Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research." If a, user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use" that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select "Pages from: first page # to: last page #" on the print dialog screen

ABSTRACT

DEEP LEARNING ON IMAGE FORENSICS AND ANTI-FORENSICS

by
Zhangyi Shen

Image forensics protect the authenticity and integrity of digital images. On the contrary, as the countermeasures of digital forensics, anti-forensics is applied to expose the vulnerability of forensics tools. Consequently, forensics researchers could develop forensics tools against possible new attacks. This dissertation investigation demonstrates two image forensics methods based on convolutional neural network (CNN) and two image anti-forensics methods based on generative adversarial network (GAN).

Detecting unsharp masking (USM) sharpened image is the first study in this dissertation. A CNN architecture comprises four convolutional layers and a classification module is proposed to discriminate sharpened images and unsharpened images. The results exhibit the superiority of the proposed CNN model over the existing sharpening detection method, i.e., edge perpendicular ternary coding (EPTC). The second study is to detect recolored images. Unlike the conventional binary classifieds, the proposed method based on CNN can be employed for binary classification as well as multiple labels classification. In order to accelerate the training process, the normalization layer is discarded in the proposed CNN. The proposed model can reach detection accuracy over 90% under all circumstances. The detection performance is perfect even when the images are weakly sharpened. To investigate the possible vulnerabilities of sharpening detectors, a GAN model is proposed to behave as an anti-forensics tool. In this study, after adversarial training, the proposed GAN model generates images with the sharpening features. However, these pictures cannot be regarded as sharpened ones. Observed from the

experimental results, even the state-of-the-art sharpening detector based on CNN can be also deceived with the images generated by our proposed model. Finally, the fourth study is to investigate whether GAN can be supervised to generate images that can impede forensics detectors from making correct decision. A GAN model with a novel architecture is proposed. Proved by our simulations, the proposed model can be applied to attack forensics detectors on variety common image manipulations.

DEEP LEARNING ON IMAGE FORENSICS AND ANTI-FORENSICS

by
Zhangyi Shen

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Computer Engineering

Helen and John C. Hartmann
Department of Electrical and Computer Engineering

May 2021

APPROVAL PAGE

DEEP LEARNING ON
IMAGE FORENSICS AND ANTI-FORENSICS

Zhangyi Shen

Dr. Mengchu Zhou, Dissertation Advisor — Date
Distinguished Professor of Electrical and Computer Engineering, New Jersey Institute of Technology

---

Dr. Yun Q. Shi, Committee Member — Date
Professor of Electrical and Computer Engineering, New Jersey Institute of Technology

---

Dr. Xuan Liu, Committee Member — Date
Associate Professor of Electrical and Computer Engineering, New Jersey Institute of Technology

---

Dr. John D. Carpinelli, Committee Member — Date
Professor of Electrical and Computer Engineering, New Jersey Institute of Technology

---

Dr. Edwin Hou, Committee Member — Date
Professor of Electrical and Computer Engineering, New Jersey Institute of Technology

---

Dr. Frank Y. Shih, Committee Member — Date
Professor of Computer Science, New Jersey Institute of Technology

BIOGRAPHICAL SKETCH

Author: Zhangyi Shen

Degree: Doctor of Philosophy

Date: May 2021

Undergraduate and Graduate Education:

- Doctor of Philosophy in Computer Engineering,
  New Jersey Institute of Technology, Newark, NJ, 2021
- Master of Science in Computer Engineering,
  New Jersey Institute of Technology, Newark, NJ, 2016
- Bachelor of Information Security,
  Hangzhou Dianzi University, Zhejiang, P. R. China, 2015

Major: Computer Engineering

Presentations and Publications:

Shen Z, Ding F, Shi Y., "Digital Forensics for Recoloring via Convolutional Neural Network," *Computers, Materials and Continua*, vol. 62, pp 1-16, 2020.

Shen Z, Ding F, Shi Y., "Anti-forensics of Image Sharpening Using Generative Adversarial Network," *International Workshop on Digital Watermarking. Springer, Cham*, pp 150-157, 2019.

Ye J, Shen Z, Behrani P, et al., "Detecting USM image sharpening by using CNN," *Signal Processing: Image Communication*, vol. 68, pp 258-264, 2018.

To my father, Zhengxue Shen and
my mother, Hong Zhang

献给我的父亲，沈正学和我的母亲，张红

ACKNOWLEDGMENT

I would like to thank Dr. Yun Q. Shi first for his guidance throughout my Ph.D program as well as his technical support for these years. Also, I would like to thank Dr. Mengchu Zhou. Dr. Zhou helped me a lot in my study and life in NJIT. Without their tremendous assistance and endless encouragement, I could not have completed my dissertation.

In addition, I would like to express my appreciation to my committee members: Drs. Xuan Liu, John Carpinelli, Edwin Hou, Frank Shih, for their active involving and instructive comments to my dissertation.

Then, I would like to thank my parents for their financial support to me in these years. They always stand behind me and offer help whether it's a physical or mental one whenever I need them. I could not have held on till now without their support and understanding.

Finally, I would like to thank my peer, Yuxi Shi. Yuxi gave me a lot of assistance in my life and study in these years and he will always be my role model.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

Chapter **Page**

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF TABLES

LIST OF TABLES
(Continued)

Table **Page**

LIST OF FIGURES

LIST OF FIGURES
(Continued)

Figure **Page**

CHAPTER 1

INTRODUCTION

1.1 Overview

Images provide information to human eyes and had been considered as secured information evidence in everyday life controversies and in trials. Images have also been utilized in various media such as newspapers, magazines and television. With rapid advancement of digital technologies, digital images have replaced the traditional photo images. For the convenience of modern life more and more techniques have been created, including image editing techniques. Since common people have acquired the ability to tamper, forge or modify digital images, the reliability of images serving as evidence has become an issue [44]. Therefore, scientists use image forensics [19][23] to identify if a given image has been illegally or improperly manipulated or not. Furthermore, as the digital images become a main information carrier in our daily life, it is necessary for us to know whether a given image has been processed or not, and if so, to what extent it has been altered needs to be exposed. In recent years, deep learning methods, such as CNN, have been justified frequently to be perfect image forensics tools.

As the antithesis of forensics, anti-forensics used to improve the original image processing methods to counter detection. Yet the appearance of generative adversarial networks (GANs) changed the situation. Unlike the most proposed models which focus on classification, GANs are designed for creation. Anti-forensics researchers started to use GANs to generate images which have similar visual effects of the images generated by traditional image processing methods and can hardly be detected by the corresponding proposed detection models.

In this dissertation, forensics and anti-forensics of digital images are mainly treated as classification problems so as to make deep learning methods applicable.

Four topics are covered: (1) detecting USM sharpening by using CNN; (2) detecting image recoloring by using CNN; (3) anti-forensics of detecting USM sharpening by using GAN; And (4) a general image anti-forensic method based on a generative adversarial network.

1.2 Contributions Made in This Dissertation Study

ML-based forensics aims to distinguish images with various image manipulation methods, and their corresponding cover images. Since advanced image processing methods alter the pixel values in the image regions, traditional ML-based forensics heavily relied on sophisticated manual feature design. To improve the state-of-the-art of some image forensics problems achieved by conventional feature engineering, deep learning is considered to be a good solution since CNN-based deep neural networks require less human involvement on feature engineering, and both feature extraction and classification are jointly optimized in the training. The outcomes are (1) and (2). A CNN architecture is proposed to detect images sharpened by the USM methods in (1). Also, I designed a new structure of CNN and applied it on another cases in (2), which is to detect image recoloring methods. These two works can be considered as the-state-of-the-art when they were published. Anti-forensics is also a very popular problem with the development of forensics. With the help of GANs, (3) is proposed to challenge (1). The model in (3) is able to generate sharpened images and the results show that these images cannot be classified accurately by the proposed CNN model in (1). Since forensics becomes more and more powerful with the help of CNN, the method in (3) is expected to be future trend of anti-forensics. (4) is an evolved version of (3), it covers several image processing effects by one GAN structure instead of just generating sharpened images.

1.3 Outline of This Dissertation Report

In Chapter 2, a CNN based method for detecting USM sharpening is implemented. the detection history of USM sharpening, the architectural design of CNNs, and the ensemble study of CNNs for forensics are described in detail. Chapter 3 elaborates the detection of image recoloring. In Chapter 4, a GAN based method is implemented to against detection of USM sharpening. In Chapter 5, a general image anti-forensics method based on GAN is employed to challenge CNN based detectors. Finally, this dissertation is summarized in Chapter 6.

CHAPTER 2

CONVOLUTIONAL NEURAL NETWORKS FOR DETECTING USM IMAGE SHARPENING

2.1 USM Image Sharpening and the Detection of Sharpened Image

Image sharpening [27] is one of the widely used manipulations on digital images. Through this manipulation people would like to obtain clearer detailed information from images because after image sharpening, the contrast of image is enhanced, edges, outlines and details become clearer. Figure 2.1 is an example of sharpening.

(a) Original image (b) Sharpened image

Figure 2.1 The comparison between original image and sharpened image. Source: [2]

Since it is important to trace the processing history of a given image, detecting image sharpening has become important in image forensics. Since 2009, several methods [9][10][17] have been proposed for Unsharp Masking (USM) sharpening detection. In [9], Cao *et al.* discovered the histogram aberration after image sharpening and a method was proposed to detect such artifact. But according to their latest study [10], the method in [9] is only effective to detect the images with wide

histogram. To overcome this drawback, Cao *et al.* [10] proposed another detection method. Firstly, the method detects the edge pixels of a given image. Secondly, the set of side-planar crosswise pixel sequences are located on the basis of detected edge pixels. Then, for each side-planar pixel sequence an overshoot strength is calculated and the overshoot metric of the whole image is measured by average of the overshoot strengths. Finally, threshold is applied on overshoot metric to make a binary decision for sharpening detection. The method in [10] can overcome the drawback of [9] and is more effective. But it is vulnerable to JPEG compression and image sharpening, which limit its use in practical applications. By regarding the appearance of overshoot artifacts as a special kind of texture modification, a detection method [17] was proposed, which is based on a widely used texture classification technique called local binary pattern (LBP) [46][47]. The LBP-based method has been validated to be more accurate for sharpening detection compared with the method in [10]. In contrast to the methods in [9][10][17], Lu *et al.* proposed a method to remove overshoot artifacts for anti-forensics of USM sharpening [38]. Inspired by the LBP-based method [17], Ding *et al.* [18] proposed a novel method called Edge Perpendicular Binary Coding (EPBC) to detect USM sharpening. Considering that the texture modification caused by USM sharpening is high mainly along the perpendicular direction of image edges, the EPBC uses a long rectangular window perpendicular to edge to characterize image textures. To avoid the long rectangular window used in EPBC, a particular ternary coding strategy is proposed by Ding *et al.* called Edge Perpendicular Ternary Coding (EPTC) [14] which is better than the method reported by Ding *et al.* [18]. EPTC was the best algorithm for image sharpening detection before this study has been published. In 2020, Wang *et al.* proposed an algorithm [70] based on difference sets composed of first-order and second-order differences in different directions on image and their results exceeded EPTC's.

2.1.1 USM Image Sharpening Algorithm

Unsharp masking (USM) is a common algorithm for image sharpening. The algorithm is used to improve the visual effect of image by increasing contrast at the high frequency part, such as the marginal area of the image. The sharpening process is implemented by adding a scaled unsharp mask M to the original image, the specific formula is:

$$Y = X + \lambda M \tag{2.1}$$

where X, Y, M and λ denote, respectively, input image, output image, unsharp mask and scaling coefficient.

In traditional USM algorithm, unsharp mask is generated through processing a high-pass filtering on the original image. The formula is:

$$Y = X \bigotimes H \tag{2.2}$$

where $\bigotimes$ and H denote convolution operator and a high-pass filter, respectively.

However, unsharp mask can also be generated via Gaussian filtering, as expressed by

$$Y = X - X \bigotimes G_\sigma \tag{2.3}$$

where G_σ denotes a Gaussian filter with variance σ, which can control the range of sharpening.

According to Equation 2.1 and 2.3, there are two parameters to control the sharpening process. One is the scaling coefficient λ and another is the variance of Gaussian filter σ. By using a combination with different values of λ and σ, the effects of sharpening will be different accordingly. After sharpening, the edge, the contour line as well as the details of image will become clearer than in the original image. In this chapter, images sharpened with different values of λ and σ will be discussed in the experiment section.

