at the same scale.

(e)

(f)

Figure 4.5: The original size of the two images is 1300x867, 1024x820.

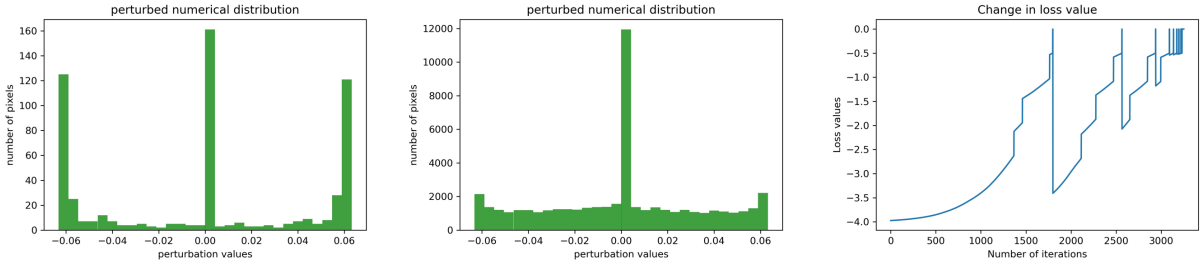

Figure 4.6: Taken from an image of size 128x128, downscaled to 0.1.

4.4.2 Structural Similarity

We compare the structural similarity of the six images in Experiment 4.4.1 using the SSIM [52] and LPIPS [56] metrics. The SSIM is a widely used similarity comparison method, where the similarity value ranges from 0 to 1, with a value closer to 1 indicating greater similarity. LPIPS, on the other hand, provides a similarity measure observed by deep learning models, and in this experiment, we use the VGG19 [47] model. A value closer to 0 indicates greater similarity as

determined by LPIPS. The results are presented in Table 4.1.

Table 4.1: Use the original image size as the data for the structural similarity comparison results.

	(a)	(b)	(c)	(d)	(e)	(f)
LPIPS	0.012	0.014	0.029	0.054	0.037	0.018
SSIM	0.968	0.970	0.955	0.971	0.943	0.978

Since this experiment was the first to use invisible perturbations to attack the MTCNN face detector, there is a lack of existing image quality evaluation baseline. Therefore, we used 2000 images from the CelebA dataset to perform a quantitative image quality evaluation of the algorithm. The evaluation experiment used the PSNR algorithm, which is quite objective in the field of image quality evaluation, and the SSIM and LPIPS algorithms were used for supplementary evaluation, as shown in Table 4.2.

Table 4.2: The "Original" row shows the evaluation score of image quality when the image has not been modified. The "CelebA" row shows the adversarial sample image quality scores in the CelebA dataset.

	PSNR	SSIM	LPIPS
Original	Inf	1	0
CelebA	30.25	0.9	0.06

4.4.3 Effective Scales

Figure 4.7 shows the bounding boxes detected by the p-net of MTCNN at different scales with the same image size and different face sizes. It can be found that the scale effect for the attack is related to face size. The original size of the image is 1440x900, and we filled the four times reduced image with white to get an image of the same size as the original image but with a different face size. The numbers in the figure represent the bounding boxes that can be detected by p-net at different scales in the original image and the reduced image.

Figure 4.7: Effective size comparison image.

4.5 Discussion

In Experiment 4.4.1, we utilized single and multi-person photos of varying sizes, which covered faces with diverse skin tones and distances from the camera, to demonstrate the general applicability and efficacy of Algorithm 1 in attacking MTCNN.

During the attack process, there are two data that need our attention, one is the original adversarial sample, and the other is the interpolation adversarial sample. The size of the two is the same. When the pixel value of the interpolation adversarial sample is close to the original adversarial sample, the interpolation adversarial sample can have the ability to attack while having the ability to resist the pyramid image. For the convenience of observation, we intercepted the pixel values with a size of 4x4 in the same position in the original image, the original adversarial example, and the interpolation adversarial example, as shown in Table 4.3. The first column of the table represents different images, including the original image, the original adversarial example image, and the interpolation adversarial example image. The 4x4 matrix to the right of the "original image" represents the pixel values in the original image. The 4x4 matrix to the right of the "original adversarial example" represents the pixel values in the original adversarial example image. And the 4x4 matrix to the right of the "interpolation adversarial example" represents the pixel values in the interpolation adversarial example image. The values in the tables correspond

to the pixel values in the images, and we have bolded a representative number in each image column of the table. When the pixel values of the original adversarial example and the original image differ greatly, it proves that during the perturbation process, these pixel values will be continuously added or decreased, and the position of these pixel values is the focus of the attack. If the interpolation adversarial sample can maintain a numerical difference that is similar to the original adversarial sample at these pixel values, then the interpolation adversarial sample will possess the attack ability of the original adversarial sample. The values in the table confirm that our method can preserve the attack power of the original adversarial examples.

Table 4.3: Before and after interpolation, the value changes of the adversarial samples are compared.

original image	-0.1344	-0.1504	-0.1130	-0.4776
	0.0006	**0.2118**	**-0.0343**	0.0276
	0.0843	0.3351	-0.4455	0.2229
	0.1381	0.1870	-0.0496	**0.0380**
original adversarial example	-0.1344	-0.1504	0.0669	0.2208
	-0.9487	**-0.7002**	**0.8679**	-0.3899
	0.3985	1.3348	-0.4455	-0.1758
	-0.6683	0.0260	0.9404	**-0.9305**
interpolation adversarial example	-0.1743	-0.1903	0.1052	0.1688
	-0.8793	**-0.5564**	**0.7289**	-0.2951
	0.3456	1.2013	-0.3702	-0.1962
	-0.6072	0.1012	0.8197	**-0.8412**

The two histograms in Figure 4.6 depict the distribution of pixel values for a small-sized perturbation and the distribution of pixel values after Bilinear Interpolation. The Bilinear Interpolation preserves the distribution of the perturbation and results in a smoother distribution. This allows the perturbation to have more weight during Area Interpolation, ensuring that the reduced image remains an effective adversarial sample. The loss value image shows the process of finding a valid perturbation through multiple interpolations of the perturbation. The drop in loss value in the graph is caused by interpolation.

Figure 4.7 illustrates the bounding boxes detected by P-Net at various scales for the same

image. Our findings indicate that when the face in the original image is larger, the features of the small-scale image offer more information for MTCNN. On the other hand, when the face in the image is small, the features of the medium-scale or even large-scale image become critical for face detection, and the small-scale image is insufficient for providing an effective bounding box for R-Net.

Algorithm 1 has not optimized the amount of perturbation, resulting in limitations in controlling facial color changes. Despite this, the attack focus is on the face, and the structural similarity between the adversarial samples and the original image remains close. In this regard, we propose the image quality evaluation baseline within the scope of this experiment to provide an objective reference for our subsequent work and other researchers.

We compared the distribution of pixel smoothness between bicubic interpolation and bilinear interpolation. As shown in Figure 4.8, Bicubic Interpolation does not produce distinct pixel boundaries. The adversarial samples generated through Bicubic Interpolation are closer to the original image in terms of structural similarity when compared to those generated through Bilinear Interpolation, but the training time for each image increases significantly. For instance, generating perturbations for an image of 128x128 using Bilinear Interpolation takes about 10 seconds, while using Bicubic Interpolation takes more than 20 seconds. We believe that Algorithm 1 can be improved by optimizing the search for perturbations and the interpolation method.

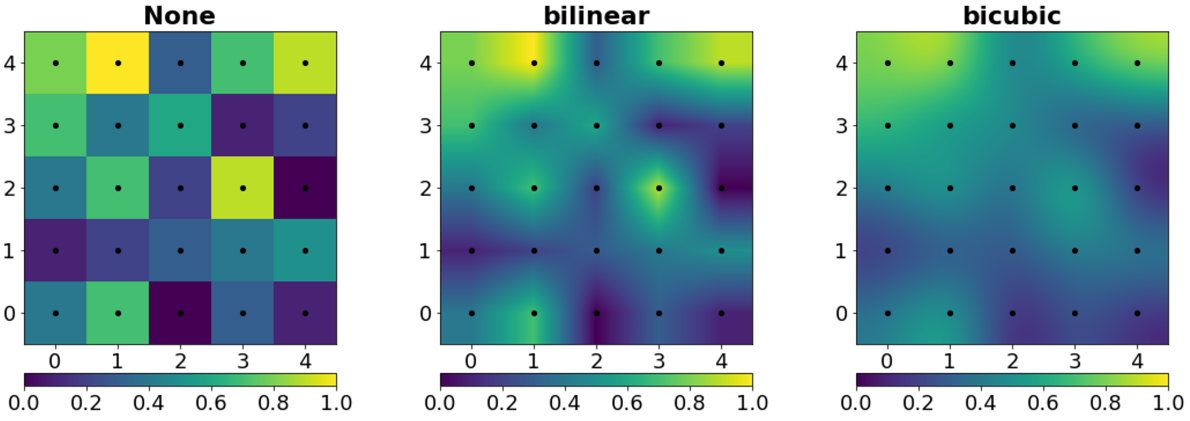

Figure 4.8: The original pixel image and the interpolated image.

4.6 Conclusions

This study is the first to use invisible perturbations in the digital domain to carry out attacks on MTCNN facial detectors and offers attack recommendations based on the features of MTCNN facial detectors. Additionally, we present image quality assessment baseline using PSNR, SSIM, and LPIPS on the CelebA dataset, which serves as a standard for future research in this field. This method falls under the category of white box attacks. Despite some limitations in the current approach, we plan to address these in future work by seeking ways to eliminate unwanted artifacts during perturbation interpolation and produce clearer images that preserve facial features. We also aim to apply existing methods to other face detection systems to uncover more general attack methods. Through this study, we hope to raise awareness of the potential benefits of adversarial attacks in protecting personal privacy and encourage more researchers to join this field of study.

5 A New Method of Disabling Face Detection by Drawing Lines between Eyes and Mouth

5.1 Introduction

As studied in Chapter 4, we found that in order to create a generic adversarial attack method to disable a face detector, the problem of feature changes under multi-scale images must be addressed. Although we have proposed an attack method through multi-scale perturbation fusion, there is still room for improvement in terms of clarity and robustness for different parameters of the image pyramid. If multi-scale iterative attacks are carried out simultaneously on MTCNN, SSD, and S3FD, the amount of perturbation produced will increase significantly, and the attack will be difficult to succeed. Additionally, the attack scope changes with the returned bounding box, and some of the perturbations are added to the background part of the image. To confine the perturbation to the face area and attack three face detectors, we change our approach and focus on attacking their shared operational features instead of optimizing the multi-scale perturbation fusion algorithm 1.

When face detectors detect an image, the facial features in the image often provide a large amount of feature information, playing a decisive role in the detection results. As shown in

Figure 5.1, we masked the facial features in the image and then performed face detection using MTCNN, SSD, and S3FD. Although the three face detectors focus on slightly different regions, they can still detect faces with covered facial features. For example, S3FD, which focuses on detecting small faces, returns multiple bounding boxes based on the obstruction. However, covering the regions between facial features with black lines may cause the face detector to fail. Even if we lower the confidence of SSD with the expectation of returning more detection results, it still cannot detect faces in the cover region. As shown in the fifth image of the second row in Figure 5.1, many bounding boxes have already been returned in the edge part, but the face still cannot be detected.

Figure 5.1: Blocking part of the facial area and performing face detection through face detectors SSD, S3FD, and MTCNN.

We have found that MTCNN, SSD, and S3FD are all based on CNN and use sliding anchor boxes to extract image features. Although covering facial features individually or multiple times can reduce the features extracted by the model, features from face other parts of the image still provide enough information for the model to make decisions. We believe that the black lines make the face detectors ineffective, likely because the black lines disrupt the continuity of features between various facial features, causing all three face detectors to fail. To validate the efficacy of the black line structure, we conducted experiments in this chapter. In addition, we use the complete coverage of the face by black lines as the baseline for image quality, and improve the quality of adversarial samples by optimizing the structure and adding random perturbations within the structure. Through a user demand questionnaire, we compare the acceptance of this method and baseline images to confirm whether our method can help users in need.

5.2 Black Line Structure Experiment

5.2.1 Dataset

In this study, we have selected the CelebFaces Attribute (CelebA) dataset [30], and the Flickr Faces High Quality (FFHQ) dataset [19]. The CelebA dataset is publicly available from The Chinese University of Hong Kong and is widely used for computer vision tasks related to faces. We have extracted 10,000 images from the 202,599 images available for this study. The FFHQ dataset was created as a benchmark for GANs and was open-sourced by NVIDIA in 2019. It is a high-quality facial dataset. We have randomly selected 6000 images from the 70,000 images available for this experiment. Both datasets are single-person datasets with a large proportion of face size, making the task of attack more challenging than for small facial images. In this way, we have a real-world multi-scenario facial dataset and a GAN-augmented dataset to ensure the generalizability of this experiment.

5.2.2 Method

Among the three networks of MTCNN, the threshold is 0.6 for P-Net, 0.6 for R-Net, and 0.7 for O-Net. The minimum size of the image pyramid is 21 pixels, with a scaling factor of 0.709. The threshold for SSD and S3FD is 0.6. These parameters are verified in a large number of experiments.

As shown in Figure 5.2(a), several simple hand-drawn black lines can cause three face detection algorithms to fail. There are many such images, but the positions and thicknesses of the black lines in each image are different. Therefore, we propose a unified method for adding black lines, as shown in Figure 5.2(b). The design method of black line structure is not unique, but rather a unified structure designed for quantitative experiments. The design concept of using straight lines was determined based on the process of feature extraction by CNN, with the objective of introducing perturbation in both horizontal and vertical directions. Curved or circular structures

do not conform to the way features are extracted by CNNs, while straight line structures are more conducive to verifying experimental results.

Figure 5.2: Image (a) is a manually added black line. Image (b) is a presentation image of the black line structure. Both images are from the FFHQ dataset.

In this study, we use MTCNN to detect the coordinates of bounding boxes and facial keypoints and connect these coordinates with black lines using the line method of OpenCV. The upper-left vertex of the bounding box is connected to the left eye, the upper-right vertex to the right eye, the lower-left vertex to the left corner of the mouth, and the lower-right vertex to the right corner of the mouth, ultimately forming a rectangle connecting the eyes and mouth. Due to the size differences between the CelebA and FFHQ datasets, the widths of the black lines added to the CelebA dataset were 6 pixels, 8 pixels, and 10 pixels, whereas the widths of the black lines added to the FFHQ dataset were 4 pixels 6 pixels and 8 pixels. In regard to the detection results, we need to prevent images from reaching the manual screening process of Faceswap. Therefore the facial region should be defined based on the boundary box detected by the facial detector, regardless of whether it is a frontal face, profile, or an erroneous detection. Black line structures will be generated within this scope. An attack is deemed successful only when the boundary box is not detected.

5.2.3 Result

The results of the experiment are presented in Tables 5.1 and 5.2 and Figure 5.3. The benchmark for the face detection accuracy is whether the face detector returns a bounding box. Even if the returned bounding box is not accurate, the attack is considered to have failed. This is because the Faceswap program provides a manual calibration step; as long as a bounding box is returned, it will be provided to this step. The tables and images show the probability of detecting a face in the image.

Table 5.1: The table shows the detection capabilities of the three face detectors on the original images in the CelebA dataset and on images with black line structures of 6-pixel, 8-pixel, and 10-pixel widths added.

CelebA	Original	6 Pixels	8 Pixels	10 Pixels
S3FD	99.67%	65.5%	49.4%	32.1%
SSD	99.71%	24.9%	15.08%	9.47%
MTCNN	99.79%	9.08%	7.49%	6.15%

Table 5.2: The table shows the detection capabilities of the three face detectors on the original images in the FFHQ dataset and on images with black line structures of 4-pixel, 6-pixel, and 8-pixel widths added.

FFHQ	Original	4 Pixels	6 Pixels	8 Pixels
S3FD	99.7%	58.7%	23.7%	9%
SSD	99.93%	22.4%	9.7%	4.3%
MTCNN	99.96%	11.7%	8.8%	7.2%

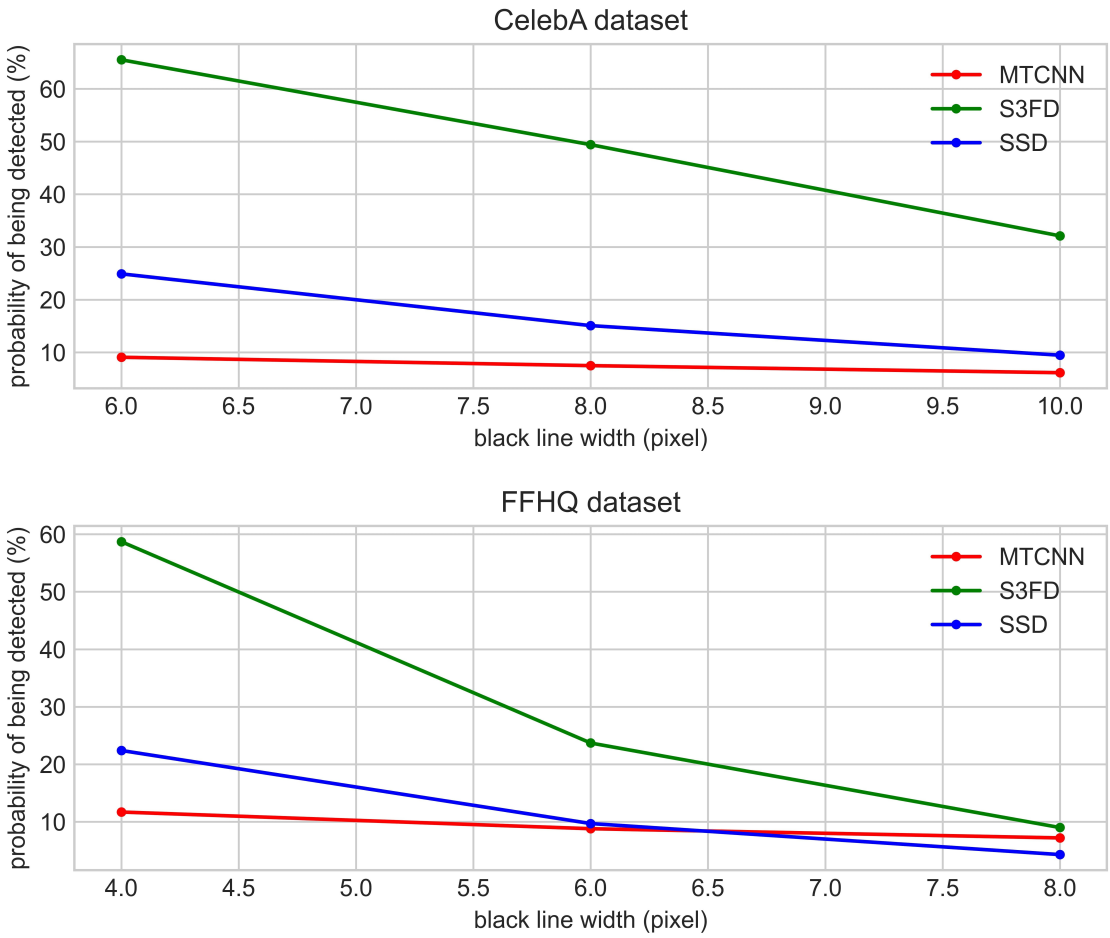

Figure 5.3: Different unit pixel widths are used due to the other face sizes in the CelebA dataset and the FFHQ dataset. The table shows the probability of the black line image being detected by MTCNN, S3FD, and SSD.

5.2.4 Discussion

The experiment shows that the black line structure can render MTCNN, SSD, and S3FD ineffective, with failure rates ranging from 34.5% to 95.7%. Even when bounding boxes are generated, it is still difficult to detect a complete face, as shown in Figure 5.4.

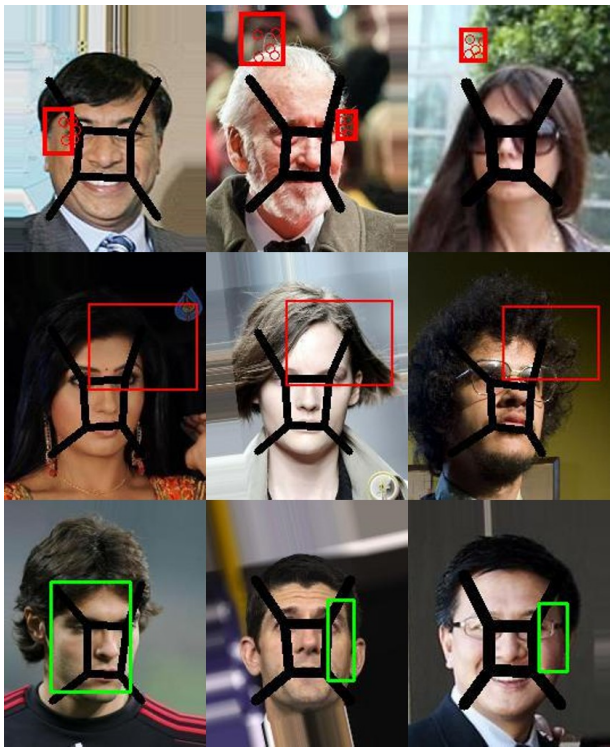

Figure 5.4: The image above shows some images with black line structure added but which can be detected by the face detector.

We use iNNvestigate [2] to perform an interpretability analysis of neural networks on images with black-lined structures. Neural networks work in an end-to-end manner and can achieve the desired results without understanding their complex internal workings. However, in order to make better use of neural networks, researchers have begun exploring methods to explain the internal structure of neural networks [3, 18, 53, 54]. These methods help us understand which pixels in the image affect the model's decisions. The iNNvestigate project integrates 13 neural network visualization methods, which are used in current works to visualize our results and

perform comparative analysis accurately.

In Figures 5.5–5.10, images of successful and unsuccessful attacks under different black line widths are compared. The first to third images in the first column on the left are images that the face detector is unable to detect, while the fourth to sixth images are images that the face detector can detect. It can be observed that effective black line structures are more visually apparent in the visualization and can perform face segmentation. The ineffective black line structures in the visualization are relatively blurry and have a high degree of fusion with the face. Therefore, it can be confirmed that when face features are blocked, the features read by the model cannot be associated with the information of face key points, leading to the failure of the face detector. It also demonstrates that breaking the continuity of features between facial key points can render the face detector ineffective.

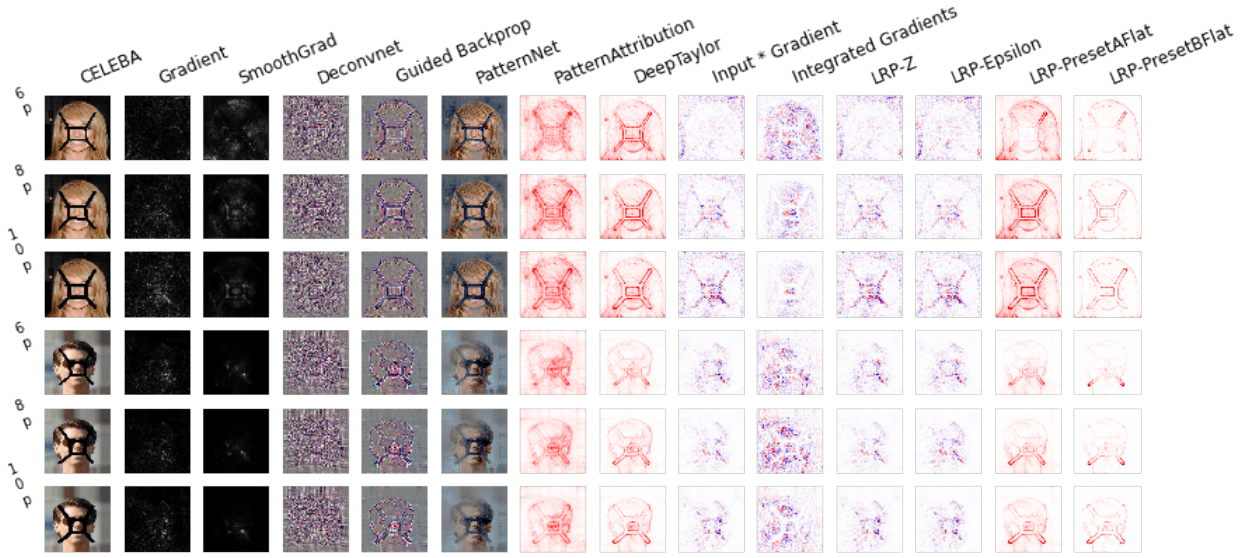

Figure 5.5: We selected two images from the CelebA dataset and added a black line structure with widths of 6 pixels, 8 pixels, and 10 pixels. MTCNN cannot detect the first three pictures, and the last three pictures are pictures that MTCNN can detect.