2.1.2 Edge Perpendicular Ternary Coding (EPTC)

The edge perpendicular ternary coding (EPTC), which is considered as the comparative method to our proposed CNN structure, is utilized to detect sharpened image and has been reported in [14]. The method was inspired by the previous study that uses LBP [17] for sharpening detection. The detailed steps are listed as follows.

Step 1. *Edge detection.* Canny operator [8], a popular edge detector, is used for this step. This operator can detect the edge pixels of image.

Step 2. *Determination of local edge datasets.* Set a rectangular window of $1 \times N$ pixels which is perpendicular to the direction of the edge, N is an odd number no less than 3. Also, the center of the window is the edge pixel. Then, a pixel set S_1 could be obtained, which can be represented as

$$S_1 = [P_0, P_1, ..., P_{N-1}] \tag{2.4}$$

where P_i donates the pixel values in the rectangular window. The edge pixel is the element $P_{(N-1)/2}$.

Step 3. *Ternary coding.* First, calculate the difference between each pixel and the pixels on its right side. As a result, a new data set S_2 can be obtained as

$$S_2 = [P_0 - P_1, P_1 - P_2, ..., P_{N-2} - P_{N-1}] \tag{2.5}$$

Second, convert S_2 into a ternary code T as follows.

$$T = [T_0, T_1, ..., T_{N-2}] \tag{2.6}$$

Where

$$T_n = \begin{cases} 1 & P_n - P_{n+1} > q \\ 0 & q \geq P_n - P_{n+1} \geq -q \\ -1 & P_n - P_{n+1} < -q \end{cases} \tag{2.7}$$

where q is a positive number calculated by

$$q = \frac{\sum_{i=0}^{i=\frac{N-3}{2}} \|P_i - P_{i+1}\| + \sum_{i=\frac{N+3}{2}}^{i=N-1} \|P_i - P_{i+1}\|}{N-3} \tag{2.8}$$

Step 4. *Calculation of EPTC histogram.* From the ternary code T obtained above, a transferred set T' is derived by

$$T' = [T'_0, T'_1, ..., T'_{2N-3}] \tag{2.9}$$

For $n < N-1$

$$T'_n = \begin{cases} 1 & T_n = 1 \\ 0 & \text{otherwise} \end{cases} \tag{2.10}$$

For $n \geq N-1$

$$T'_n = \begin{cases} 1 & T_{n-(N-1)} = -1 \\ 0 & \text{otherwise} \end{cases} \tag{2.11}$$

Then, a decimal number, denoted as $EPTC(T)$, can be derived as shown below.

$$EPTC(T) = \sum_{n=0}^{n=2N-3} T'_n \times 2^n \tag{2.12}$$

As we can see, $EPTC(T)$ is in the range of $[0, 2^{2N-3} - 1]$. These $2^{2N-3} - 1$ different integer values can be regarded as $2^{2N-3} - 1$ patterns. The histogram of the EPTC pattern for a given image can be calculated as follows:

$$\begin{aligned} H(i) &= \sum_{T \in \Delta} \delta(EPTC(T) - i) \\ i &= 0, 1, ..., 2^{2N-3} - 1 \end{aligned} \tag{2.13}$$

Where $\delta(x)$ denotes the indicator function which equals to 1 if $x = 0$ and 0 otherwise, Δ denotes the sets of T calculated from all of the local edge areas in the given image. Also, to make H invariant to image, H could be normalized as follows.

$$\begin{aligned} \tilde{H}(i) &= \frac{H(i)}{\sum_{i=0}^{2^{2N-3}-1} H(i)} \\ i &= 0, 1, ..., 2^{2N-3} - 1 \end{aligned} \tag{2.14}$$

This normalized histogram $\tilde{H}$ is used as feature of given images.

Step 5. *SVM training and classification.* Finally, the features obtained in Equation (2.14) need to be prioritized and selected and then, used as input to SVM classifier. This trained SVM is able to recognize whether images have been sharpened or not.

2.2 Convolutional Neural Network (CNN)

The Convolutional Neural Networks (CNNs) have made tremendous achievements in computer vision since 2012. This has aroused interests of researchers to seek the way to use CNNs for image forensics. In this chapter, a CNN based approach is proposed to detect image sharpening. Experimental results have shown that the use of CNNs gives better results in image sharpening detection than the EPTC does, which is so far the best method to detect USM based sharpening.

2.2.1 History of CNN

In 1989, a neural network named 'Net-5' was designed to solve a handwritten digit recognition problem [39]. The author had two major contribution in designing the network. The first one is reducing the number of free parameters to gain better generalization, and the second one is to force hidden units to learn from local information in order to achieve better results. The hidden layers in the Net-5 are composed of several feature maps, while each unit in one feature map is connected to units within a fixed size neighborhood, for instance 3×3, on the input plane. Therefore, the number of free parameters is largely reduced comparing with traditional fully connected neural networks. Furthermore, all units in a feature map share the same set of weights, and subsampling is utilized as well to reduce the complexity of the network. Thus, much less parameters are employed during the computation. Besides, back propagation technique [5] is also employed to train the neural network. According to the reported results, Net-5 has achieved the best performance among five compared structures. Noted, Net-5 is the first CNN as known and the idea behind is still the essence of today's various deep CNNs. Later in [40], the above introduced investigation was applied on recognizing handwritten digits taken by U.S. Mail, and the network has been extended from Net-5's two hidden layers to three hidden layers, including two convolutional layers, and one fully connected layer. Although the number of convolutional layer was not increased, the number of kernels adopted, that is the set of in each hidden layer is significantly increased. The results turned out to be the state of the art. In [41], another CNN named 'LeNet-5' is proposed for handwritten character recognition. However, the network is still shallow. The first deep CNN architecture called 'AlexNet' [37] was presented in 2012. This network has achieved remarkable success in the ILSVRC-2012 competition which is considered as a huge step to the machine learning society. In 'AlexNet', five convolutional layers are employed to generate hierarchical feature maps. Besides,

Max pooling is applied to reduce the size of the network. To increase non-linearity, an activation function named ReLU is utilized in 'AlexNet' as well. Finally, 'AlexNet' achieved a top-5 test error rate of 15.3% on the ImageNet database [57] while the second-best result was 26.2%. After the big success of 'AlexNet', deep CNN has aroused tremendous interests and several successful CNNs have been presented for image classification, such as 'ZF Net' [74], 'VGGNet' [61], 'GoogLeNet' [67], 'ResNet' [31], etc. Not only the study of image classification, deep CNN has been widely spread to other related areas and achieved successes, such as face recognition, human action recognition, and steganalysis [51][49][72] as well.

2.2.2 Common Layers in Design CNN Structure

Neural network is built by the accumulation of different types of layers. By reviewing the proposed CNNs, most of the networks are based on a similar structure that is a hierarchical architecture starts with multiple stages of convolutional modules and ends with a classification module. A common convolutional module includes a convolutional layer, an activation layer, and a pooling layer. By stacking a series of convolutional modules, hierarchical feature maps are extracted and then fed into the classification module composed of one or more fully-connected layers, and the SoftMax layer with cross-entropy loss. In this part, several essential types of layer will be introduced.

A. Convolutional layer

Convolutional layer is a trainable filter bank which can be considered as a feature extractor. It transforms images to feature maps or feature vectors. For example, a 6×6 input image on the left is filtered in order by a 3×3 filter on the upper-left in Figure 2.2. The filter will scan the whole image row by row. Each element in the scanning will be multiplied by the elements on the corresponding position in filter.

The sum of each products in one block will be the result on the corresponding position in feature map. In this example, after scanning the whole image, a 4×4 feature map is generated.

The filter block can also be described as kernel. Normally, there will be several different kernels. As a result, the corresponding feature maps are also increased. Colorful image has three channels (R, G, B), so the kernel should be in three channels so as to the feature maps.

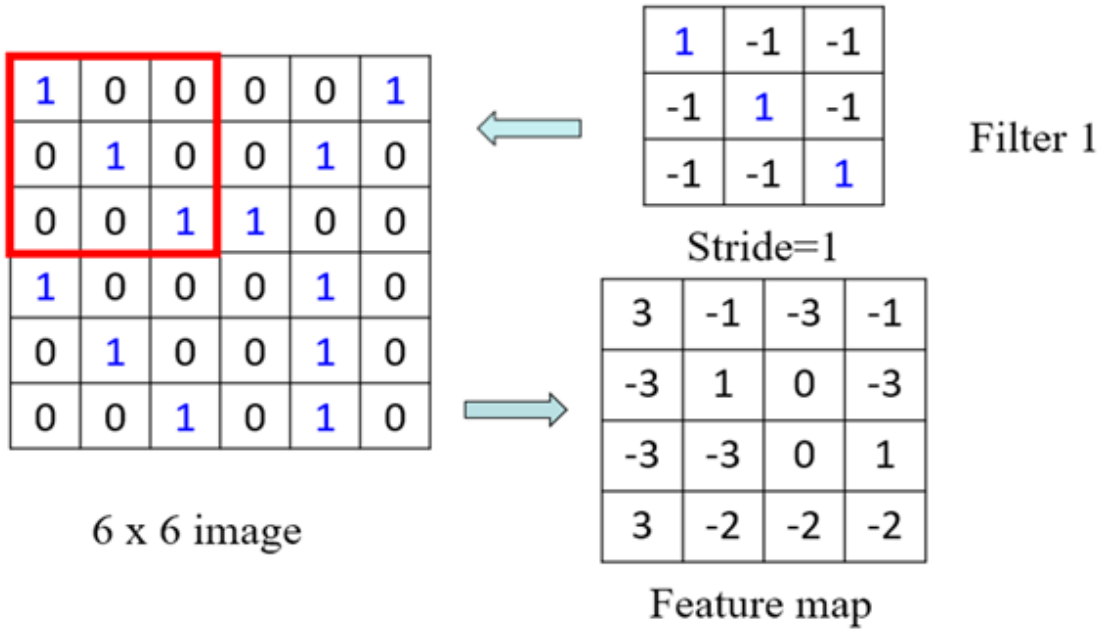
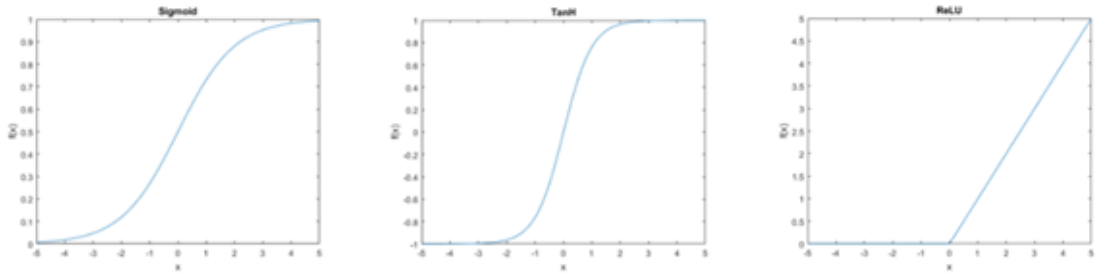


Figure 2.2 The example of convolution.

B. Activation layer

Activation layer brings some nonlinear factors to the neural network so that the neural network can solve the more complex problems better. In CNN designing, there are three popular activation functions- Sigmoid, TanH and ReLU. They are plotted in Figure 2.3. ReLU, as the activation function our CNN structure selected, will be further discussed in Section 2.3 in this chapter.



(a) Sigmoid (b) TanH (c) ReLU

Figure 2.3 Activation functions.

C. Pooling layer

Pooling layer reduces the quantity of features extracted from immediately prior convolutional layer to avoid overfitting. There are two types of pooling layers, average pooling and max pooling. The example of average pooling is shown in Figure 2.4. The result is generated by taking the average values for each pooling mask. Figure 2.5 shows the example of max pooling. The maximum values for each pooling mask are taken as the results.