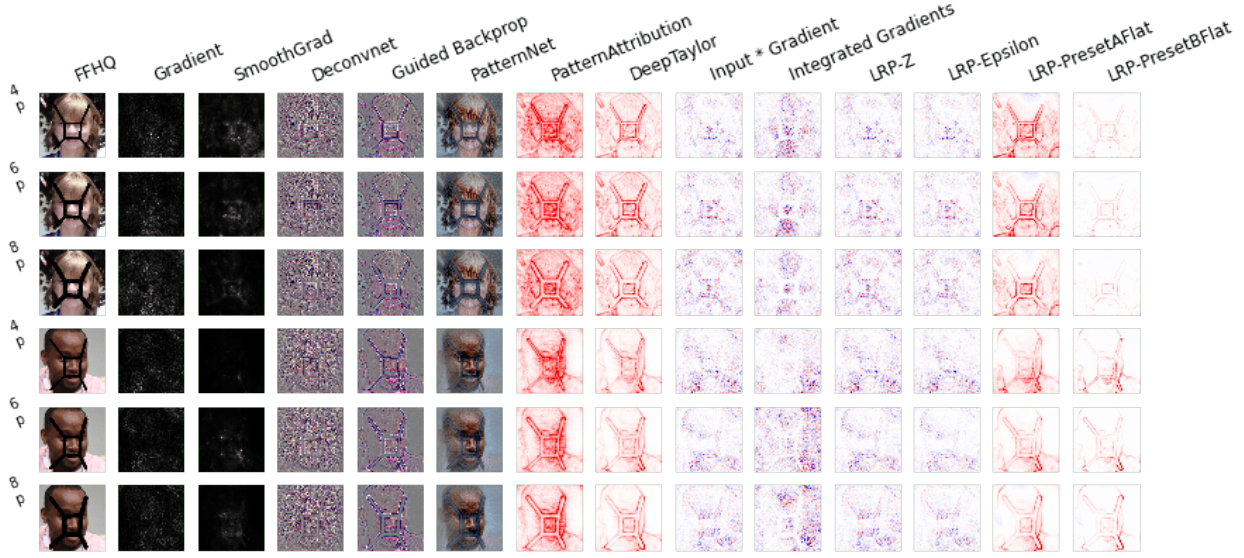

Figure 5.6: We selected two images from the FFHQ dataset and added a black line structure with widths of 4 pixels, 6 pixels, and 8 pixels. mtcnn cannot detect the first three pictures, and the last three pictures are pictures that mtcnn can detect.

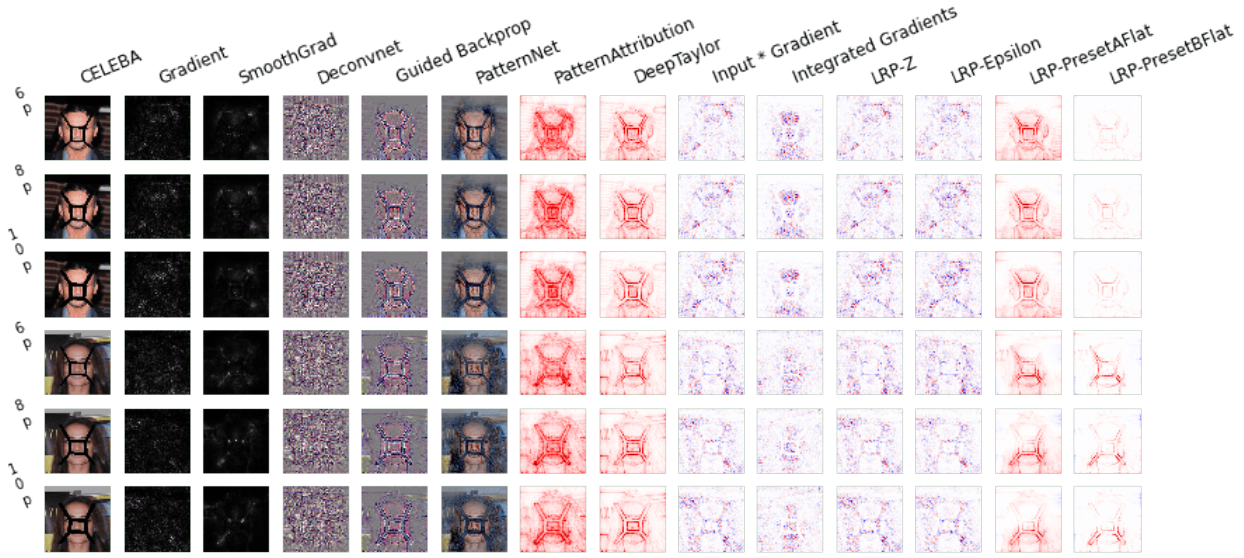

Figure 5.7: We selected two images from the CelebA dataset and added a black line structure with widths of 6 pixels, 8 pixels, and 10 pixels. S3FD cannot detect the first three pictures, and the last three pictures are pictures that S3FD can detect.

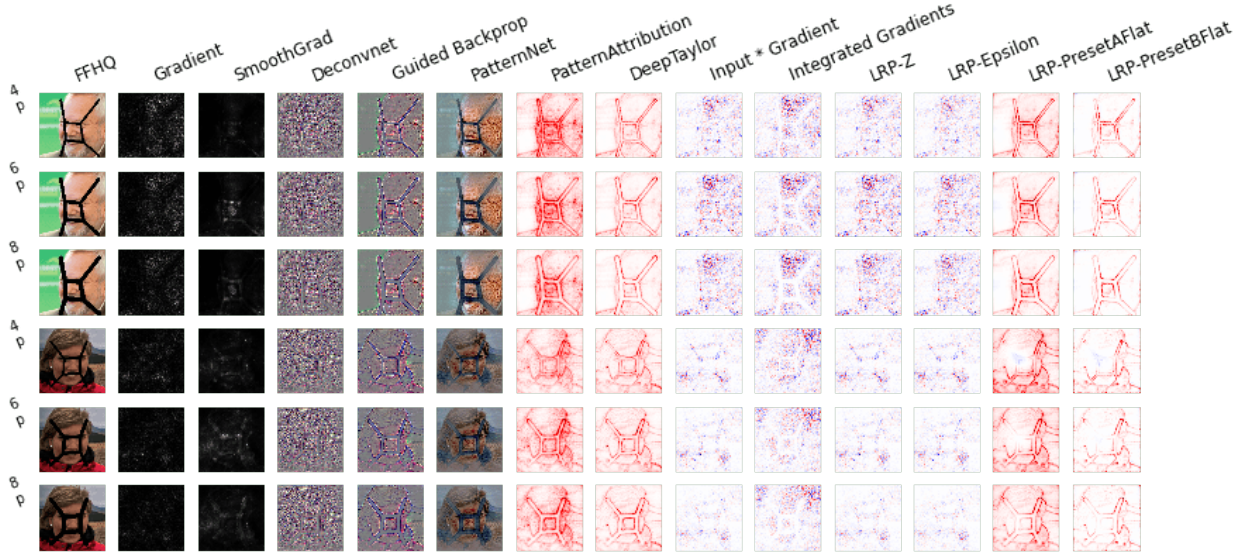

Figure 5.8: We selected two images from the FFHQ dataset and added a black line structure with widths of 4 pixels, 6 pixels, and 8 pixels. S3FD cannot detect the first three pictures, and the last three pictures are pictures that S3FD can detect.

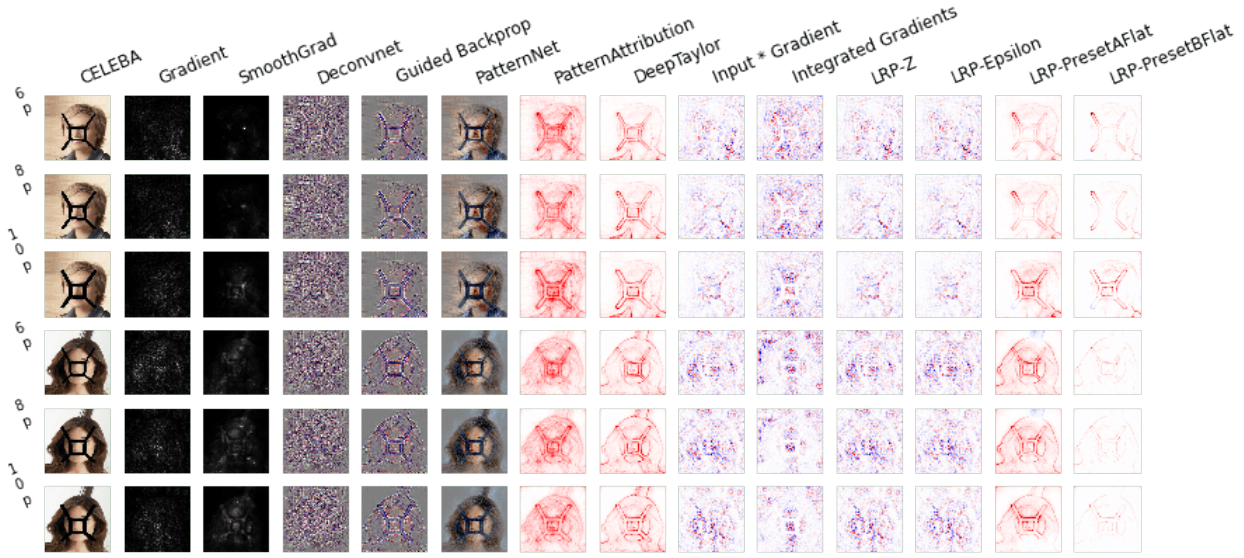

Figure 5.9: We selected two images from the CelebA dataset and added a black line structure with widths of 6 pixels, 8 pixels, and 10 pixels. SSD cannot detect the first three pictures, and the last three pictures are pictures that SSD can detect.

Figure 5.10: We selected two images from the FFHQ dataset and added a black line structure with widths of 4 pixels, 6 pixels, and 8 pixels. SSD cannot detect the first three pictures, and the last three pictures are pictures that SSD can detect.

5.3 Structure Optimization Experiment

The black line structure is an artificially designed attack structure. Its effectiveness has been confirmed in the aforementioned experiments, and its reasons for effectiveness have been analyzed. However, adding black line structures as disturbances in facial images can lead to a decrease in image usability. To test if the black line structure can be optimized artificially, I conducted the experiments in this section. The data and evaluation criteria used in this experiment are the same as those in the black line structure experiment.

5.3.1 Method

The experiments in this section adjusted the number, length, and width of the line segments in the black line structure, respectively, and the results are shown in Figure 5.11.

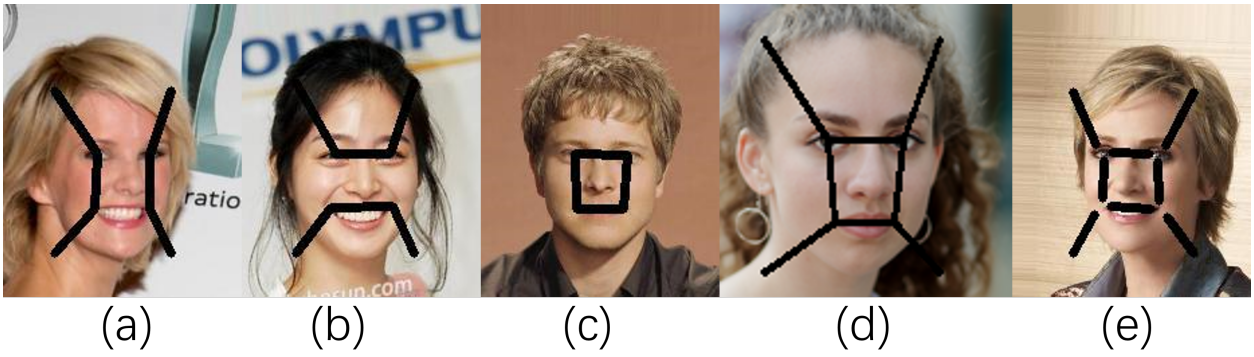

Figure 5.11: An example diagram of an optimized structure.

Reduce the number of line segments. In order to verify if the line segments can be reduced in the black line structure, three experimental designs for the structure are carried out in this section.

1. Eliminate the line segment between the eyes and the line segment between the corners of the mouth, the face is divided vertically into three regions as illustrated in Figure 5.11 (a).

2. Eliminate the line segment between the eyes and the corners of the mouth, and divide the

face horizontally into three regions, as depicted in Figure 5.11 (b).

3. Eliminate the line segments connecting the vertices of the bounding box and the corners of the mouth and eyes, and divide the face into two regions, inside and outside the facial features, as illustrated in Figure 5.11 (c).

This design was chosen because preserving some line segments has more experimental value for the line segments to segment the face effectively, rather than removing them arbitrarily. In this experiment, three structures were evaluated using 10,000 images from the CelebA dataset and line segment widths of 6, 8, and 10 pixels, and 6,000 images from the FFHQ dataset with line segment widths of 4, 6, and 8 pixels, respectively.

Reduce the width of line segments. To prevent the convolution kernels from crossing the black line during the feature extraction process and to ensure the ability to interrupt the feature continuity, the minimum width of the line segment is set to 4 pixels in the black line structure experiment. The experiment in this section tests the effect of reducing the width of the line segment from 4 pixels to 2 pixels while maintaining the same structure, as illustrated in Figure 5.11 (d). The experiment is conducted using 10,000 images from CelebA and 6,000 images from FFHQ, respectively, to verify the feasibility of having a thinner line segment. The convolution kernels in MTCNN, SSD, and S3FD have a general size of 3x3 pixels.

Reduce the length of the line segment. Finally, in order to expose keypoints like eyes while reducing the coverage of the black lines on the face, the ends of each line segment were shortened by 6 pixels, as shown in Figure 5.11 (e). The experiment was conducted using 10,000 images from CelebA and 6,000 images from FFHQ, with line segment widths of 6, 8, and 10 pixels for CelebA, and 4, 6, and 8 pixels for FFHQ, respectively.

5.3.2 Result

Tables 5.3,Table 5.4, and Table 5.5 present the experimental results of reducing the number of line segments and are displayed as graphs inFigure 5.12, Figure 5.13, and Figure 5.14. The exper-

iment employs line segment widths of 6, 8, and 10 pixels in the CelebA data, and line segment widths of 4, 6, and 8 pixels in the FFHQ data. The "original structure" represents the full black line structure, while "remove horizontal lines" represents the removal of the line segment between the eyes and the line segment between the corners of the mouth. "Remove vertical lines" represents the elimination of the line segment between the eyes and the corner of the mouth, and "remove oblique lines" signifies the elimination of the line segments between the vertices of the bounding box and the corners of the mouth and eyes.

Table 5.3: The values in the table are the detection results that the MTCNN face detector can detect the face.

MTCNN			
CelebA	**6 Pixels**	**8 Pixels**	**10 Pixels**
original structure	9.08%	7.49%	6.15%
remove horizontal lines	33.23%	23.63%	17.52%
remove vertical lines	34.34%	25.79%	18.85%
remove oblique lines	22.63%	17.05%	13.82%
FFHQ	**4 Pixels**	**6 Pixels**	**8 Pixels**
original structure	11.70%	8.80%	7.20%
remove horizontal lines	40.75%	26.73%	18.67%
remove vertical lines	45.35%	33.02%	22.83%
remove oblique lines	29.03%	19.60%	15.37%

Table 5.4: The values in the table are the detection results that the SSD face detector can detect the face.

SSD			
CelebA	**6 Pixels**	**8 Pixels**	**10 Pixels**
original structure	24.90%	15.08%	9.47%
remove horizontal lines	79.84%	67.10%	52.17%
remove vertical lines	88.61%	83.43%	76.82%
remove oblique lines	96.83%	95.50%	93.08%
FFHQ	**4 Pixels**	**6 Pixels**	**8 Pixels**
original structure	22.40%	9.70%	4.30%
remove horizontal lines	75.65%	59.08%	41.88%
remove vertical lines	93.90%	85.83%	76.12%
remove oblique lines	98.45%	97.18%	95.82%

Table 5.5: The values in the table are the detection results that the S3FD face detector can detect the face.

S3FD			
CelebA	**6 Pixels**	**8 Pixels**	**10 Pixels**
original structure	65.50%	49.40%	32.10%
remove horizontal lines	89.59%	83.42%	73.27%
remove vertical lines	91.98%	85.50%	76.10%
remove oblique lines	94.95%	91.75%	86.46%
FFHQ	**4 Pixels**	**6 Pixels**	**8 Pixels**
original structure	58.70%	23.70%	9.00%
remove horizontal lines	84.80%	67.18%	50.25%
remove vertical lines	88.03%	67.67%	48.45%
remove oblique lines	89.98%	71.78%	54.67%

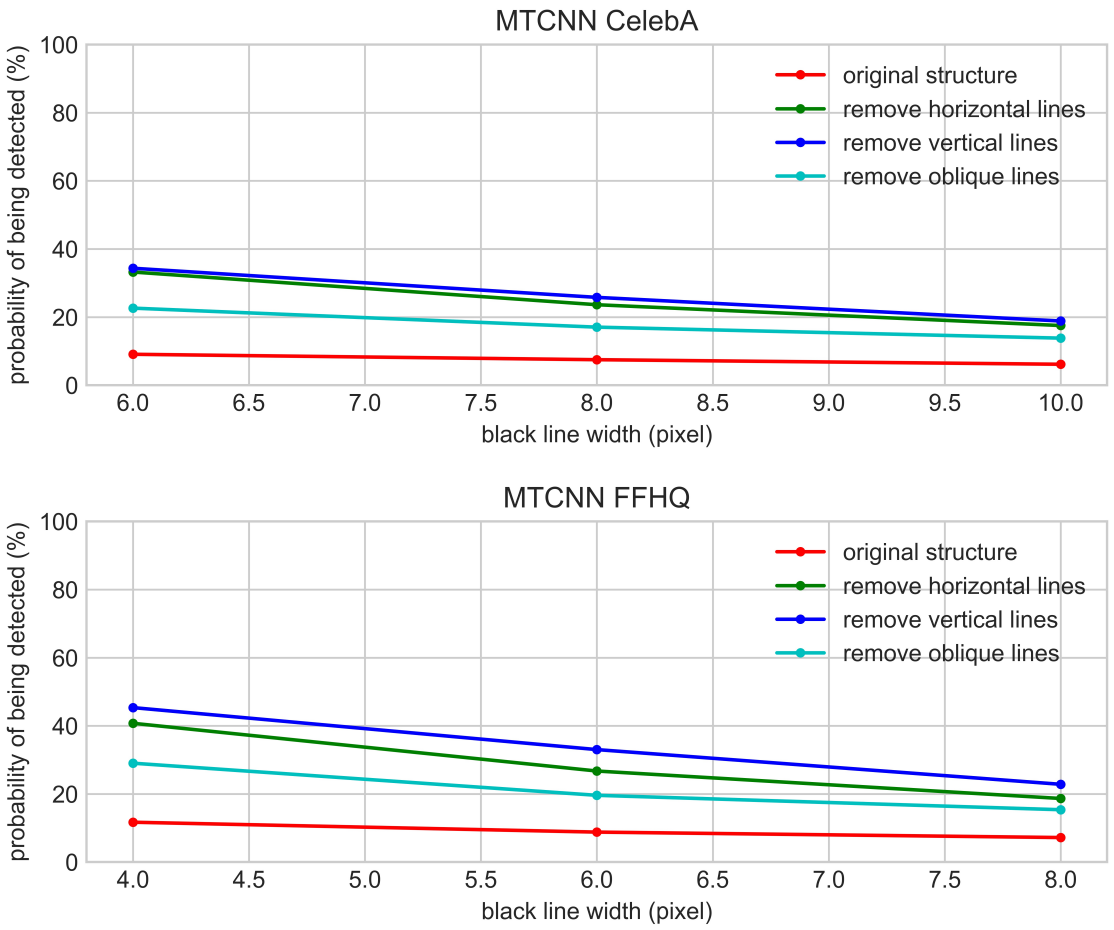

Figure 5.12: The figure shows the resulting curve that the MTCNN face detector can detect the face.

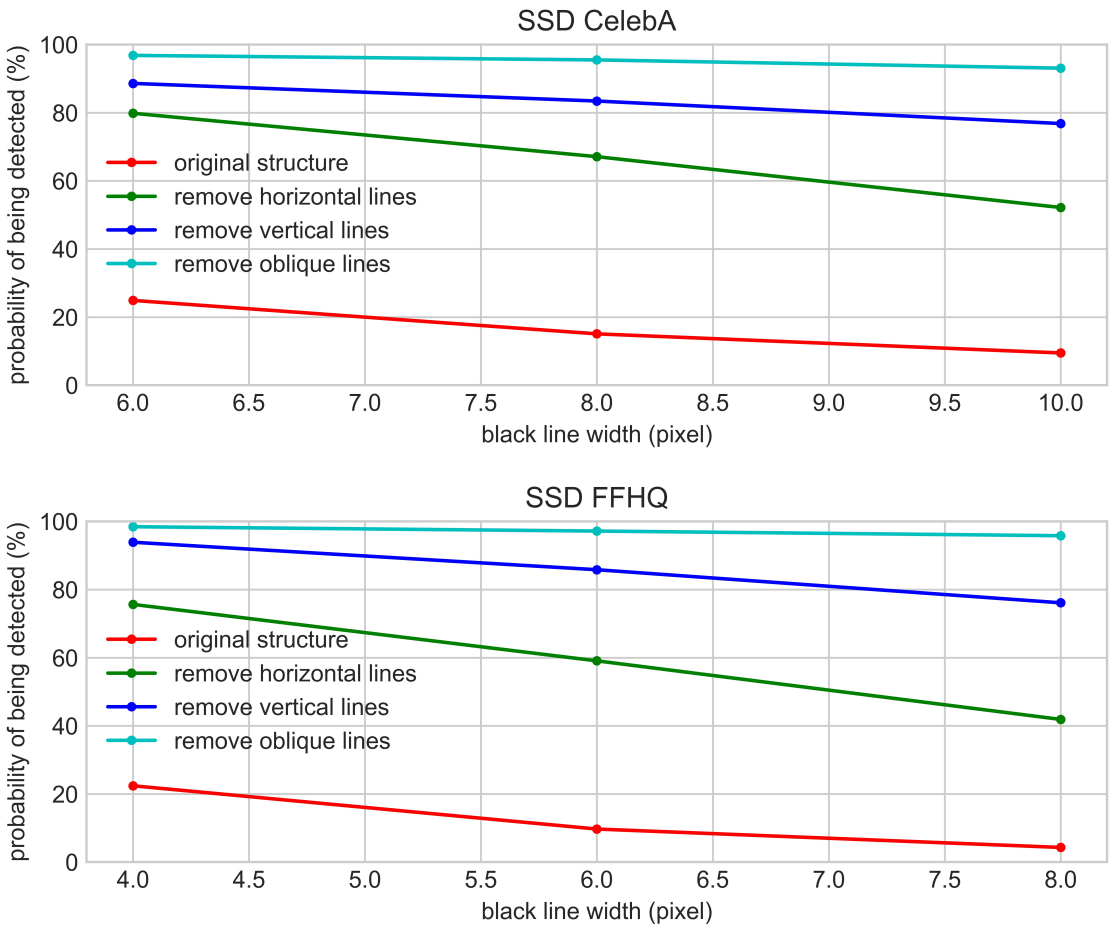

Figure 5.13: The figure shows the resulting curve that the SSD face detector can detect the face.