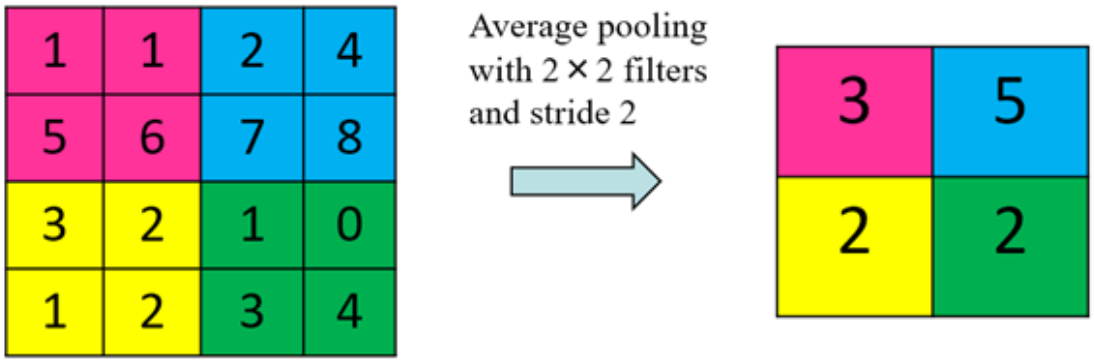


Figure 2.4 The example of average pooling.

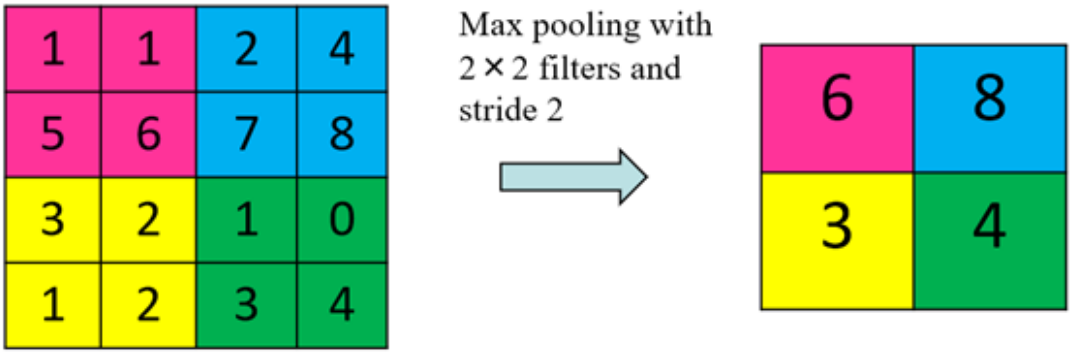


Figure 2.5 The example of max pooling.

D. Normalization layer

Stochastic gradient descent is a process that CNN searches for optimum solution and then optimizes the whole network through back-propagation. More specifically, through back-propagation, weights and biases in convolutional layers will be optimized so as to reduce the training loss, and the power of the network will then be enforced to predict the labels of unseen data. After each stochastic gradient descent (SGD), the corresponding activation is normalized by mini-batch, which makes the mean of the result (each dimension of the output signal) 0 and the variance of 1. This process is the function of normalization layer and can speed up training and improve model precision.

E. Fully connected layer

Fully-connected layer (FC) has the function to map the feature maps or vectors, which are generated from the previous convolutional module, to the setting classes of the network and give scores of this feature set for each class. FC has some hidden layers and each hidden layer has some neurons with learnable weights and setting biases. The input vectors or feature maps are fully connected with the first hidden layers, same with the neurons between adjacent hidden layers. After going through all the hidden layers, the input feature will transform to scores for each class. The

higher the score on one class means that the network thinks the input image is more likely considered as belonging to that class. On the contrary, the lower the score on one class suggests that the input image is considered as that it is not belonging to that class by the CNN model.

F. SoftMax layer

SoftMax layer is to transform the scores from fully-connected layer to probabilities and ensure the sum of them 1. This layer always be the last layer of the classification module.The formula of SoftMax algorithm is shown as follow:

$$y_i = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \tag{2.15}$$

Where y_i stands for the output probability for i class,Z_i and Z_j are input scores, k is a constant, which stands for the number of classes.

The six layers introduced above are the most common layers used in CNN designing. In the next section, our proposed CNN structure will be presented.

2.3 Proposed CNN Structure

Figure 2.6 presents the overall architecture of the proposed CNN structure, which is motivated by "LeNet-5" [41]. The whole CNN contains four convolutional modules and one linear classification module.

The input of the whole network are the original images and the sharpened images. Following the input layer is the convolutional modules. The function of convolutional modules is transforming the images to 64 of 1×1 feature vectors. Since each module contains a group of functional layers, they are presented as group with labels in the figure. The modules are comprised by four groups (displayed as

"Group 1", "Group 2", "Group 3" as well as "Group 4" in Figure 2.6). Each group contains four layers starting with a convolutional layer to generate feature maps. In convolutional layers, the input image is to be filtered by 8 kernels of size 1× (3×3) each (the kernel size follows number of input feature maps × (height × width)) in Group 1. In the following convolutional layers, there are 16 kernels of size 8× (3×3) in Group 2, 32 kernels of size 16× (3×3) in Group 3 and 64 kernels of size 32× (3×3) in Group 4, respectively. Then, the generated feature maps get into Batch Normalization layer, which can prevent data from gradient diffusion. The rectified linear units, which is an efficient activation function, is applied next to enhance the power of statistical modeling. Finally, each group except Group 4 ends with a Max Pooling layer which perform local maximum taking as well as down sampling on the feature maps. The feature maps are to be filtered by a mask of size 5×5 with stride 2 in Group 1, 2, 3. As for Group 4, the pooling layer merges each map to a single element through global averaging. The kernel size for this pooling layer is fixed to the spatial size of the input feature maps. In this way, 64-D features will be generated and then enter the linear classification module.

The linear classification module consists of a fully connected layer and a SoftMax layer. It transforms feature vectors to output probabilities for each class. These probabilities indicate the accuracy of the sharpening detection achieved by our proposed CNN structure.

As an activation function, ReLU has the ability to increase the nonlinearity of CNN model. In addition, ReLU can efficiently speed up the convergence of stochastic gradient descent and process learning optimizing easily because of its piecewise linear nature [26]. As a result, it has been selected as the activation function in the proposed CNN structure.

Rectified linear unit (ReLU) is an activation function defined as

$$F(x) = Max(0, x) \tag{2.16}$$

where x is the input to a neuron.

In addition, max pooling layers have been taken to reduce the sizes of feature maps. Since USM applied high-pass filtering on the original image to generate mask, CNN should have concerns on the contour of the sharpened image. Therefore, compared with average pooling, max pooling is able to reserve more efficient features. The superiority for using the max pooling is also proved by the experimental results shown in Section 2.4.

2.4 Experiment Results

2.4.1 Datasets and Settings

There are two different image data sets have been utilized in our experimental works. One is BOSSbase v1.01 [2], a well-established image database designed for steganography and steganalysis, which contained 10,000 uncompressed grayscale images with size of 512×512. Another image database, which is same as that used for testing the EPTC scheme in [14], is generated by randomly selecting 1,000 images from the UCID and another 1,000 images from the NRCS databases. That is there are totally 2,000 uncompressed grayscale images with size of 384×384.

In Section 2.1.1, we have introduced two parameters: λ and σ. In the experimental works we have worked on eight different combined λ and σ parameter pairs. That is, for each image in the databases, the USM sharpening algorithm is applied with the following eight different combinations, i.e., $'\sigma = 0.7, \lambda = 1.0'$, $'\sigma = 1.0, \lambda = 0.8'$, $'\sigma = 1.0, \lambda = 1.0'$, $'\sigma = 1.0, \lambda = 1.3'$, $'\sigma = 1.0, \lambda = 1.5'$, $'\sigma = 1.3, \lambda = 1.0'$, $'\sigma = 1.3, \lambda = 1.5'$, $'\sigma = 1.5, \lambda = 1.0'$. The experiments have been conducted to detect eight USM sharpening cases so as to evaluate the performance of the proposed CNN architecture. Since we have two datasets as said above, there are in total 18 sharpened image sets. In this study, the USM sharpening algorithm has been implemented in MATLAB.

In the experiments, the proposed CNN architecture has been implemented using a modified version of the Caffe toolbox [33], and stochastic gradient descent is applied to train all the CNNs with the batch size of 64 images. As for the essential parameters used for building a CNN, the momentum is fixed as 0.9 and the weight decay is 0.0005. The learning rate was initialized to 0.001 and forced to decrease 10% after each 5000 iterations. Additionally, we used 2-fold cross validation to conduct our experiment. The method is to divide each of the datasets equally to two parts and first take one

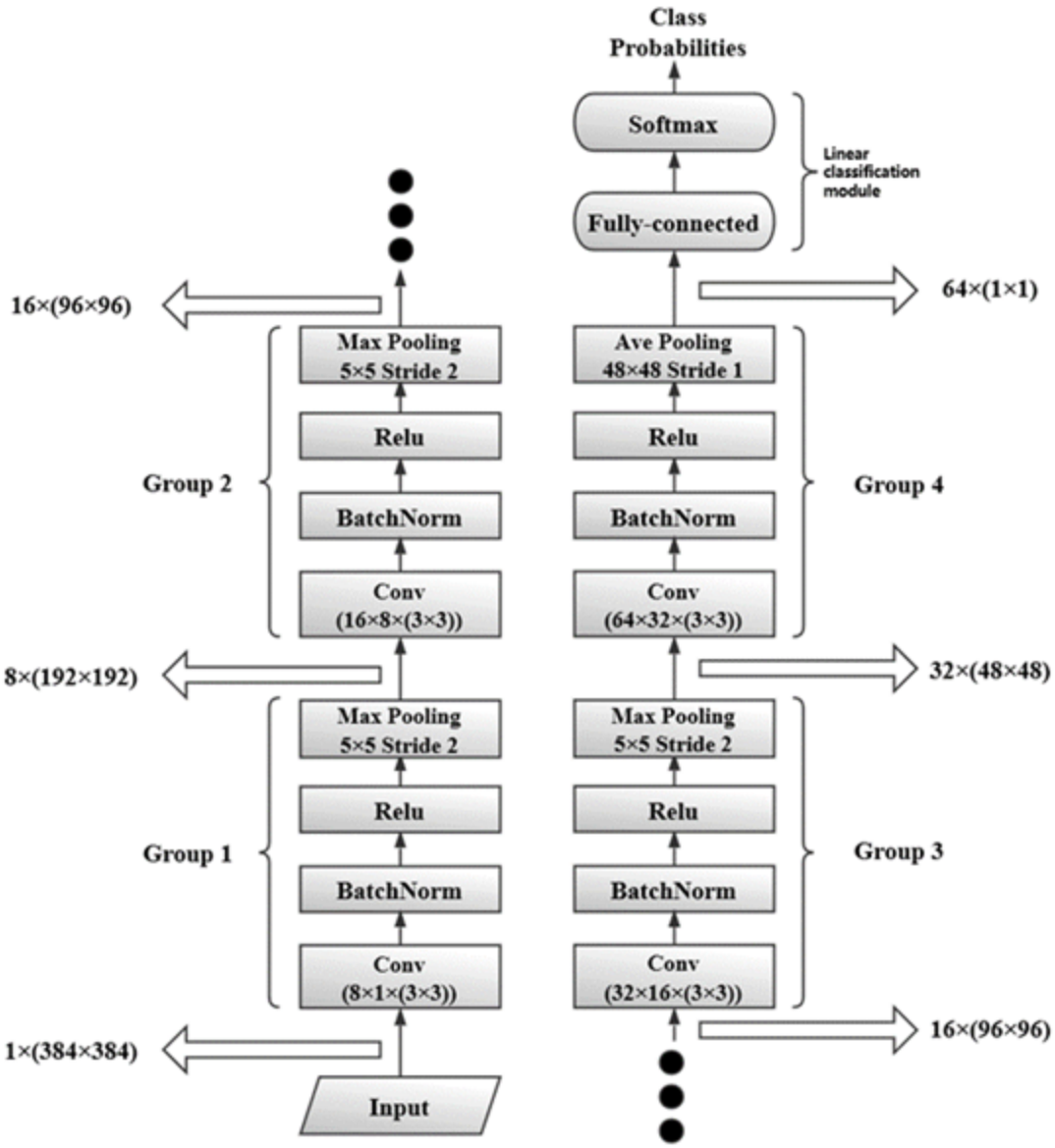


Figure 2.6 Proposed CNN structure. Layers types and parameter choices are displayed inside boxes. Sizes of feature maps are displayed on the two sides, shown as (number of feature maps) × (height× width). Size of convolution kernels are shown in the boxes in the format of (number of kernels × number of input feature maps × (height× width)).

part for training and another part for testing. Then, switch the partition and conduct the experiment again. The final accuracy is the average of the two experiments.