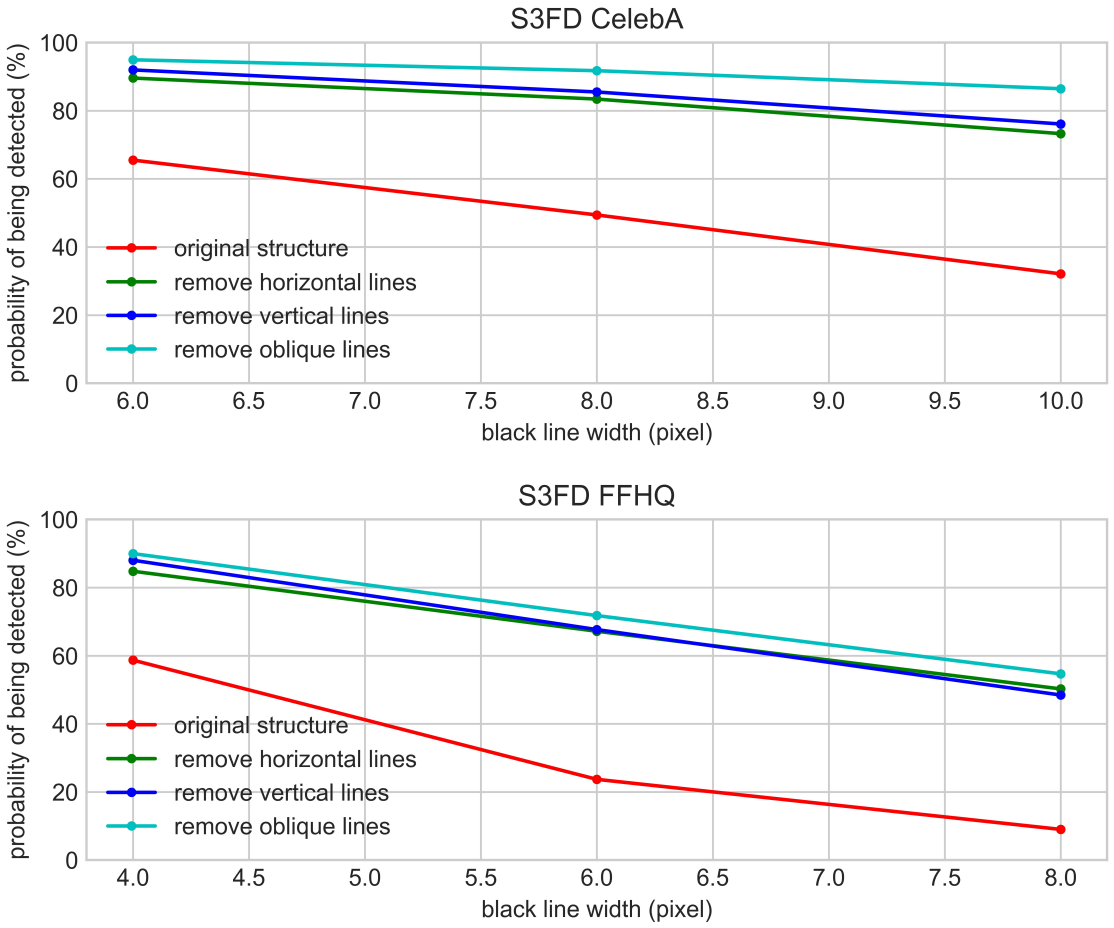

Figure 5.14: The figure shows the resulting curve that the S3FD face detector can detect the face.

Table 5.6 presents the experimental results of reducing line segment width, shown in Figure 5.15 as a graph.

Table 5.6: The values in the table are the detection results that MTCNN, SSD, and S3FD can detect faces when the line segment width is 2 pixels.

2 Pixels	MTCNN	SSD	S3FD
CelebA	53.13 %	63.80%	94.85%
FFHQ	34.15%	52.35%	91.17%

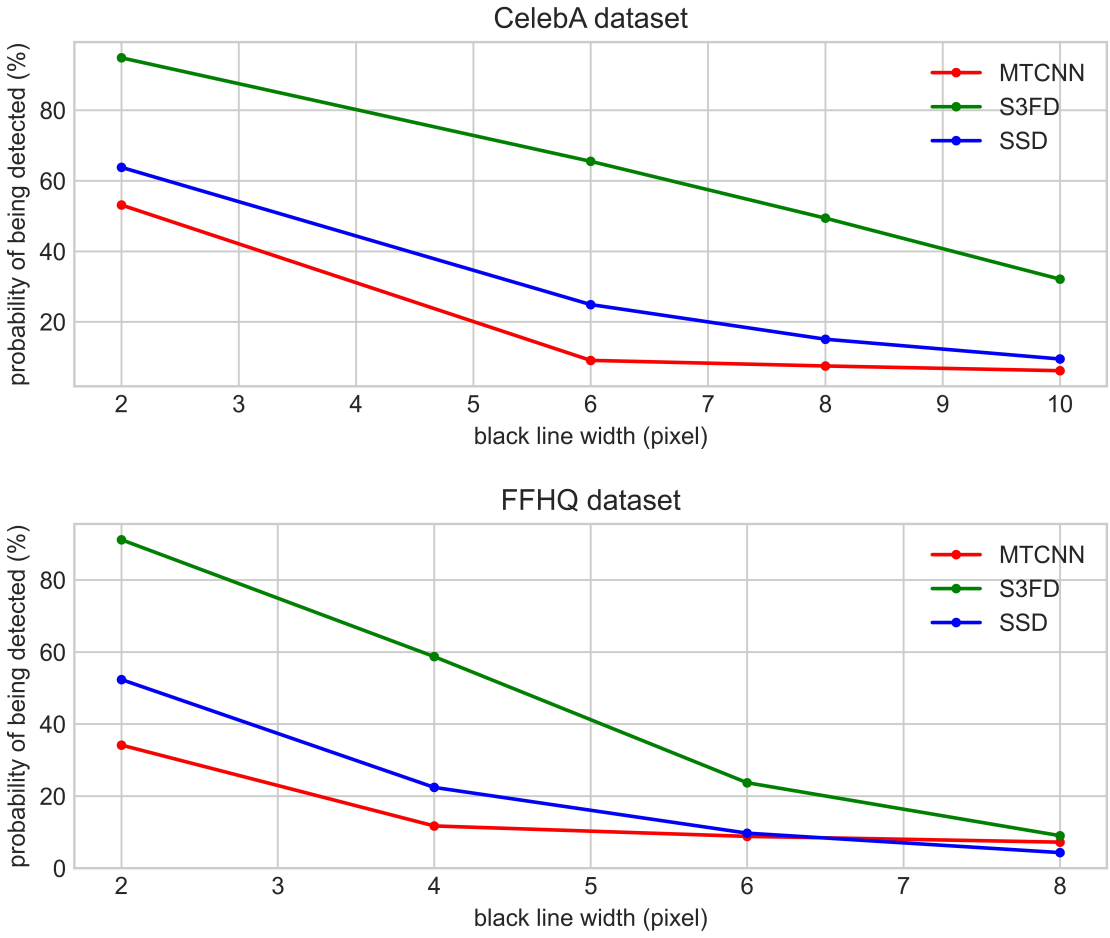

Figure 5.15: The figure shows the detection results of MTCNN, SSD, and S3FD face detectors at various widths of line segments.

Table 5.7 presents the experimental results of reducing the length of line segments, which are displayed in graphs in Figure 5.16, Figure 5.17, and Figure 5.18. The "short line structure" refers to the shortened black line structure.

Table 5.7: The table below shows the detection capabilities of three face detectors for short line structures.

CelebA	**6 Pixels**	**8 Pixels**	**10 Pixels**
MTCNN	14.88%	12.83%	10.62%
SSD	40.22%	34.51%	27.51%
S3FD	77.94%	69.73%	57.07%
FFHQ	**4 Pixels**	**6 Pixels**	**8 Pixels**
MTCNN	17.32%	12.88%	10.47%
SSD	34.35%	14.88%	8.78%
S3FD	73.18%	33.42%	17.93%

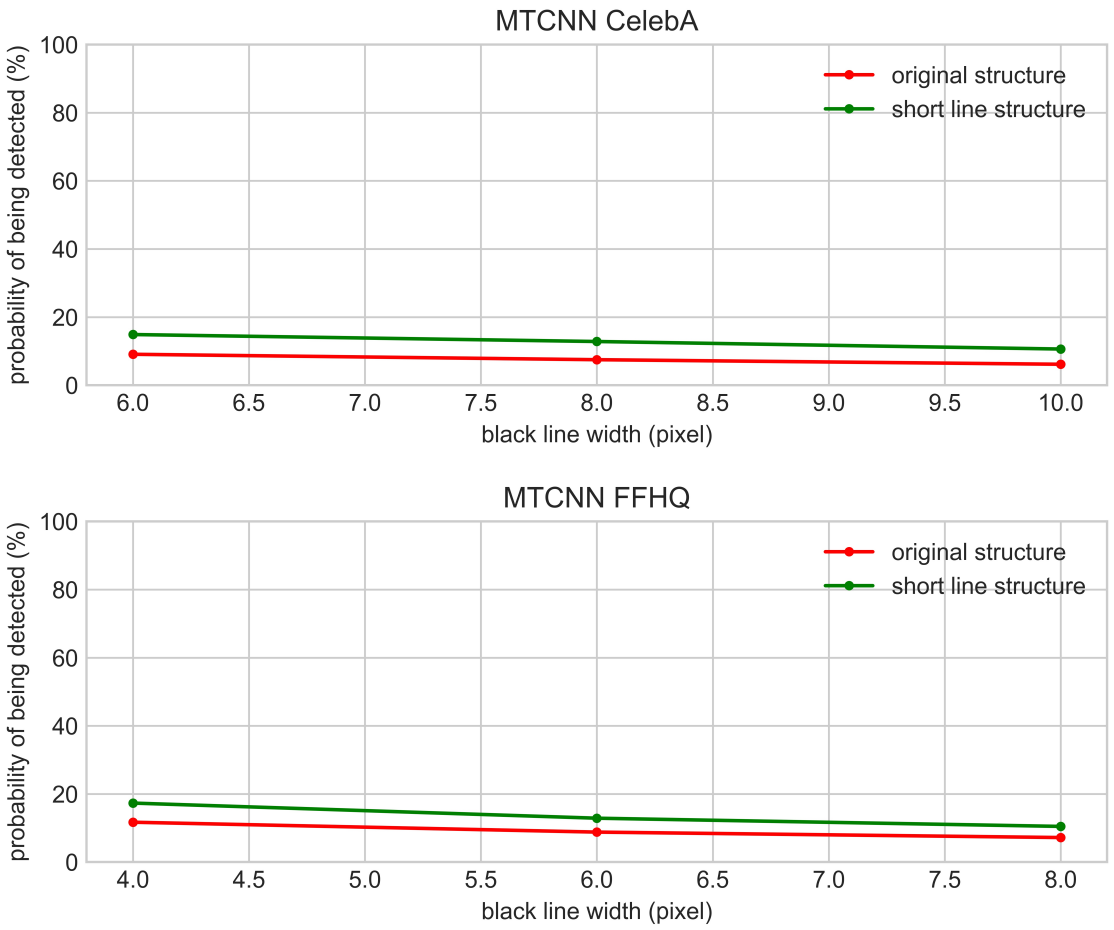

Figure 5.16: Comparison of the detection results of the short line structure and the complete black line structure detected by MTCNN.