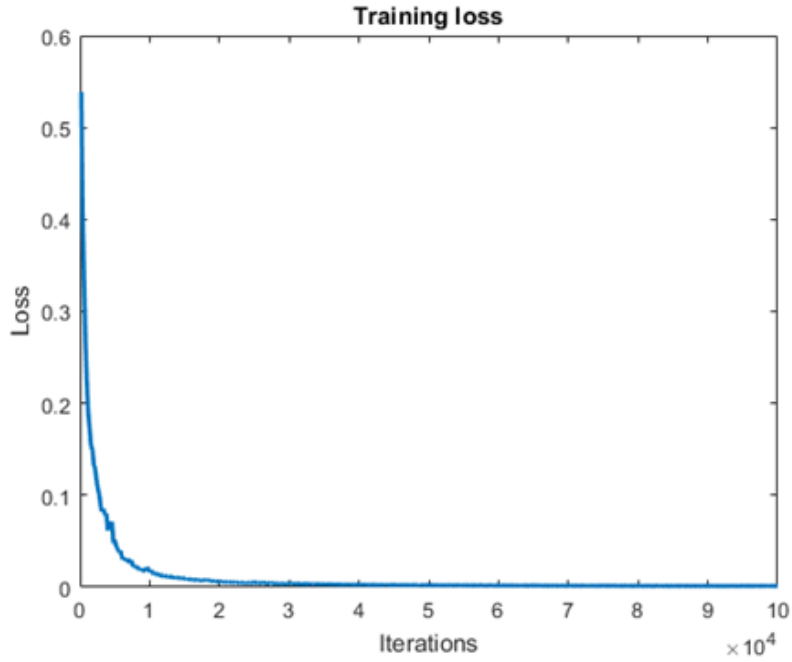
2.4.2 Results

For each group of σ and λ, we ran 10 times of our proposed CNN structure and used the average values of 10 final accuracies as the results of experiment. In addition, we performed the EPTC scheme [14] on these two databases for the performance comparison. By the way, the implementing code of EPTC is provided by the authors of [14]. In addition, we compared our method with EPBC [18] and Wang *et al.* 's method [70]. The results obtained on BOSSbase is recorded in Table 2.1 and the results obtained from UCID & NRCS is recorded in Table 2.2.

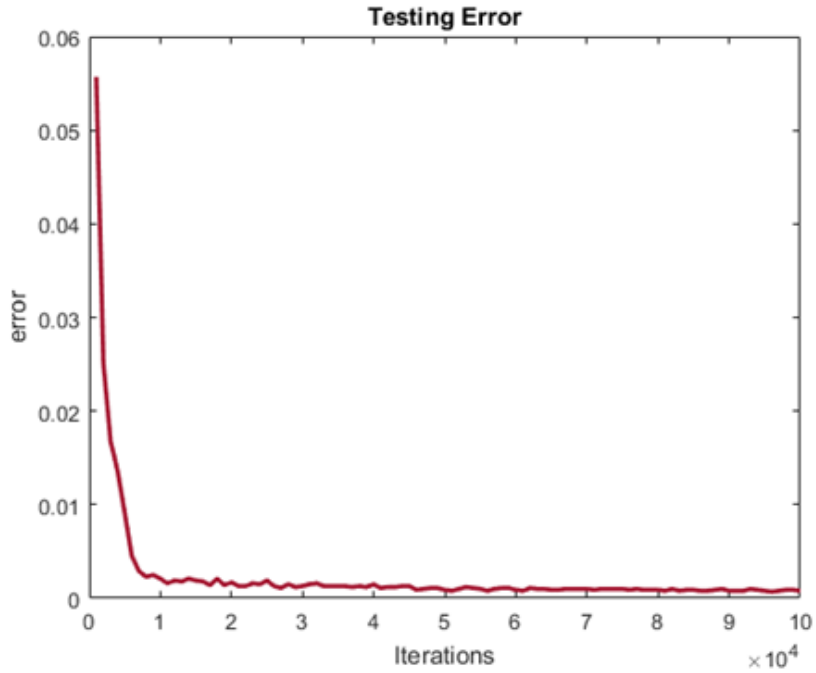
As shown in Table 2.1 and Table 2.2, the proposed CNN structure significantly outperforms the other methods. We also tested weakly sharpening when σ=1.0, λ=0.5 on CNN and EPTC. The results on two datasets are shown in table 2.3 Even on such worse case of EPTC as σ=1.0, λ=0.5, the performance of CNN can achieve 99.91% on BOSS and 98.89% on UCID& NRCS, which are 10.09% and 13.12% higher than the performance of EPTC, respectively. The training loss and testing error on the worst case as σ=1.0 and λ=0.5 are plotted in Figure 2.7. These two plotted curves show that the convergence of results occurred in 20,000 iterations.

Table 2.1 Detection Accuracy on BOSS Datasets

Parameters of USM	**CNN**	**EPTC [18]**	**EPBC [14]**	**Wang *et al.* [70]**
$\sigma = 0.7, \lambda = 1.0$	99.96%	93.22%	92.00%	98.37%
$\sigma = 1.0, \lambda = 0.8$	99.95%	93.95%	91.50%	99.67%
$\sigma = 1.0, \lambda = 1.0$	99.98%	95.26%	92.75%	98.50%
$\sigma = 1.0, \lambda = 1.3$	99.98%	96.53%	93.67%	99.83%
$\sigma = 1.0, \lambda = 1.5$	99.96%	97.10%	95.50%	99.67%
$\sigma = 1.3, \lambda = 1.0$	99.98%	95.96%	92.83%	99.17%
$\sigma = 1.3, \lambda = 1.5$	99.97%	97.48%	92.00%	99.83%
$\sigma = 1.5, \lambda = 1.0$	99.98%	96.23%	94.33%	98.17%



(a) Training loss



(b) Testing error

Figure 2.7 Training loss and testing error when $\sigma = 1.0, \lambda = 0.5$.

Table 2.2 Detection Accuracy on UCID&NRCS Datasets

Parameters of USM	CNN	EPTC [18]	EPBC [14]	Wang *et al.* [70]
$\sigma = 0.7, \lambda = 1.0$	99.53%	88.73%	90.72%	94.83%
$\sigma = 1.0, \lambda = 0.8$	99.50%	91.00%	90.00%	94.92%
$\sigma = 1.0, \lambda = 1.0$	99.65%	92.87%	90.07%	96.67%
$\sigma = 1.0, \lambda = 1.3$	99.81%	94.92%	93.71%	98.00%
$\sigma = 1.0, \lambda = 1.5$	99.86%	95.32%	95.41%	98.75%
$\sigma = 1.3, \lambda = 1.0$	99.75%	94.15%	91.17%	96.42%
$\sigma = 1.3, \lambda = 1.5$	99.89%	95.67%	95.55%	98.92%
$\sigma = 1.5, \lambda = 1.0$	99.78%	94.23%	94.13%	96.75%

Table 2.3 Detection Accuracy When $\sigma = 1, \lambda = 0.5$

Datasets	CNN	EPTC [18]
BOSS	99.91%	89.82%
UCID&NRCS	98.89%	85.73%

In addition, we have also demonstrated the rationality for choosing 4 layer-groups. It is noted that the number of layer-groups and the testing performance are not in a proportional relation that the more the better. The specific working means of adding layer-groups is to add similar convolutional modules, which always contain four layers (convolutional layer, normalization layer, activation layer and pooling layer), described in Subsection 2.2.2. The principle is that the last pooling layer should transform the feature maps in any size to a single element. To find a proper number of layer-groups in CNN designing, 4, 5, 6 and 7 layer-groups are compared, respectively. It appears that the worst case occurred when $\sigma = 1$ and $\lambda = 0.5$. Our experiments are shown in Table 2.4.

Table 2.4 Comparison of Different Number of Layer-groups When $\sigma = 1, \lambda = 0.5$

Group number	4	5	6	7
Accuracy (%)	99.16%	96.85%	98.35%	97.35%

Table 2.4 reveals that using 4 layer-groups has the best performance. Furthermore with the increasing of the number layer-groups, the overfitting to image contents by

CNN training becomes much more serious. As a result, 4 layer-groups is applied on the proposed CNN structure.

USM adds a ratio of high pass regions, such as the edge of image, to original image. Hence, the high pass regions of digital image become more obvious. In this situation, choosing max pooling instead of average pooling for down sampling is considered as a more proper choice. The comparison of performance in experiments also fully proved this consideration. The result on the worst case, when $\sigma = 1$ and $\lambda = 0.5$, is listed in Table 2.5.

Table 2.5 Comparison Between Max Pooling and Average Pooling

Type of pooling layer	Max pooling	Average pooling
Accuracy (%)	98.15%	93.35%

As shown in Table 2.5, the performance of Max pooling is almost 5% higher than using average pooling. Therefore, the obvious improvement of using Max pooling is testified.

CHAPTER 3

DIGITAL FORENSICS FOR RECOLORING VIA CONVOLUTIONAL NEURAL NETWORK

3.1 Introduction

As a common medium in our daily life, images are important for most people to gather information. There are also people who edit or even tamper images to deliberately deliver false information under different purposes. Thus, in digital forensics, it is necessary to understand the manipulating history of images. That requires to verify all possible manipulations applied to images. Among all the image editing manipulations, recoloring is widely used to adjust or repaint the colors in images. The color information is an important visual information that image can deliver. Thus, it is necessary to guarantee the correctness of color in digital forensics. On the other hand, many image retouching or editing applications or software are equipped with recoloring function. This enables ordinary people without expertise of image processing to apply recoloring for images. Hence, in order to secure the color information of images, in this chapter, a recoloring detection method is proposed. The method is based on convolutional neural network which is quite popular in recent years. Unlike the traditional linear classifier, the proposed method can be employed for binary classification as well as multiple labels classification. The classification performance of different structure for the proposed architecture is also investigated in this chapter.

3.2 Image Recoloring

Nowadays, people are willing to use graphics editor, such as Photoshop, to retouch their photos. Actually, a large proportion of pictures processed by common graphics editors were implemented by image recoloring algorithms. Although the final

impressions of these pictures varied in many different categories of visual effects, the manipulation of image recoloring can be generally divided into the operation based on three elements: Hue, Saturation and Luminance.

Among image recoloring algorithms, color transfer is one of the most typical one, which belongs to hue transfer. In color transferring process, the color space of image will be transferred from RGB to others, such as $L\alpha\beta$. The specific steps of this algorithm are shown below.

Step 1. Give a target image and a source image, obtain R, G, B of them.

Step 2. Convert RGB to $L\alpha\beta$ space. The transfer rule is shown below.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{3.1}$$

$$L^* = 116 f(\frac{Y}{1.0}) - 16 \tag{3.2}$$

$$\alpha^* = 500 \left[f(\tfrac{X}{0.9502456}) - f(\tfrac{Y}{1.0}) \right] \tag{3.3}$$

$$\beta^* = 200[f(\frac{Y}{1.0}) - \frac{Z}{1.088754}] \tag{3.4}$$

$$f(t) = \begin{cases} t^{\frac{1}{3}}, & \text{if } t > (\frac{6}{29})^3 \\ \frac{1}{3}(\frac{6}{29})^2 t + \frac{4}{29}, & \text{otherwise} \end{cases} \tag{3.5}$$

Step 3. Calculate transferred $L\alpha\beta$. The formula is shown as follow.

$$l_k = \frac{\sigma_t^k}{\sigma_s^k}(S^k - mean(S^k)) + mean(T^k) \quad k = (l, \alpha, \beta) \tag{3.6}$$

where $l_k, \sigma_t^k, \sigma_s^k, S^k, T^k$ stand for the transferred $L\alpha\beta$, the variance of k on target image, the variance of k on source image, the value of k on source image, the value of k on target image, respectively.

Step 4. Convert transferred $L\alpha\beta$ back to RGB space. The transfer rule is shown below.

$$Y = 1.0 f^{-1}(\frac{1}{116}(L^* + 16)) \tag{3.7}$$

$$X = 0.950456 f^{-1}(\frac{1}{116}(L^* + 16) + \frac{1}{500}\alpha^*) \tag{3.8}$$

$$Z = 1.088754 f^{-1}(\frac{1}{116}(L^* + 16) - \frac{1}{200}\beta^*) \tag{3.9}$$

$$f^{-1}(t) = \begin{cases} t^3, & \text{if } t > \frac{6}{29} \\ 3(\frac{6}{29})^2(t - \frac{4}{29}), & \text{otherwise} \end{cases} \tag{3.10}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 3.240479 & -1.5371500 & 0.498535 \\ -0.969256 & 1.875992 & 0.041556 \\ 0.055648 & -0.204043 & 1.057311 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{3.11}$$

Figure 3.1 shows a sample of color transfer. Giving a source image and a target image, the color transfer algorithm can change the hue style of target image closing to source image. As shown in Figure 3.1, The target image is bright-coloured because of the vivid color of leaves initially. However, the color of leaves was transferred from orange to green though color transfer process resulting in a new recolored image which has cold hue style visually. Although the visual effect was changed, the recolored image still seems natural. In other word, it is hard to be exactly recognized as a recolored image, which has been manipulated by color transfer algorithm, though the subjective judgment of human. As a result, the study of image forensics on recolored image is meaningful. In our experiment, we used a popular type of color transfer named Aibao, which was created by a Chinese photographer in 2008. The effect of Aibao is shown on the 2nd row of Figure 3.2. It is a tone closing to cyan-blue.

Nonlinear mapping is another common method of image recoloring. It can be also regarded as a color transferring without sample target. In this case, considering the RGB channel, all pixels in original images are non-linearly converted. Typically, it is used to change the tone of given image such as creating a warm tone. The red components are largely amplified especially for the pixels with less red components to bring a warm view to human eyes. The 3rd row of Figure 3.2 is an example of warming images. Besides Aibao and warming, there are also three different styles of image recoloring, which were used in our experiment. Their descriptions were listed below.

High-dynamic range (HDR). High dynamic range is an image format that can expand the range of brightness level. In image editing tools, a filter that can produce a similar effect is named as HDR filter. HDR filter expands the luminance difference of image, making the bright area much brighter and the dark area much darker. Three samples of HDR images are shown at the 2nd row of Figure 3.3. Compared with the original images at 1st row, the differences are visually obvious on first two images but not so obvious on the last image since the major areas on the last image are moderate-brightness.

Retro. Retro filter turns pictures to old photo look. Three samples of Retro images are shown at the 3rd row of Figure 3.3. Compared with original images, the overall hue of Retro images is partial to yellow, which presents a sense of age.

Post youth. Post youth filter also turns pictures to old photo look. However, Post youth images are much brighter than Retro images. Three samples of Post youth image are shown at the 4th row of Figure 3.3. These images look sprightly, presenting a sense that filled with youth memories.

(a) (b) (c)

Figure 3.1 Sample of color transfer. (a)Source image (b)Target image (c)Recolored image.
Source: [59]

Figure 3.2 The comparison of original images, Aibao images, Warming images. 1st row: original images. 2nd row: Aibao images. 3rd row: Warming images.
Source: [59]

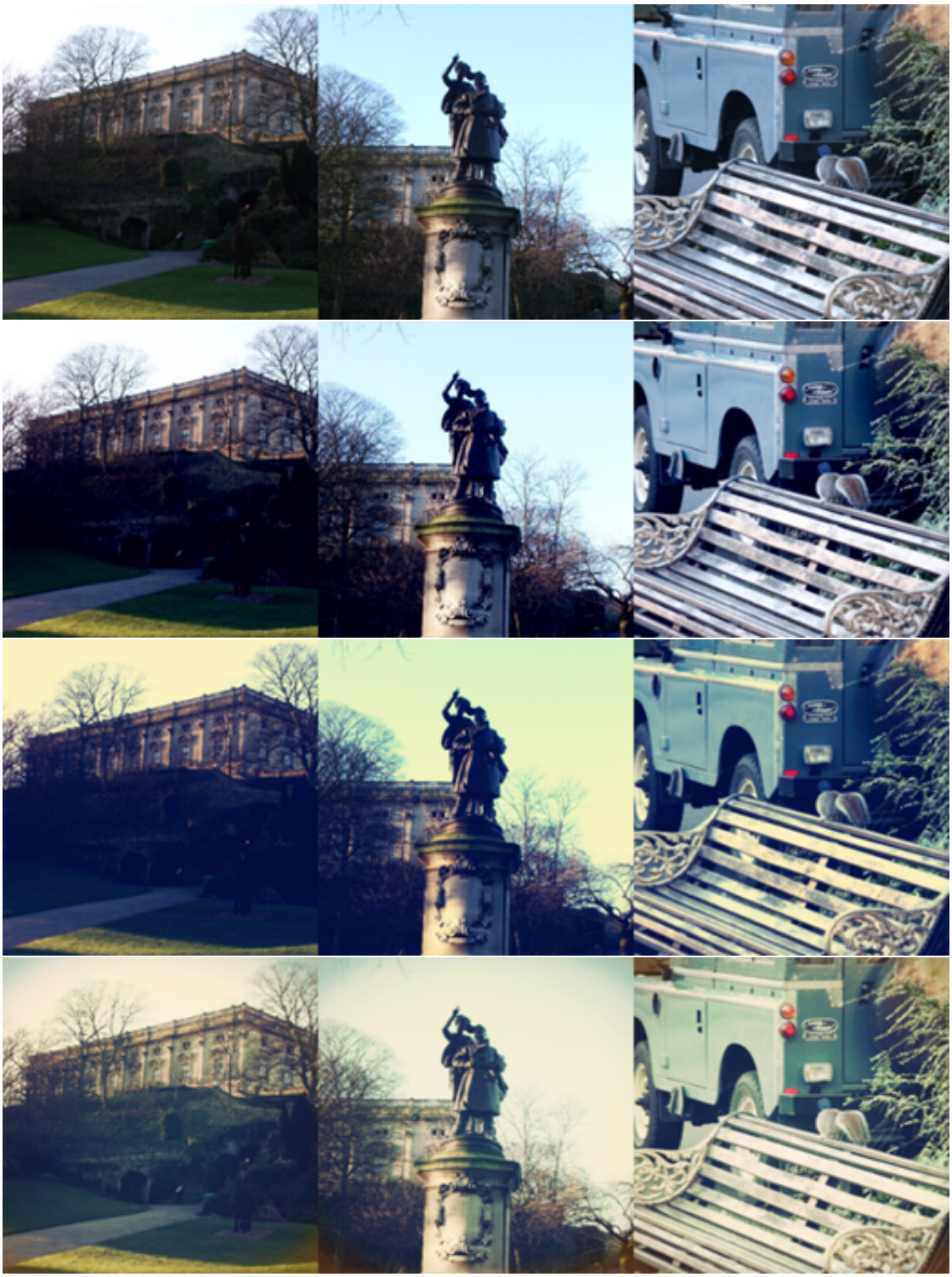


Figure 3.3 The comparison of original images, HDR images, Retro images, Post youth images. 1st row: original images. 2nd row: HDR images. 3rd row: Retro images. 4th row: Post youth images.
Source: [59]

3.3 Proposed CNN Structure

The proposed method is based on CNN. The CNN structure, motivated by "LeNet-5", is drawn in Figure 3.4. It is composed with four convolutional modules and one classification modules.

UCID image dataset [59] used for our experiments. All the images are cropped into the size of 384×384. As a result, the parameters in our proposed CNN are all optimized to fit this size. The input layer are the original images and the images that have been processed by recoloring. Four convolutional modules are set behind the input layers, displayed as "Layer Group 1", "Layer Group 2", "Layer Group 3" as well as "Layer Group 4" in Figure 3.4. Each layer group contains three layers begin with convolutional layer to filter the input feature maps. In first layer group, the input feature map is in size of 1× (384×384) and it goes through 8 filters in size of 1× (3×3). In following modules, there are 16 filters of size 8× (3×3) in layer group 2, 32 filters of size 16× (3×3) in layer group 3 and 64 filters of size 32× (3×3) in layer group 4, respectively. Then, the feature maps generated from convolutional module will get into activation layers, which has the ability to optimize the statistical model. ReLU is chosen as the activation function for all layer groups. At the end of each layer group, pooling layer processes down sampling to feature maps. In first three layer groups, max pooling layer in size of 5×5 with stride 2 is performed to get the local maximum. The size of feature maps will be reduced by $\frac{3}{4}$ through each max pooling. As for the last layer group, average pooling in size of 48×48 is hired to make sure the output are 64 feature vectors. Finally, 64 feature vectors go through classification module and the final probabilities comes out. These probabilities represent the accuracy of filters in common image editing software detection achieved by this CNN structure. By the way, fully-connected layer contains two hidden layers in our proposed designing.

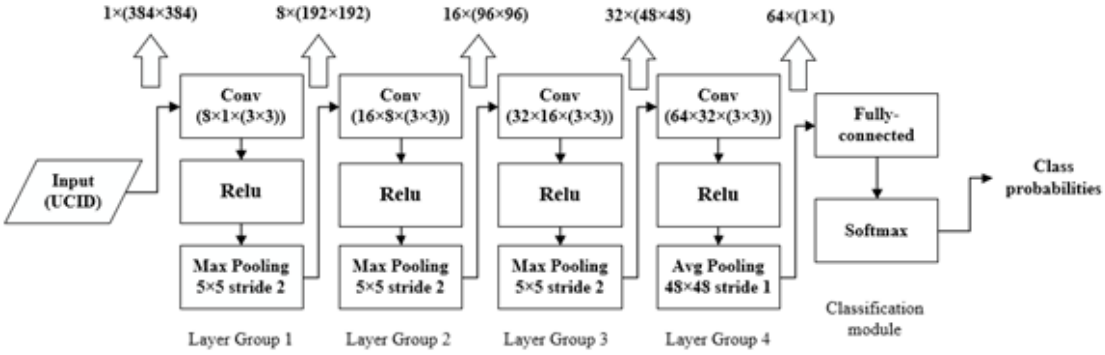


Figure 3.4 The structure of proposed CNN. The configuration of each layer is displayed inside the boxes. The sizes of feature maps are listed on the top, shown as number of feature maps × (height× width). The sizes of convolutional filter groups are shown in the boxes follows number of filters × number of input feature maps × (height× width).

3.4 Experiment Results

3.4.1 Datasets

UCID [59] was employed as the image dataset in our experimental. It contains 1,338 uncompressed color images with size of 384×512 or 512×384 in "tiff" format. For convenience purposes, all the images were cropped into the size of 384×384. In addition, consider of that "tiff" is not a mainstream image format at present, all the images were converted to "png" format. Overall, the experimental image database contains totally 1,338 uncompressed colorful images with size of 384×384 in "png" format.

3.4.2 Platform and Settings

TensorFlow, the most popular deep learning framework nowadays, is selected to compose our CNN structure. The open source feature and high expansibility of TensorFlow hastened the network building. The version number of the TensorFlow for this experiment is 1.11.0. All the experiment codes were implemented on Spyder 3.3.1, which is a common python development environment. Adam optimizer, as the most common optimizer in TensorFlow using, was applied to train the whole network.

Two graphics cards were employed for training process. The model of the graphics cards is NVIDIA GeForce GTX 1080Ti with 10GB memory. The training batch size was set to 64, meaning that for each iteration, 64 images will get into the network. The training iteration was fixed as 5000 for two-category classification and 20000 for six-category classification. One epoch means all the images in training dataset get into the network once. So, for two-category classification, the total number of epochs is 152. As for four-category classification, the total number of epochs is 200.

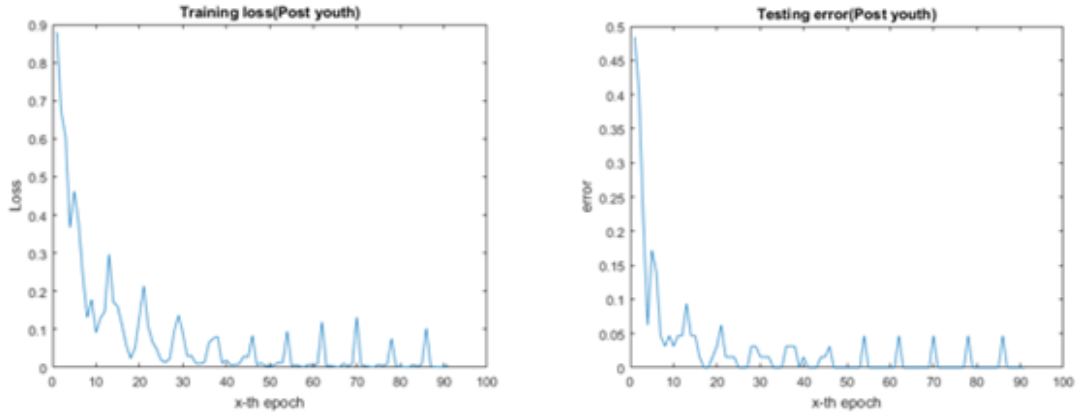
3.4.3 Results

First, five training processes of two-category classification, which were HDR images with original images, retro images with original images, post youth images with original images, aibao images with original images as well as warming images with original images, were performed individually. Then, a model to distinguish all the six styles of images was trained. The classification accuracy and the consumed epochs for convergence are recorded in Table 3.1.