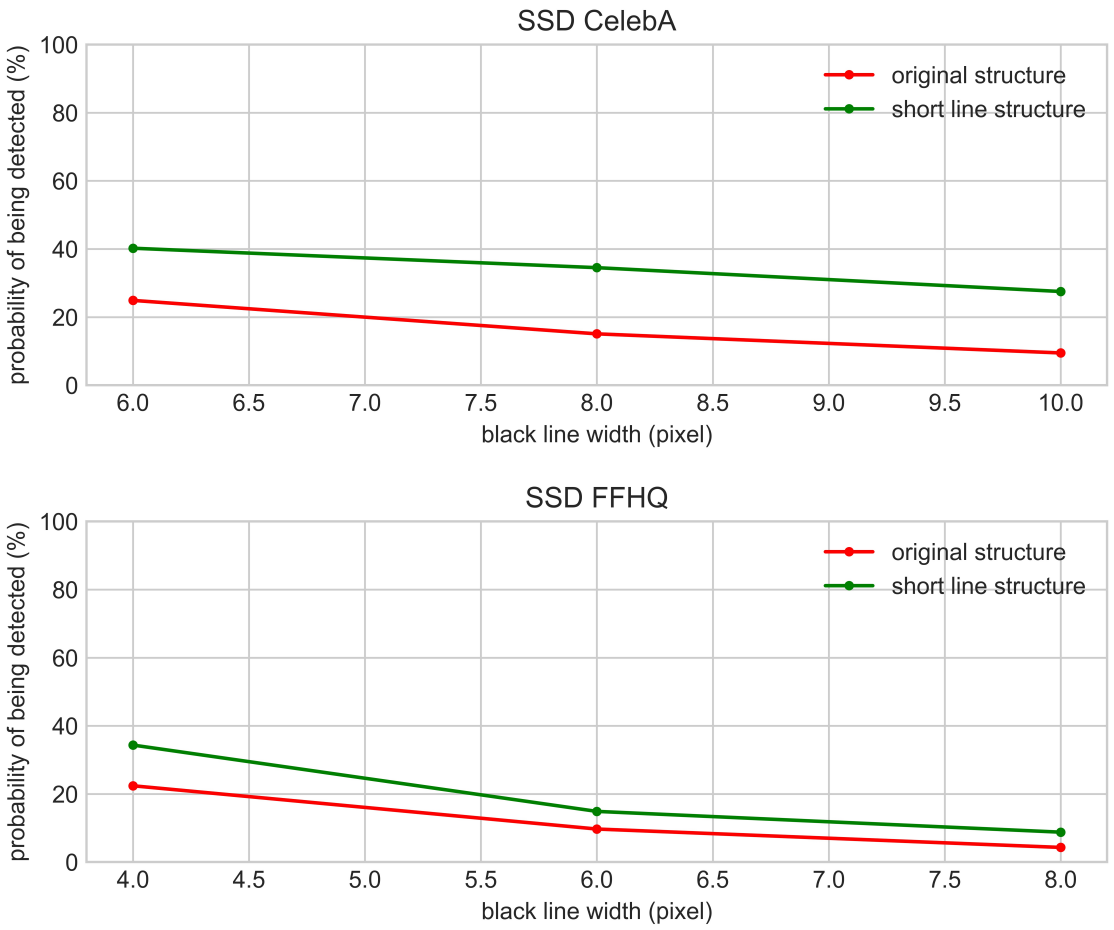

Figure 5.17: Comparison of the detection results of the short line structure and the complete black line structure detected by SSD.

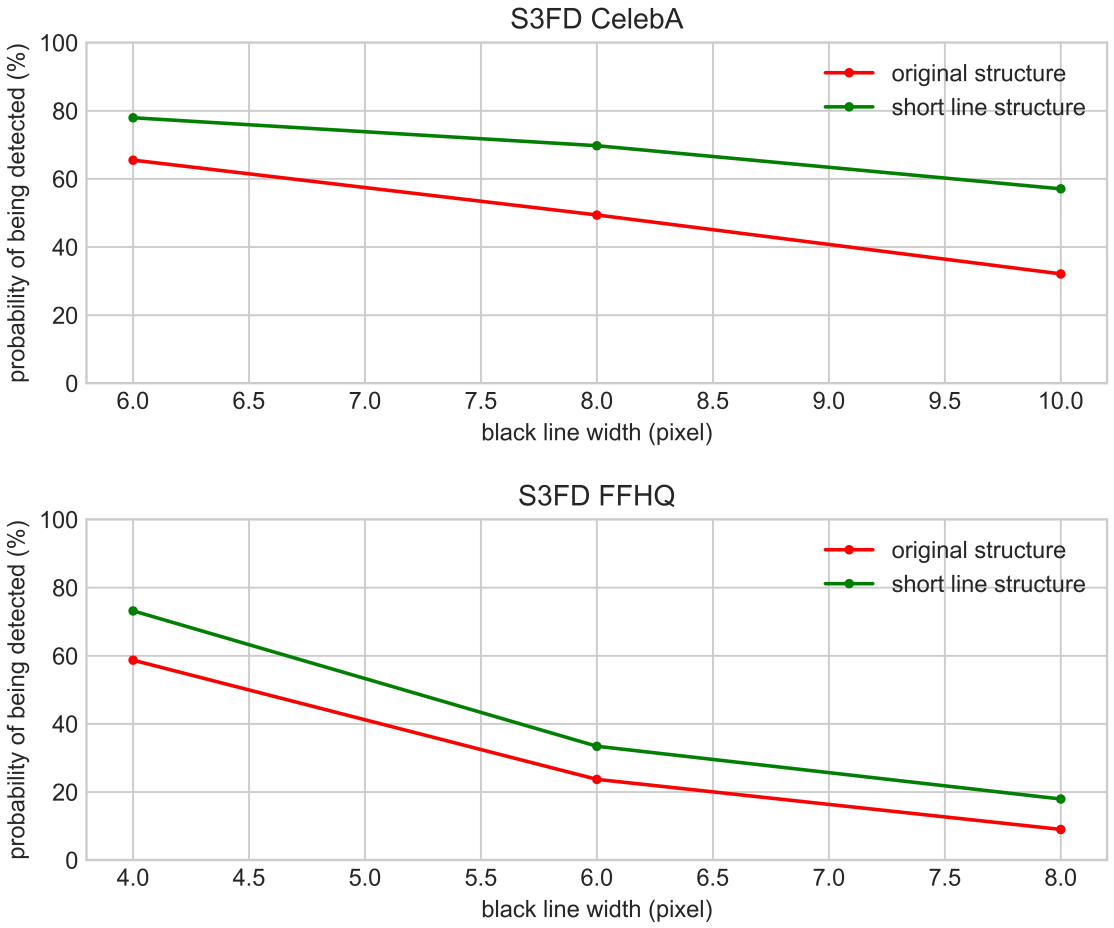

Figure 5.18: Comparison of the detection results of the short line structure and the complete black line structure detected by S3FD.

5.3.3 Discussion

The experiments have revealed that modifying the number of line segments significantly reduces the probability of a successful attack. In addition, changing the structure while maintaining the number of line segments may also lead to a decrease in the effectiveness of attacks. For example, changing four line segments that extend outward to those that extend inward does not block feature extraction from the exterior of the quadrilateral, and the attack capability can be referred to the rectangular structure. Therefore, optimizing the black line structure should not start from optimizing its structure.

As illustrated in Figure 5.15, the attack efficiency of the 2-pixel-width black line structure is weaker compared to other widths, and this difference does not vary proportionally. This outcome aligns with the initial design consideration of the line segment width. The black line structure aims to disrupt the feature extraction process, and the size of the convolution kernel plays a crucial role in feature extraction. When the width of the line segment is smaller than the size of the convolution kernel, the probability of the convolution kernel crossing the line segment increases, thereby reducing the attack efficiency. It can be observed that reducing the width of the line segment cannot optimize the black line structure. From this, it can also be deduced that when modifying the edges of the black line using facial pixels near the black line structure, the attacking capability of the black line structure may decrease as the line segment becomes narrower and facial features increase.

It can be seen from Figure 5.16, Figure 5.17, and Figure 5.18 that as the width of the line segment increases, the attack capability will also increase, and the trend of change is consistent with that of the full black line structure. Despite exposing key points such as eyes, the attack capability has not decreased significantly, demonstrating that the black line structure is an attack method that disrupts feature extraction rather than simply covering up the perturbation. Although, in comparison to the other two optimization experiments, reducing the length of the line segment

can optimize the black line structure to some extent. However, it is not considered a reasonable optimization as there will still be visible line segments on the face, reducing its usability. Moreover, the deleted region still contains valid perturbations.

The above three experiments have been conducted through manual screening to optimize the black line structure. It is evident that, while maintaining the structure unchanged, adjusting the length of the line segment can achieve optimization to a certain extent. However, this approach also has its limitations as it is not possible to try all possible experiments.

5.4 Image Quality Optimization Experiment

In order to increase the usability of images, we conducted an image quality optimization experiment. The black line structure generates black lines in the image to attack the facial detector. Although complete coverage of the face with black lines can render the facial detector ineffective, it also lowers the usability of the image to the minimum. As shown in (a) of Figure 5.19. Therefore, in this section's experiment, complete coverage of the face by black lines was taken as the balseline for image quality, and the coverage was reduced from this baseline to improve the usability of the image.

In this experiment, 3000 images from CelebA and FFHQ are used as evaluation data with black line widths of 10 pixels and 8 pixels, respectively. The image quality scores are calculated using PSNR, SSIM, and LPIPS. When evaluating two completely identical images, the score for PSNR is infinity, the score for SSIM is 1, and the score for LPIPS is 0. Table 5.8 presents the baseline for image quality, as well as the image quality scores of the black line structure. Based on the comparison with the original images, we consider that the image quality scores must be higher than 11.03 and 9.24 in PSNR evaluation, lower than 0.34 and 0.56 in SSIM evaluation, and higher than 0.67 and 0.49 in LPIPS evaluation. Although the image quality with the black line structure has exceeded the baseline, there may still be deficiencies in terms of usability. Therefore, we

addressed this issue from two aspects. Firstly, we optimized the image quality further to achieve a higher quality score than the baseline and the image quality score with the black line structure. Secondly, we conducted a questionnaire survey on user requirements to confirm the usability of this method. To achieve the goal of assisting users with image privacy and security needs, the participants who select optimized images should be more than those who choose images with the entire face covered by black lines.

Table 5.8: The "Original" row displays the image quality evaluation score for the unaltered image. "Baseline" show the image quality benchmarks for the CelebA and FFHQ datasets when the faces are fully obscured by black. Each column represents the score obtained from the relevant evaluation method. The "Black line structure" row, "Random perturbation" row, and "Genetic Algorithm" row respectively show their image quality scores in the corresponding dataset.

		PSNR	SSIM	LPIPS
Original	Original	Inf	1	0
Baseline	CelebA	11.03	0.66	0.33
	FFHQ	9.24	0.44	0.51
Black line structure	CelebA	16.42	0.82	0.14
	FFHQ	14.44	0.73	0.22
Random perturbation	CelebA	20.3	0.84	0.12
	FFHQ	17.8	0.74	0.18
Genetic Algorithm	FFHQ	18.17	0.75	0.17

5.4.1 Method and Experiment

In this experiment, 3000 images from CelebA and FFHQ datasets were used as experimental data, respectively. According to the short-line structure experiment, the ability to attack can be maintained to some extent when the eyes and mouth corners are not covered. This section of the experiment will not cover the eyes and mouth corners. This experiment still uses the boundary box coordinates, eye coordinates, and mouth coordinates detected by MTCNN as the basis for computing the perturbation coverage range. This experiment uses the Bresenham algorithm to

calculate the accurate coordinates between two points [4]. The method of drawing lines used in this approach differs from the method used in the black line structure, resulting in slightly different line shapes. However, the coordinates and overall structure used in this method are identical to those used in the black line structure. We shorten the end of each line segment connecting the eyes or the corner of the mouth by 6 pixels. Then, we calculate all coordinate points between two points with a line segment width of 10 pixels and 8 pixels, and these coordinate points are the positions to be perturbed. The experiment used random search perturbation with values ranging from 0 to 255 and added the perturbed value to the original pixel value. The stopping condition is the successful attack on three facial detectors or the completion of the loop times. To reduce the coverage rate of perturbation, we control the amount of perturbation to be below 45%. The attack success rate of random perturbation adversarial samples on the MTCNN, SSD, and Sfd face detectors was 67.8%, 85.7%, and 69.5%, respectively, in the CelebA dataset, and 86.4%, 96%, and 72.7%, respectively, in the FFHQ dataset. The experimental example is shown in Figure 5.19(b), and the image quality results shown in the ”Random perturbation“ row of Table 5.8.

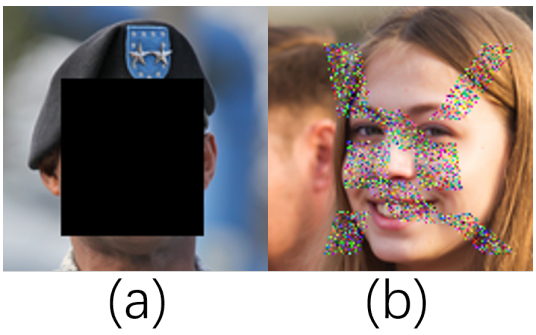

Figure 5.19: The (a) represents the images with the faces fully covered and the (b) represents the images with added random perturbation.

The image quality scores show that using random perturbations for optimization has improved image quality for adversarial examples in both datasets. However, the improvement in image quality in the FFHQ dataset is relatively limited. This is due to the small size of images in the FFHQ dataset and the large proportion of the face in the images. Therefore, the experiment

further optimized image quality in the FFHQ data using the Genetic Algorithm [16]. The Genetic Algorithm finds the optimal solution by fitting the target function through processes such as selection, crossover, and mutation. In the selection process, we discarded individuals ranking in the bottom 50%; In the crossover process, we employed the top 20 ranked individuals and randomly crossed them with other individuals; In the mutation process, we perturbed 1% pixels of each individual. This experiment focuses on improving image quality and designs a target function that meets the requirements. We use the pixel values in the original image on the coordinate as the target array and select individuals similar to the target array in the random perturbation population to rank higher. The experiment needs to balance the attack ability and image quality, and we also included a value for judging the attack result in the target function, as the Formula 5.1. Finally, the loop will terminate upon successful attack or completion.

$$fitness = -(a * IQ + b * BB) \tag{5.1}$$

In this experiment, IQ represents the difference between the individual and target array, or the image quality, while BB represents the number of returned bounding boxes. The cosine similarity [43] was chosen to calculate IQ because more than 55% of the positions in the individual array are 0. This makes PSNR too strict when calculating similarity, magnifying the difference between the individual and target array. However, these 0 values do not interfere with the image. The control parameters a and b are designed for balancing purposes, as sorting by size requires that the fitness value be assigned negative values. The performance results of the experiment on FFHQ are shown in the Table 5.8, and the image quality score exceeded random perturbations. The performance of attack success rates on MTCNN, SSD, and S3FD are 80.1%, 94.4%, and 75.3%, respectively. The optimized images are shown in Figure 5.20.

Figure 5.20: The images are from the FFHQ dataset, with the first row being randomly perturbed images and the second row being genetically perturbed images.

5.4.2 Discussion

In the image quality optimization experiment based on data from CelebA and FFHQ, the image quality scores in PSNR were 20.3 and 18.17, surpassing the baseline scores of 11.03 and 9.24. Compared to the original images, the scores of SSIM were 0.16 and 0.25, lower than the baseline scores of 0.34 and 0.56; the scores of LPIPS were 0.88 and 0.83, higher than the baseline scores of 0.67 and 0.49. The results demonstrate that, in the three image quality evaluation algorithms of PSNR, SSIM, and LPIPS, the optimized images are superior to the baseline, indicating a significant improvement in image quality.

We conducted a survey regarding image privacy and security in the form of an online questionnaire.The questionnaire first introduced the state of the art of AI face swapping technology [1]. The questionnaire surveyed the respondents' knowledge and usage of AI face-swapping technology, as well as their perceptions of potential risks.The survey also introduced the experimental content and asked whether the respondents would use our method to protect their images.The survey also focused on the respondents' professions to identify potential user groups based on occupational characteristics. A total of 73 participants took part in this questionnaire survey, including 15 police officials from the Public Security Bureau of Ning'an City, Heilongjiang Province, 10 healthcare workers from Ning'an City Chinese Medicine Hospital, 14 Public officials from the People's Government of Qiaodong District, Xingtai City, Hebei Province, 14 teachers and 20 stu-

[1] https://www.wjx.cn/vm/h47xeFk.aspx#

dents from Xingtai College, Hebei Province.

In the survey, 94.52% of the respondents expressed their unwillingness for their images to be used without authorization, indicating a significant demand for a solution that protects facial privacy. We also investigated the willingness for using adversarial samples in this study, and the results showed that no one would use a photo with their face completely covered by black lines. 6.85% of the respondents chose the black line structure, including three police officers, one teacher, and one student. We conjecture that an individual's willingness to use adversarial samples may be correlated with their occupation. In the case of adversarial samples with random perturbations, 34.25% of the respondents indicated that they accept this method of protecting privacy, an improvement of 27.4% compared to black line structures. As the image quality improves, an increasing number of respondents will be willing to protect their images. Figure 5.21 shows the proportion of respondents who selected different adversarial samples in various occupations. The proportions within different occupations, such as police officials, public officials, teachers, healthcare workers, and students, respectively increased by 46.67%, 42.86%, 28.57%, 20%, and 5%. Based on the above information, we have concluded that as the image quality improves, the number of users opting to protect their facial privacy will also increase. Moreover, for professions with high security requirements, such as police officers, the demand for photo privacy protection is higher, and when necessary, the requirements for image quality are low. Public officials facing the public also have a need for photo privacy and security, but they place more attention on the image quality of photos. We believe that since photos of these groups may be publicly displayed, such as on official websites or bulletin boards, they are more attentive to the possibility of their images being misused. According to the survey results and our analysis of potential user groups, we believe that there are additional professions beyond those covered in the questionnaire that may also have a need for image privacy and security. For example, ID photos of soldiers who are at war are likely to be maliciously face-swapping in order to defraud their relatives. In scenarios such as official websites, bulletin boards, or ID photos of soldiers, as long as a person can visually

identify who is in the photo and it cannot be maliciously exploited, it can meet the user's usage needs.

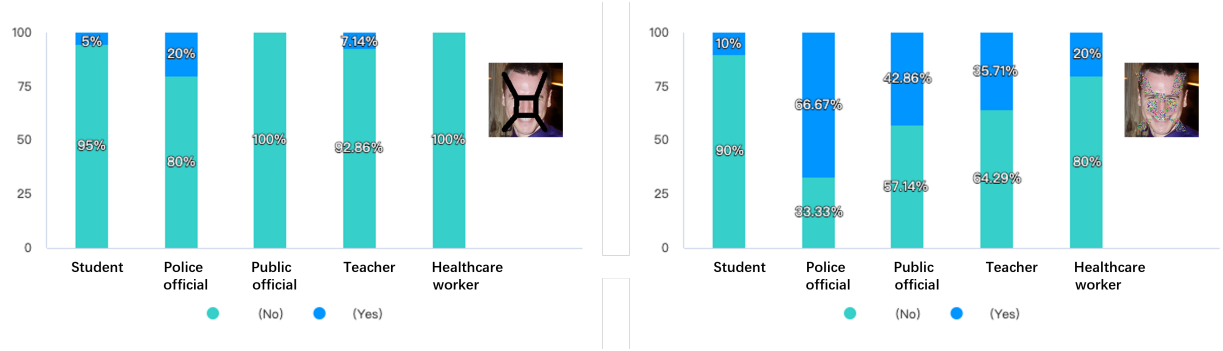

Figure 5.21: A comparative chart of different professions' acceptance levels of different adversarial samples was included in the user privacy requirements survey. The 'yes' represents the number of participants who chose the adversarial sample, while 'no' represents the number of participants who did not choose it, both displayed as a percentage.

In this experiment, a short-line structure was used as a reference, and the image quality was improved by using random perturbation and reducing the amount of coverage. The performance in the PSNR, SSIM and LPIPS algorithms was better than the baseline and also better than the image quality score of the black-line structure. An adaptation function combining the face pixels as the target and the attack success rate was proposed to improve the adversarial samples' quality score and achieve the study's desired objective. A user demand survey was conducted, and the results were analyzed for user groups with needs. It was proven that our method could be accepted to some extent by users, achieving the goal of helping users with privacy security needs.

5.5 Conclusions

This chapter presents a black line structure that disrupts the feature continuity, causing MTCNN, S3FD, and SSD to be ineffective. In order to enhance the usability of the adversarial samples, optimization experiments were also performed on its structure, and a short line structure was found

to have a smaller impact on the attack capability. Taking the short-line structure as a reference, further optimization was performed on the image quality of the adversarial samples, resulting in a score that surpasses the baseline score of the experiment and the image quality score of the black line structure. The questionnaire survey combined with user requirements demonstrated that our method can provide assistance to users with a need for image privacy and security.

6 Conclusions

Face detection is a significant research field in computer vision. Face detection is the basis of face recognition and provides ideas and innovations for developing target detection. In order to protect the privacy of personal photos from malicious face-swapping behaviors, this paper attacks MTCNN, SSD, and S3FD. In many kinds of research on adversarial attacks, people always worry that adversarial attacks will affect the judgment of artificial intelligence and lead to a series of security problems. However, the experiment in this paper does not aim to carry out destructive attacks, but rather plays a role in reverse protection. The study considers that technology is neutral and its use determines whether it is for good or evil purposes. The methods and objectives of the study are clear: to use simple solutions to address specific problems.

6.1 Chapters Summary

Chapter 1. At the beginning of this chapter, examples of malicious face-swapping behaviors and harms are first presented. The work goal of the full text is clarified, which is to hand over the autonomy of face-swapping to the owner of the image and prevent the occurrence of malicious face-swapping. And by sorting out the face-swapping technology and analyzing the popular face-swapping software Faceswap, specific targets for attacking MTCNN, SSD, and S3FD face detectors are proposed. And according to Faceswap's face-swapping process, we believe that deactivating the face detector is more effective and advanced among various methods of attacking face-swapping models. Finally, the related technologies of face detection are sorted out.

Chapter 2. This chapter introduces the techniques and related work involved in the experiments. It includes Convolutional Neural Networks, the network structures, workflows, and differences of MTCNN, SSD, and S3FD face detectors, as well as the principle of adversarial attack.

Chapter 3. In this chapter, based on previous research on attack methods, an attempt is made to determine the effective attack range in the image. Facial features are extracted through the use of style transfer technology and then merged with the background. The resulting merged image is then subjected to detection by MTCNN. The results show increasing the detection probability in the background using style transfer does not affect the detection results of the MTCNN facial detector for faces in the image.

Chapter 4. Unlike SSD and S3FD, MTCNN uses image pyramid processing for input data. The image pyramid reduces the size of the image proportionally, and in this process, both the number and value of pixels change. Correspondingly, changes in effective perturbation pixels will reduce the attack ability of adversarial samples. Therefore, this chapter uses an approach that only interpolates the perturbation and fusions perturbations of different sizes to enhance the ability of adversarial samples to resist shrinking, thus achieving the goal of attacking MTCNN. Additionally, we present image quality assessment baseline using PSNR, SSIM, and LPIPS on the

CelebA dataset, which serves as a standard for future research in this field. The valuable findings of this research are not limited to this, and MTCNN's model structure is not complicated, but the image pyramid provides it with strong anti-attack capabilities, which is worth further exploration.

Chapter 5. This chapter proposes a method of adding black lines on the face to attack MTCNN, SSD, and S3FD face detectors. Its effectiveness is verified in the CelebA and FFHQ datasets. The attacking ability is enhanced with the increase of line segment width. The black line structure connects the coordinates of eyes, mouth corners, and boundary box vertices detected by MTCNN. Analysis of the visualization of adversarial samples by neural networks reveals that in the successfully attacked images, the black line structure has distinct boundaries and spaces, while in the failed attacked images, it is not obvious. Therefore, it is determined that the black line structure is a method for attacking the continuity of facial features. However, covering the face reduces the usability of images and may not be helpful to users with privacy and security requirements. Therefore, structure optimization experiments and image quality optimization experiments are also conducted in this chapter.

In the structure optimization experiments, we designed five structures. They are 1) removing the segment between the eyes and the segment between the corners of the mouth, 2) removing the segment connecting the eyes to the border vertex and the segment connecting the corners of the mouth to the border vertex, 3) removing the segment between the eyes and the corners of the mouth, 4) reducing the width of the segments to 2 pixels, and 5) the short-line structure which shortens the length of each segment. Based on these five structures, we conducted the effectiveness experiments respectively. The results showed that only the short-line structure could maintain a certain level of attack ability, however, the adversarial samples produced under this structure are still not easily accepted visually.

In the image quality optimization experiment, we proposed a method of adding random perturbations within the structural scope. These random perturbations are below 45% coverage and maintain the attacking capability in tests on the CelebA and FFHQ data. The image quality ex-

periment established a baseline with the face completely covered by black lines. The experiment compared the image quality scores of black line structures and random perturbations using the baseline, and the results showed that random perturbations with low coverage could significantly improve image quality, achieving the goal of optimizing image quality. Finally, we also proposed a random perturbation optimization method based on genetic algorithms and designed a fitness function that balances image quality and attacking capability.

In order to further confirm the acceptability of this method by users, we conducted a questionnaire survey on image privacy and security needs. The survey results indicated that 34.2% of users were able to accept adversarial samples under random perturbation, which was an improvement of 28.8% compared to the black-line structure and 34.2% compared to the image quality baseline. Additionally, the demand for this method is high among personnel in departments such as law enforcement and government. Our study demonstrates that adding random perturbations as an optimization technique can improve image quality, be acceptable to users, and provide a solution for users with image security and privacy requirements.