Table 3.1 The Performance of the Proposed Method Towards the Recoloring Algorithms

Cases	HDR	Retro	Post youth	Aibao	Warming	All
Accuracy	100%	100%	100%	100%	93.75%	96.88%
Epoch	34	6	19	46	68	175

As shown in Table 3.1, the proposed CNN emerged its overwhelming ability to recognize the images not only between recolored images and original images, but also between different styles of recolored images. In addition, the curves of the training losses and the test errors on Post youth case and all classes case were plotted on Figure 3.5. As shown in the figure, the convergence of two-category classification occurred in 25 epochs and the convergence of six-category classification occurred in 60 epochs. However, there are some small fluctuations after convergence occurred.



(a) (b)

(c) (d)

Figure 3.5 (a) Training loss of two-category classification (Post youth and original).
(b) Testing error of two-category classification (Post youth and original).
(c) Training loss of six-category classification (All six styles).
(d) Testing error of six-category classification (All six styles).

In our proposed CNN, there are totally 12 layers (4-layer groups) in convolutional modules, which is a comparatively shallow CNN. In order to find out the most suitable depth of the network, we compared our 4-layer groups structure with 3-layer groups, 5-layer groups as well as 6-layer groups on two-category classification and six-category classification, respectively. The results of two-category classification, which is HDR images against original images, is shown in Table 3.2. As shown in the table, all the depths of the network can achieve 100% in this two-category classification case. 5-layer groups and 6-layer groups is even faster than 4-layer groups to achieve the best accuracy. However, it doesn't mean that 5 or 6 layer groups will be a better choice on detecting recolored images. Table 3.3recorded the results of six-category classification case. In Table 3.3, 4-layer groups outperformed other depths on detecting accuracy. As a result, although deeper network structures can speed up convergence of training process, they are easier to suffer from overfitting with the increasing of data size and then have a performance degradation. This comparison proved that 4 layer groups is still the best choice on image recoloring detection.

Table 3.2 TThe Comparison of the Proposed Method with Different Depths for Binary Classification of HDR and Original Image

Depths	3 layer groups	4 layer groups	5 layer groups	6 layer groups
Accuracy	100%	100%	100%	100%
Epoch	74	34	17	18

Table 3.3 The Comparison of the Proposed method with Different Depths Towards the Classification of All Recoloring Algorithms

Depths	3 layer groups	4 layer groups	5 layer groups	6 layer groups
Accuracy	93.75%	96.88%	92.19%	93.75%
Epoch	72	175	38	61

Xu *et al.* [72] used TanH to replace ReLU as activation layer after first two convolutional layers and this strategy lead to a immense improvement for their method. However, all of the activation functions in our proposed CNN structure are ReLU. We tried two other combinations of activation layers, which are replacing all ReLU with TanH and replacing all ReLU with TanH except the last layer group, also on two-category classification and six-category classification, respectively. The Warming as recoloring algorithm, which is the most challenging problem other than the other methods, is chosen as the subject of two-category classification for our experiment. The classification results are shown in Table 3.4. Although all three kinds of combinations has the similar performance on two-category classification, the combination of all ReLU, which is adopted in our proposed CNN structure, can achieve better accuracy on six-category classification.

Table 3.4 The Comparison of Different Combinations of Activation Layers

Cases	two-category classification			six-category classification		
Combinations	All ReLU	All TanH	3 TanH+1 ReLU	All Relu	All TanH	3 TanH+1 ReLU
Accuracy	93.75%	93.75%	93.75%	96.88%	93.75%	93.75%
Epoch	68	143	11	175	95	160

CHAPTER 4

ANTI-FORENSICS OF IMAGE SHARPENING USING GENERATIVE ADVERSARIAL NETWORK

4.1 Introduction

Image sharpening is an image enhancement method which has been widely used to improve the quality of images. Therefore, in image forensics, it is required to be identified as all possible manipulations applied in images need to be detected. In recent years, sharpening detection get evolved with new detectors proposed every year to gradually boost the detection performance. This situation continues for several years till the introduction of convolutional neural networks (CNNs). With the assistance of CNNs, the detection of sharpening seems to be completely solved that the detection performance for sharpening achieves perfect, even when the images are weakly sharpened. Is it true that we should no longer pay attention to sharpening forensics any more? To answer this question, in this chapter, an anti-forensics method based on generative adversarial network(GAN) is proposed to investigate the philosophy. The images generated via our method possess the feature of sharpening, however, they cannot be simply considered as sharpened images because no traditional sharpening manipulation is applied during the procedure. Observed from the experimental results, even the state-of-the-art sharpening detector based on CNN can be deceived with the GAN generated images.

4.2 Literature Review

The explosive development of internet makes the propagation and distribution of information easier than ever. Under this circumstance, it also enables the dissemination of false information which brings immense harm to the community. Therefore, it is necessary to justify the authenticity and integrity of information from all possible channels. Generally speaking, people prefer visualized information such as images and videos over information in other forms. Thus, the study of image forensics [23][50][20] is the guardian to protect people from all kinds of image attacks. The study about detecting USM sharpened images beginning in 2009. In 2009, Cao *et al.* found that there were aberrations in the histograms of sharpened images and proposed an algorithm to detect such aberration [9]. However, regarding to their report, this algorithm is not very effective when detecting the images without wide histogram. Then, Cao *et al.* revised their algorithm in order to improve the performance on images with narrow histogram and proposed a new detecting algorithm [10] in 2011. The algorithm employs a set of side-planar crosswise pixel sequences to locate on the basis of edge pixels of the detected image. Then, a set of overshoot strengths is calculated for each side-planar pixel sequence. The average of the overshoot strengths measures the overshoot metric of the whole image. Finally, the detected image will be identified refer to which interval this average overshoot strength belongs to.

After solving the problem about detecting images with narrow histogram, another weakness of their algorithm has been found, that is, the performance is limited when detecting images with JPEG compression. This drawback limits its generality use in practical applications. However, Ding *et al.* [17] proposed an novel algorithm based on local binary pattern (LBP) [46][47] in 2013. The authors thought that the appearance of overshoot artifacts can be regarded as a special kind of texture modification. Meanwhile, LBP is a widely used texture classification technique. As a result, the performance of the LBP-based algorithm exceeds all the sharpening

detection algorithm before. However, after that, Lu *et al.* [38] proposed a method to remove overshoot artifacts for anti-forensics of USM sharpening.

Then, inspired by the LBP-based method, Ding *et al.* [18] proposed a much more effective algorithm to detect USM sharpening. The algorithm is called Edge Perpendicular Binary Coding (EPBC). Since that the texture modification generated by USM sharpening is mainly along the perpendicular direction of image edges, EPBC employs a long rectangular window, which is perpendicular to the edges of images, to extract features of image textures and uses a binary coding strategy to reduce the size of feature sets. Furthermore, an improved algorithm, which is Edge Perpendicular Ternary Coding (EPTC), is proposed in [16]. EPTC replaced the binary coding with ternary coding in EPBC, which outperformed the EPBC.

The sharpening forensics was further improved later since CNN was introduced. The detection scheme based on CNN came out in 2018. Ye *et al.* [73] proposed an advanced CNN architecture that contains four convolutional modules with four layers each. By using max pooling as the pooling function and 'Relu' as the activation function, the results of this paper showed that the detection accuracy on all the cases were over 98% on the CNN model they trained. And this invest represents the state-of-art on image sharpening detection at present.

Since the forensics on sharpening detection has achieved tremendous success, an anti-forensics sharpening algorithm via GAN is proposed in this chapter to deeply challenge the current state-of-art.

4.3 Pix2pix

Pix2pix [32] is a conditional adversarial network structure, which is proposed by Berkeley AI Research (BAIR) Laboratory. Compared with classical GAN, it provides a solution of image-to-image translations. In other words, pix2pix accepts image pairs as input. One image pair contains an input image and a target image. Pix2pix

will learn the regular pattern of translation from input image to target image, then perform the translation on input image based on the pattern it learned and generate an output image. Considering that image sharpening algorithm only enhances the visual effect of image but not tampers with content, pix2pix is selected to implement similar treatment with USM sharpening algorithm.

4.3.1 The Network Architecture

The architecture of pix2pix is shown in Figure 4.1. The discriminator learns to judge the generated image as unsharpened image and the target image as sharpened image. Meanwhile, the generator learns to deceive the discriminator by adjust the output image it generated. With continuously training of the network, the generated image will be closer and closer to the target image and also has better performance on the resist of detection.

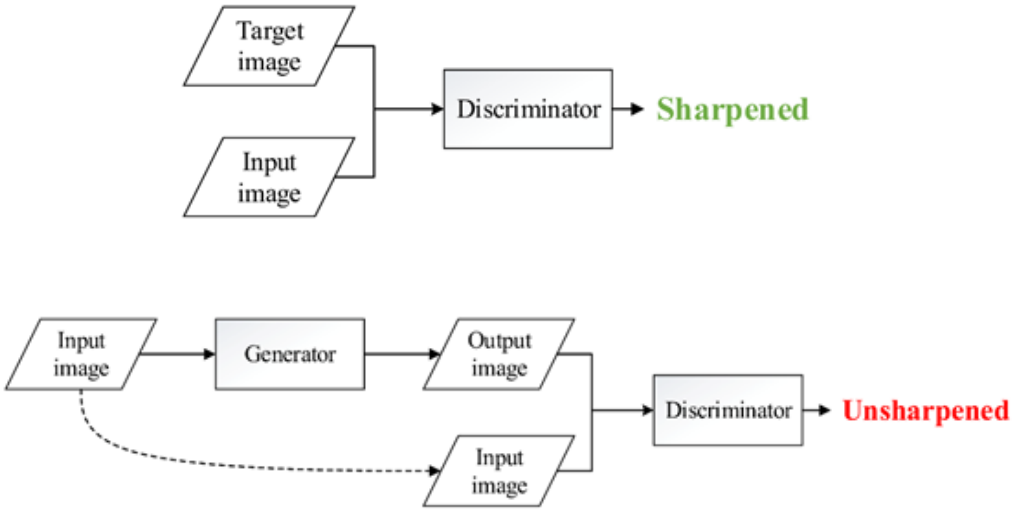


Figure 4.1 The architecture of pix2pix.

4.3.2 Generator and Discriminator

The generator in pix2pix is "U-Net" [56], which follows the rule of skip connection. The authors considered that for image translation, the input and the output should share some information from the layers on the bottom of the network. The feature maps in the first half layers from input to output will be added to the symmetrical layers in the second half. As a result, some features on the bottom will be preserved as the reference to output image.

Image generated via $L1$ or $L2$ loss is fuzzy because $L1$ and $L2$ is not effective to restore the high frequency part of image. In order to overcome this draw back, pix2pix provides a discriminator structure called PatchGAN. First, PatchGAN slices image to patches. Then, it tried to classify each patch, respectively. Finally, it averages the results of all the patches and made its decision. This method can avoid the loss of texture to some extent.

4.4 Experimental Results

4.4.1 Datasets

Two image databases were utilized in the experimental. One is named Boss, which was designed for steganoagraphy and steganalysis. Boss contains 10,000 uncompressed grayscale images in "pgm" format. Another image database, which named UCID&NRCS by us, was consisted of 1,000 images from the UCID image database and 1,000 images from NRCS image database. For convenience purposes, all the images from Boss were scaled to the size of 256×256. Since the images of UCID&NRCS are not square, all the images from UCID&NRCS were cropped into the size of 384×384 first and then scaled to the size of 256×256 as well. In addition, all the images were converted to "png" format and grayscale images.

Note that, in section 2, we have introduced two parameters, which were λ and σ. They determine the effect of USM sharpening. In our experiment, we have

worked on five combinations of λ and σ, i.e., 'λ=1.5, σ=1.3', 'λ=1.0, σ=1.5', 'λ=1.0, σ=1.3', 'λ=1.0, σ=0.7', 'λ=1.5, σ=1.0'. Two image databases that introduced before were sharpened based on these five cases. The original image and its sharpened image, which became a pair of images, were labeled as 0 and 1, respectively. 10,000 Pairs of images generated by Boss were used for training pix2pix model, which can transfer images, and CNN model, which can distinguish images. 2,000 Pairs of images generated by UCID&NRCS were used for testing the performance of generated models.

4.4.2 Platform and Settings

TensorFlow is one of the most popular deep learning frameworks at present. It has open source feature as well as a large total of convenient APIs. All the pix2pix architecture and CNN architecture were implemented by TensorFlow. The version number of the TensorFlow for this experiment is 1.11.0. The graphics cards employed in the experiment were two NVIDIA GeForce GTX 1080Ti with 10GB memory. The version number of CUDA is 9.0.

For training pix2pix models, the training batch size was fixed to 1, meaning that for each iteration, 1 image will get into the network. One epoch means all the images in training dataset get into the network once. The number of epochs for training was fixed to 100. Adam optimizer was employed to train the whole network. The initial learning rate of Adam was fixed to 0.0002 and the momentum was fixed to 0.5. The images generated by pix2pix would be classified by the CNN model, which trained though the CNN architecture in [73].

The CNN architecture is shown in Figure 4.2. It has four convolutional modules. Each convolutional module has a convolutional layer, a normalization layer, a activation layer and a pooling layer, respectively. The function of activation layers is ReLU. Max pooling is adopted in first three convolutional modules. The pooling

algorithm in the last convolutional module is average pooling, which concluded the feature maps to 64 single feature elements before getting into classification module. For training CNN models, the batch size is 64 and the number of epochs is 50. The learning rate of Adam is 0.001 and the momentum is 0.9.

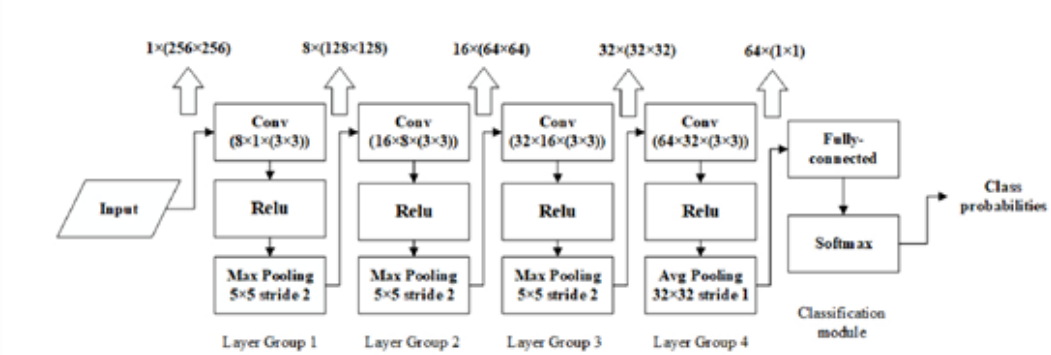


Figure 4.2 The architecture of CNN for testing. The configuration of each layer is displayed inside the boxes. The sizes of feature maps are listed on the top, shown as number of feature maps × (height× width). The sizes of convolutional filter groups are shown in the boxes follows number of filters× number of input feature maps × (height× width).

4.4.3 Results

At first, we used the prepared 10,000 pairs of images, which were generated from Boss, of each five cases to train pix2pix models individually. Second, we used these models to generate images on UCID&NRCS. The samples of the comparison between the generated images, the original images and the sharpened images on the case when 'λ=1.5, σ=1.3' were shown in Figure 4.3. Compared with original images, the generated images do have similar sharpening effect with generated images visually.

Then, we still used the image pairs of Boss to train CNN models of each five cases. Finally, 2,000 generated images of each cases were validated by corresponding models. The comparison of the average precision of generated images and the duration of training pix2pix models for each case are shown in Table 4.1. The time consumption indicates the time count on minutes that consumed for the model reach convergence.

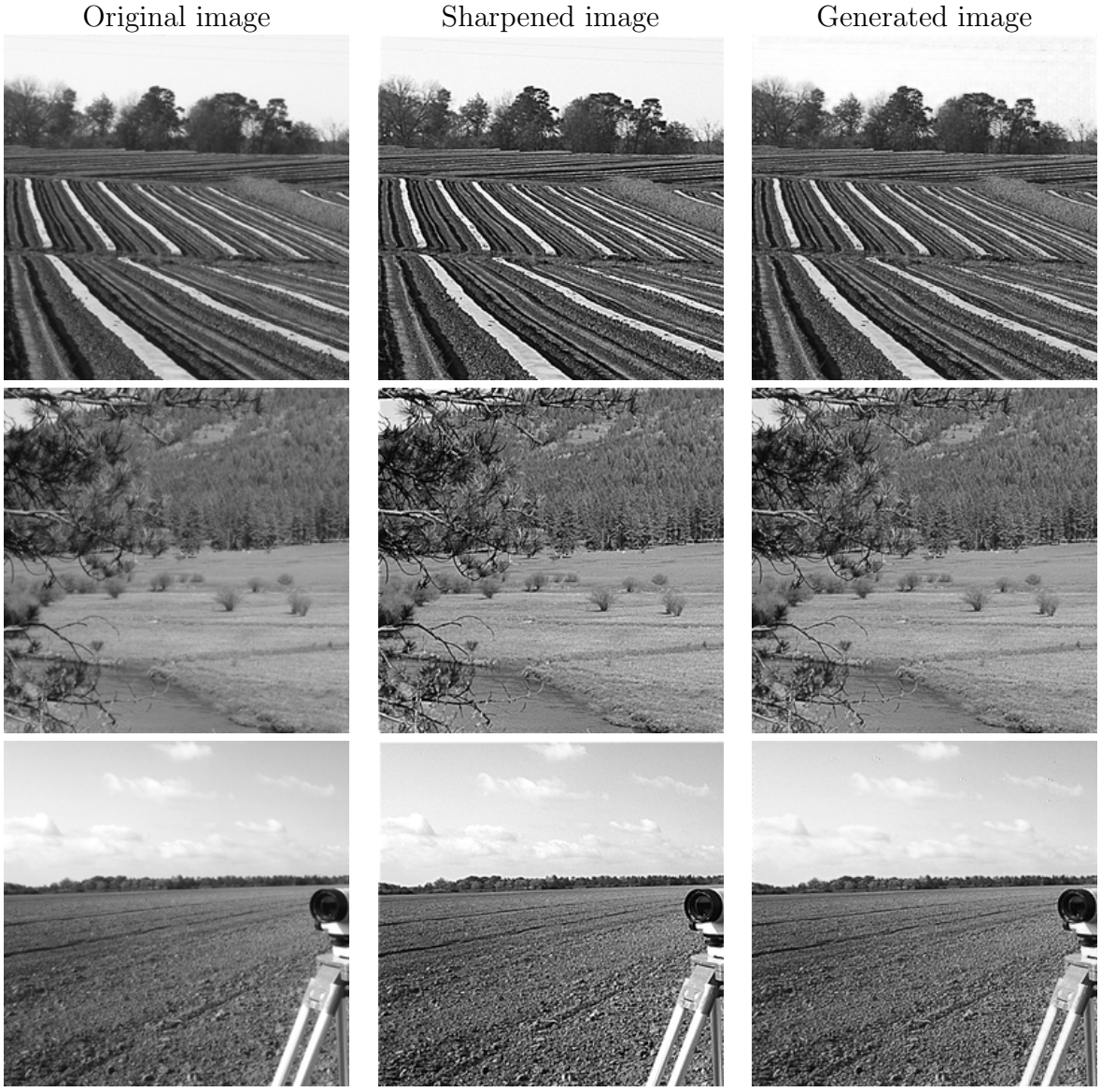


Figure 4.3 The samples of the generated images, the original images and the sharpened images on the case of $'\lambda = 1.5, \sigma = 1.3'$.
Source: [59]

Table 4.1 The Average Precision for Each Cases and the Duration of Pix2pix Training Process

Cases	Precision	Time consumption
$\lambda = 1.5, \sigma = 1.3$	12.28%	1259
$\lambda = 1.0, \sigma = 1.5$	16.28%	1267
$\lambda = 1.0, \sigma = 1.3$	21.79%	1260
$\lambda = 1.0, \sigma = 0.7$	26.68%	1255
$\lambda = 1.5, \sigma = 1.0$	14.38%	1260

Table 4.2 The Average PSNR of the Generated Images and the Sharpened Images on Each Cases

Cases	PSNR(dB)	
	Generated	Sharpened
$\lambda = 1.5, \sigma = 1.3$	30.74	26.00
$\lambda = 1.0, \sigma = 1.5$	33.40	28.02
$\lambda = 1.0, \sigma = 1.3$	33.11	28.94
$\lambda = 1.0, \sigma = 0.7$	35.73	34.64
$\lambda = 1.5, \sigma = 1.0$	31.44	27.94

Considering that the classification accuracy for all the cases are over 98% in [73], the generated images does have anti-forensics property to some extent. Besides, the average value of peak signal to noise ratio (PSNR) for the generated images and the sharpened images of different parameters are computed. The results can be found in Table 4.2. It is the evidence that the generated images not only possess the sharpening effect, but also higher image quality when compares with the images processed by USM sharpening algorithm.

CHAPTER 5

IMAGE ANTI-FORENSICS USING EXTRA SUPERVISED GENERATIVE ADVERSARIAL NETWORK

5.1 Introduction

Where there is sunshine, there is also shadow. For images, there are many manipulations that can be used to attack images in different ways; nevertheless, there are also numerous forensics tools designed to defend images from all possible attacks [50][44]. Many digital forensics researchers are dedicated to creating algorithms [68][18] to secure images as reliable channels for people to communicate accurate information. In the past, researchers have built mathematical models [10][69] to trace the alteration of image statistics. In addition, many forensics tools have been developed based on designing handcrafted features to be classified by linear classifiers for a variety of forensics purposes such as identifying source devices [11], detecting manipulations [16][30], and exposing forgeries [22].

In recent years, deep learning has made colossal progress. As the most well-known neural network, convolutional neural networks (CNNs) [7][54] and recurrent neural networks (RNNs) [42][29] are widely applied in various research fields to analyze data in different forms. In image forensics, it is quite common to adopt CNNs as classifiers to perform detection tasks [11][3]. The feed-forward structure and learning ability enabled by backward propagation make neural networks ideal forensic detectors. CNNs can learn high-dimensional features that cannot be comprehended by the human brain. These features are highly efficient for detection. It has been shown in many publications that well-trained CNN models achieve remarkable detection performance against various image editing manipulations and thoroughly outperform traditional methods [1][15]. To the best of our knowledge,

nearly all possible image editing manipulations can be precisely identified by deep neural networks with properly labeled training [4].

Whereas deep learning has been justified frequently as the perfect image forensics tool, with the more sophisticated architectures developed in recent years, new challenges have also appeared. Unlike the most commonly proposed models focusing on classification, generative adversarial networks (GANs) [28][53] are designed for creation. GANs are composed of multiple neural networks. A typical GAN model consists of two neural networks: one network functions as a discriminator, and the other network serves as a generator. Both the discriminator and generator simultaneously learn during training to enhance their designated ability to compete with each other in a game. In most cases, after training, GANs are capable of generating images that are similar to the input samples. Note that 'similar' here can be measured in many different ways. For instance, it could be objects of a homogeneous category, the same species, shapes with identical textures and colors, and analogous styles.

Because the images are generated by GANs without any natural information, they could be used by actors with malicious purposes to deliberately deliver false information. For example, GANs can generate images of a person in a scene who does not exist [35]; GANs can also transform an image captured in daylight into an image with a night scene [75]. Usually, these images can easily deceive human eyes. In addition, it is impossible for humans to process tremendous amounts of images. Thus, we rely on forensics algorithms to verify the authenticity of images. If these GAN-synthesized images are capable of defeating existing forensics tools, they may become huge threats to our community. In addition, most anti-forensics algorithms proposed in the past rely on specialists with expertise to build corresponding anti-forensics models for different manipulations. However, unlike traditional approaches, this process is now significantly simplified in that ordinary

people without any professional training can easily build their own attacks based on GANs by collecting the proper data. This makes GANs more dangerous than any anti-forensics methods previous.

Therefore, in this chapter, we would like to investigate the anti-forensicsability [25][63] of the GAN model to enlighten study on image forensics [6]. We propose our GAN model as a universal anti-forensics tool that features the removal of manipulated fingerprints in manipulated images. In other words, the proposed GAN model can generate images that are capable of subtly hiding traces of a variety of common image editing manipulations [71][65] without altering the original image contents. Generally, these fingerprints are widely employed by forensics detectors to identify manipulations. By removing these fingerprints, the generated images are assumed to impede the forensics tools to make incorrect judgments.