6.2 Future Work

In protecting the privacy of faces in images, I believe there are still more methods that have yet to be discovered. In future research, I will face these challenges and seek broader solutions for image privacy protection to help more people. In future work, I will focus on the following three directions:

Reducing the amount of perturbation in multi-scale perturbation fusion and finding methods to attack multiple face detectors with imperceptible disturbance. Optimizing random perturbation to enhance image quality further. Exploring the applicability of image pyramids in defense against attacks.

Acknowledgements

During the six years from Master to Ph.D., I would like to thank the Tokyo University of Technology for providing me with a learning environment. With a long history, this school has brought me a relaxed and focused learning atmosphere.

I would like to give my heartfelt thanks to Prof. Hiroyuki KAMEDA for his guidance in my study and scientific research. From Prof. Hiroyuki KAMEDA , I learned how to become a scientific researcher and understood the meaning of conducting scientific research. Every research should be beneficial to society, and the research results have their value, so we should persevere and keep the original intention. Maintaining a pure love for scientific research and doing research freely according to my ideas are my most significant gains in Kameda Laboratory. This sense of social responsibility will accompany me throughout my life and has become my guideline for doing things. Thank you again, Prof. Hiroyuki KAMEDA .

I would like to express my gratitude to Prof. Terumasa AOKI, Prof. Yuichi FUTA, Prof. Masayuki KIKUCHI, Prof. Kiyohiko HATTORI, and Prof. Hiroyuki KAMEDA for their guidance on this thesis. It clarified the research purpose of the thesis and the nature of the research results and completed the last piece of the puzzle by adding supplementary experiments. It is precise because of the professors' guidance on the details and structure of the thesis that today's complete doctoral thesis is possible. My heartiest thanks flow to professors.

I would also like to thank the students in the Kameda Laboratory for their support and encouragement. Especially, QI Yu is my partner in research and a good friend who helps me develop

ideas and progress together.

Finally, I would like to thank my family. Without your selfless support, I would not have the courage to study for a Ph.D. at 30. It is your encouragement that allowed me to complete my final studies.

Reference

[1] Naveed Akhtar and Ajmal Mian. "Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey". In: *IEEE Access* 6 (2018), pp. 14410–14430.

[2] Maximilian Alber, Sebastian Lapuschkin, Philipp Seegerer, Miriam Hägele, Kristof T Schütt, Grégoire Montavon, Wojciech Samek, Klaus-Robert Müller, Sven Dähne, and Pieter-Jan Kindermans. "iNNvestigate Neural Networks!" In: *Journal of Machine Learning Research* 20.93 (2019), pp. 1–8.

[3] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. "On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation". In: *PLOS ONE* 10.7 (2015), e0130140.

[4] J. E. Bresenham. "Algorithm for computer control of a digital plotter". In: *IBM Systems Journal* 4.1 (1965), pp. 25–30.

[5] Kang-Tsung Chang. "Introduction to Geographic Information Systems". In: *McGraw-Hill Boston* 4 (2008).

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. "Imagenet: A large-scale hierarchical image database". In: *IEEE Conference on Computer Vision and Pattern Recognition* (2009), pp. 248–255.

[7] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. "Robust Physical-World Attacks on Deep

Learning Visual Classification". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 1625–1634.

[8] Faceswap. In: *https://faceswap.dev/* (2022).

[9] FakeApp. In: *https://www.malavida.com/en/soft/fakeapp/* (2023).

[10] Yoav Freund and Robert E Schapire. "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting". In: *Journal of Computer and System Sciences* 55.1 (1997), pp. 119–139.

[11] Kunihiko Fukushima and Sei Miyake. "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". In: *Competition and Cooperation in Neural Nets* (1982), pp. 267–285.

[12] Ross Girshick. "Fast R-CNN". In: *Proceedings of the IEEE International Conference on Computer Vision* (2015), pp. 1440–1448.

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative Adversarial Networks". In: *Communications of the ACM* 63.11 (2020), pp. 139–144.

[14] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. "Explaining and Harnessing Adversarial Examples". In: *Proceedings of the International Conference on Learning Representations* (2015).

[15] David Haussler. "Probably approximately correct learning". In: *University of California, Santa Cruz, Computer Research Laboratory* (1990).

[16] John H Holland. "Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence". In: *MIT Press* (1992).

[17] Petr Hurtik and Nicolas Madrid. "Bilinear Interpolation over fuzzified images: Enlargement". In: *IEEE International Conference on Fuzzy Systems* (2015), pp. 1–8.

[18] Brian Kenji Iwana, Ryohei Kuroki, and Seiichi Uchida. “Explaining Convolutional Neural Networks using Softmax Gradient Layer-wise Relevance Propagation”. In: *IEEE/CVF International Conference on Computer Vision Workshop* (2019), pp. 4176–4185.

[19] Tero Karras, Samuli Laine, and Timo Aila. “A Style-Based Generator Architecture for Generative Adversarial Networks”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 4401–4410.

[20] Edgar Kaziakhmedov, Klim Kireev, Grigorii Melnikov, Mikhail Pautov, and Aleksandr Petiushko. “Real-world Attack on MTCNN Face Detection System”. In: *International Multi-Conference on Engineering, Computer and Information Sciences* (2019), pp. 0422–0427.

[21] Robert Keys. “Cubic convolution interpolation for digital image processing”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 29.6 (1981), pp. 1153–1160.

[22] Diederik P Kingma and Max Welling. “An Introduction to Variational Autoencoders”. In: *Foundations and Trends® in Machine Learning* 12.4 (2019), pp. 307–392.

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Communications of the ACM* 60.6 (2017), pp. 84–90.

[24] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. “Adversarial Examples in the Physical World”. In: *Artificial Intelligence Safety and Security* (2018), pp. 99–112.

[25] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. “Backpropagation Applied to Handwritten Zip Code Recognition”. In: *Neural Computation* 1.4 (1989), pp. 541–551.

[26] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[27] Haoxiang Li, Zhe Lin, Xiaohui Shen, Jonathan Brandt, and Gang Hua. “A Convolutional Neural Network Cascade for Face Detection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 5325–5334.

[28] Yuezun Li, Xin Yang, Baoyuan Wu, and Siwei Lyu. “Hiding Faces in Plain Sight: Disrupting AI Face Synthesis with Adversarial Perturbations”. In: *arXiv preprint arXiv:1906.09288* (2019).

[29] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. “Ssd: Single Shot Multibox Detector”. In: *European Conference on Computer Vision* (2016), pp. 21–37.

[30] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. “Deep Learning Face Attributes in the Wild”. In: *Proceedings of the IEEE International Conference on Computer Vision* (2015), pp. 3730–3738.

[31] Jonathan Long, Evan Shelhamer, and Trevor Darrell. “Fully Convolutional Networks for Semantic Segmentation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 3431–3440.

[32] Jiajun Lu, Hussein Sibai, and Evan Fabry. “Adversarial Examples that Fool Detectors”. In: *arXiv preprint arXiv:1712.02494* (2017).

[33] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. “Towards Deep Learning Models Resistant to Adversarial Attacks”. In: *arXiv preprint arXiv:1706.06083* (2017).

[34] Vinod Nair and Geoffrey E Hinton. “Rectified Linear Units Improve Restricted Boltzmann Machines”. In: *International Conference on Machine Learning* (2010).

[35] Yuval Nirkin, Yosi Keller, and Tal Hassner. “FSGAN: Subject Agnostic Face Swapping and Reenactment”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), pp. 7184–7193.

[36] OpenCV. In: *https://opencv.org/* (2022).

[37] OpenFaceSwap. In: *https://opencv.org/* (2023).

[38] Ivan Perov, Daiheng Gao, Nikolay Chervoniy, Kunlin Liu, Sugasa Marangonda, Chris Umé, Mr. Dpfks, Carl Shift Facenheim, Luis RP, Jian Jiang, Sheng Zhang, Pingyu Wu, Bo Zhou, and Weiming Zhang. "DeepFaceLab: A simple, flexible and extensible face swapping framework". In: *CoRR* abs/2005.05535 (2020).

[39] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. "You Only Look Once: Unified, Real-Time Object Detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 779–788.

[40] Henry A Rowley, Shumeet Baluja, and Takeo Kanade. "Neural network-based face detection". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20.1 (1998), pp. 23–38.

[41] Henry A Rowley, Shumeet Baluja, and Takeo Kanade. "Rotation invariant neural network-based face detection". In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (1998), pp. 38–44.

[42] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. "Learning Internal Representations by Error Propagation". In: *California Univ San Diego La Jolla Inst for Cognitive Science* (1985).

[43] G. Salton, A. Wong, and C. S. Yang. "A Vector Space Model for Automatic Indexing". In: *Commun. ACM* 18.11 (1975), pp. 613–620.

[44] Robert E Schapire. "The strength of weak learnability". In: *Machine learning* 5.2 (1990), pp. 197–227.

[45] Oscar Schwartz. "You thought fake news was bad? Deep fakes are where truth goes to die". In: *The Guardian* 12 (2018).

[46] David Sheehan. “Visualising Activation Functions in Neural Networks”. In: *Dashee87. Github. Io* (2020).

[47] Karen Simonyan and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *International Conference on Learning Representations* (2015).

[48] Zekun Sun, Yujie Han, Zeyu Hua, Na Ruan, and Weijia Jia. “Improving the Efficiency and Robustness of Deepfakes Detection Through Precise Geometric Features”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 3609–3618.

[49] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, D. Erhan, Ian J. Goodfellow, and Rob Fergus. “Intriguing properties of neural networks”. In: *CoRR* abs/1312.6199 (2014).

[50] Simen Thys, Wiebe Van Ranst, and Toon Goedemé. “Fooling Automated Surveillance Cameras: Adversarial Patches to Attack Person Detection”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop* (2019).

[51] Paul Viola and Michael Jones. “Rapid object detection using a boosted cascade of simple features”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 1 (2001), pp. I–I.

[52] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. “Image quality assessment: from error visibility to structural similarity”. In: *IEEE Transactions on Image Processing* 13.4 (2004), pp. 600–612.

[53] Matthew D Zeiler and Rob Fergus. “Visualizing and Understanding Convolutional Networks”. In: *European Conference on Computer Vision* (2014), pp. 818–833.

[54] Matthew D Zeiler, Graham W Taylor, and Rob Fergus. “Adaptive deconvolutional networks for mid and high level feature learning”. In: *International Conference on Computer Vision* (2011), pp. 2018–2025.

[55] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. “Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks”. In: *IEEE Signal Processing Letters* 23.10 (2016), pp. 1499–1503.

[56] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. “The Unreasonable Effectiveness of Deep Features as a Perceptual Metric”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 586–595.

[57] Shifeng Zhang, Xiangyu Zhu, Zhen Lei, Hailin Shi, Xiaobo Wang, and Stan Z Li. “S3FD: Single Shot Scale-Invariant Face Detector”. In: *Proceedings of the IEEE International Conference on Computer Vision* (2017), pp. 192–201.

[58] Hanqing Zhao, Wenbo Zhou, Dongdong Chen, Tianyi Wei, Weiming Zhang, and Nenghai Yu. “Multi-Attentional Deepfake Detection”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 2185–2194.

[59] Tianfei Zhou, Wenguan Wang, Zhiyuan Liang, and Jianbing Shen. “Face Forensics in the Wild”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 5778–5788.

List of Publications Related to This Thesis

Journal

Zhang, Chongyang, and Kameda, Hiroyuki. "A New Method of Disabling Face Detection by Drawing Lines between Eyes and Mouth." Journal of Computers 11.9 (2022): 134.

International Conference

Zhang, Chongyang, Yu Qi, and Kameda, Hiroyuki. "Multi-scale Perturbation Fusion Adversarial Attack on MTCNN Face Detection System." 2022 4th International Conference on Communications, Information System and Computer Engineering (CISCE). IEEE, 2022.

List of Publications

Journal

Qi, Yu, Zhang, Chongyang, and Kameda, Hiroyuki . "Animal Exercise: A New Evaluation Method. "Journal of Computer Science Research, 4.2 (2022): 24-30. Web. 25 Oct. 2022.

Qi, Yu, Zhang, Chongyang, and Kameda, Hiroyuki. "Motion Transfer in Crawling Stance for Performance Teaching". Journal of Computers Under Review.

International Conference

Qi, Y., Zhang, C. Y., and Kameda, H. Y. Historical Summary and Future Development Analysis of Animal Exercise. In ICERI2021 Proceedings (pp. 8529-8538). (2021). IATED.