To summarize the above, the main contributions of this investigation are as follows:

1) A new problem is raised. Unlike traditional image attacking approaches, GANs can be easily trained as anti-forensics tools for many image manipulations. The current forensics detectors may compromise toward the images synthesized by GANs.

2) A GAN model is proposed as a universal anti-forensics tool to investigate the anti-forensicsability of GANs. Discussions are made to outline the prospect of image forensics with the development of deep learning and GANs.

3) Alternative GAN structures and generative networks are considered and studied to refine GANs to achieve greater antiforensicability and image quality.

4) Comparisons are made with prior works. Our proposed model outperforms other anti-forensics GAN models. It also reaches a trade-off between anti-forensicsability and image quality when compared with traditional anti-forensics methods.

5.2 Generative Adversarial Networks

A GAN is a concept defined by Ian Goodfellow *et al.* in 2014. It is actually a class of machine learning systems consisting of generative networks and discriminative networks. In this system, given training samples, both the generative network

and discriminative network are trained simultaneously for different purposes. The generative network generates new data to be evaluated by a discriminative network. The discriminative network is trained to discriminate the synthetic data from the training samples. Meanwhile, the generator learns from the discrimination procedure to generate new data with closer statistics to the training samples to fool the discriminator. Generally. this system can be regarded as a competition between two networks. Through back-propagation during training, both networks are optimized and become more intelligent. Typically, the new data synthesized by generators have attracted the most attention from researchers. They have been widely studied and applied in a variety of areas [58].

Many GAN models have been proposed, being driven by different motivations. Among all the GANs, the conditional GAN (cGAN) is a special category with fully supervised learning that concentrates on minimizing the process of setting the generating process conditions. Unlike many other GANs that only focus on generating vivid images, cGANs can generate vivid images with different characteristics. With proper supervision, cGANs are capable of delivering desired new data to satisfy different purposes. Because of this feature, cGAN is the preferred option to translate images from one style to another [75].

In image processing, many image editing manipulations leave unique traces in images producing particular visual effects. These visual effects can also be regarded as image styles. For example, sharpening can enhance the contrast of edges, which leads to sharp silhouettes as a visual feature. Taking a step further, theoretically, untouched images without any manipulations can also be considered as a style, i.e., a raw style. Therefore, given that cGANs can translate image styles, they are also assumed to be capable of transforming images from other styles into a raw style. In other words, the fingerprints left by a variety of manipulations can be removed by cGANs such that the image may appear untouched. In image forensics, such operations could lead to

the possibility of manipulations being applied to images becoming more difficult to detect. Thus, cGANs may serve as general anti-forensics tools.

5.3 Anti-forensics

Within digital forensics, anti-forensics, also known as counterforensics, is a branch that has raised much debate and discussion. Anti-forensics is a set of techniques that are used to combat digital forensics. Generally, anti-forensics tools are designed for malicious purposes. However, for scientific study, anti-forensics tools can also serve as countermeasures to forensics algorithms. By exposing the weaknesses of current forensics tools, anti-forensics helps researchers further develop powerful forensics tools for the future to guarantee that the collected data are authentic and dependable.

Anti-forensics falls into several subcategories such as data hiding, artifact wiping, and trail obfuscation. In this chapter, as discussed in the previous section, we examine the anti-forensicability of GANs from the perspective of artifact wiping to deceive forensics tools.

Although many anti-forensics works on erasing manipulated fingerprints have been reported, most of them focus on overcoming a single manipulation, that is, either JPEG compression or median filtering. JPEG compression is commonly applied to downscale an image, while a median filter is commonly used to remove noise from images while preserving edges. The two manipulations are ideal counter-forensics targets because limiting data size and denoising are fundamental needs for image processing.

Among all the JPEG anti-forensics works conducted by different groups, it is recognized by most forensics researchers that Stamm *et al.* made the greater contribution to this topic to date. They initiated related study in 2010 [65] and refined the JPEG anti-forensics methods in 2011 [64]. Their works were followed by other groups [48][52], where a variety of JPEG anti-forensics models were later proposed to

enhance the JPEG anti-forensics performance under different circumstances. The majority of related study hides the compression trails by building anti-forensics models to tamper the image statistics. A similar phenomenon also occurs in the history of median filter anti-forensics. After Fontani *et al.* started counterforensics study on the median filter in 2012 [21], this topic has made considerable progress, with more median filter anti-forensics models being proposed [71][62]. Most of them also achieve the anti-forensics effect by attacking the image statistics.

In addition to building anti-forensics models as described above, a few anti-forensics works based on adversarial networks have been proposed in recent years. By employing GANs for anti-forensics, researchers no longer need to analyze the image statistics or tamper with any specific fingerprints because GANs are capable of self-learning to achieve anti-forensics objectives automatically. With supervised training, GANs can synthesize images that preserve exactly the same content as the attacked images. In addition, the manipulated fingerprints, once employed as clues for forensics detectors, are removed in synthesized images, that is, the GANs can serve as anti-forensics tools. Kim *et al.* employed GANs to restore images processed by median filters [36]. The images reconstructed via their GAN proved to be able to outperform images processed by other anti-forensics methods with higher anti-forensicsability, as reported in their paper. Luo *et al.* [43] proposed a GAN model that can reach acceptable anti-forensibility compared with [64]. Although there is no doubt that both works are brilliant efforts involving new methods of GANs, they are similar to other anti-forensics works that have contributed to improving anti-forensics performance for single manipulations.

Additionally, another novel anti-forensics application of GANs needs to be mentioned here: attacks on camera model forensics. Chen *et al.* first proposed a GAN framework to falsify forensics information of camera models [12]. Consequently, it could prevent forensic detectors based on CNNs from making correct judgements.

Later, the same group refined the framework by introducing advanced structures [13]. Their latest study demonstrated that the proposed model in [13] can attack forensic detectors under both black-box and white-box scenarios.

For image manipulation anti-forensics, note that, other than the anti-forensicsability, image quality is the other benchmark for evaluation. In most cases, image quality must be sacrificed to enhance the anti-forensibility. Thus, most anti-forensics works have made strong efforts to achieve an a trade-off between anti-forensicsability and image quality. This is extremely important for our investigation, as successful anti-forensics tools should be capable of deceiving forensic detectors and humans simultaneously. In summary, in this chapter, we concentrate on investigating the anti-forensibility of GANs by proposing a GAN model that serves as a universal anti-forensics tool that targets multiple common image manipulations. The proposed method attempts to remove traces of different manipulations while avoiding distortions being introduced to the images. The entire procedure is depicted in Figure 5.1.

5.4 Proposed Method

5.4.1 Prototype GAN Model

As mentioned above, both the discriminator and generator are important components in GANs. One of the most typical and fundamental GAN models, the deep convolutional GAN (DCGAN), consists of a single classification network as the discriminator and a single generative network as the generator. This GAN generates new images from random noise. However, the image content and texture generated in the DCGAN cannot be well supervised. Thus, in most application scenarios of GANs, the structures have to be refined and optimized. Thus, to function as an anti-forensics tool, our proposed prototype model is designed as illustrated in Figure 5.2.

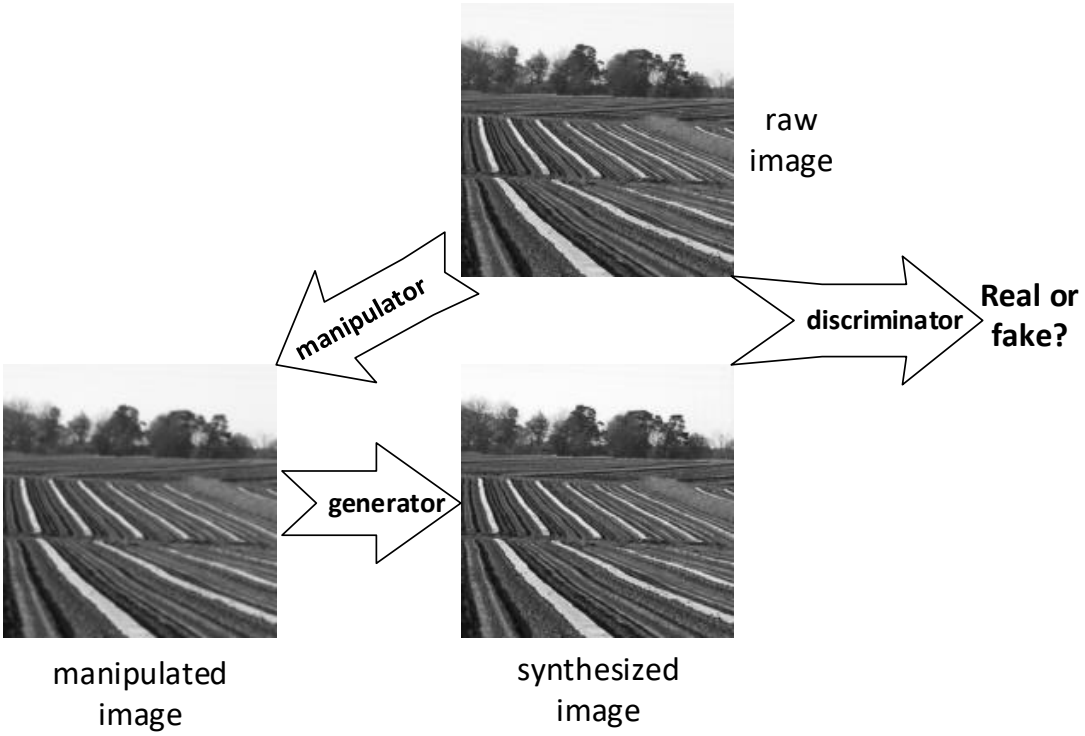


Figure 5.1 Training GAN models to remove traces left by image editing manipulations.

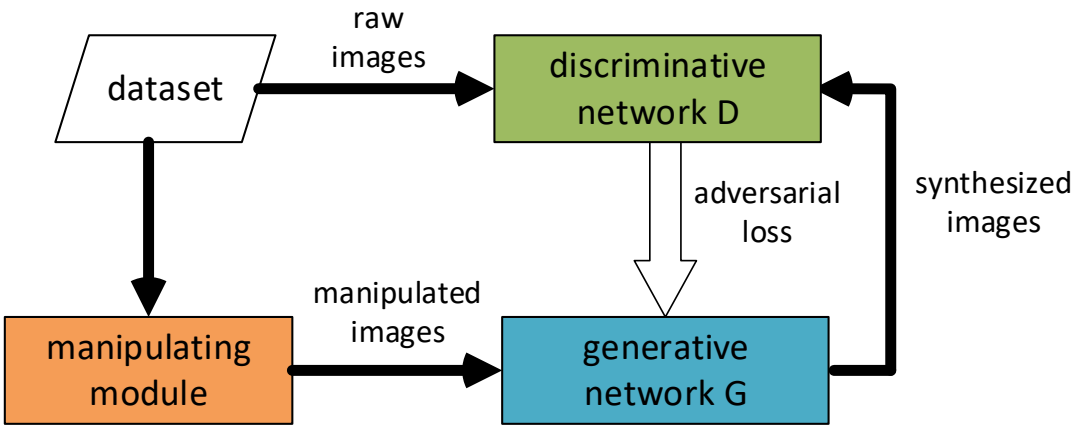


Figure 5.2 Architecture of the proposed prototype GAN model.

The only input to the GAN model is the raw image dataset. Since our objective is to remove the manipulated fingerprints while keeping the image content untouched, we would like to have the image content be generated under strict supervision. This objective can be satisfied by inputting paired images to enhance the supervision of the image content. Hence, there must be two parallel input channels for feeding image pairs into the generator as source signals and target signals.

As observed from Figure 5.1, the manipulating module in our model contains optional image manipulations that could be chosen to convert the raw images to manipulated images. The manipulated images are source signals for the generator to synthesize new images. The output of the generator can be employed in association with raw images to train the discriminator. In other words, the discriminator is trained to discriminate the raw images and synthesized images. Note that for anti-forensics, our objective is to produce images that can deceive forensic detectors. Hence, we expect that the images synthesized from the generator will be close to their raw version and that the discriminator fails to classify them. For this reason, the raw images here can be regarded as the target signals. As a result of this arrangement, the source signals and target signals share the same image content, and the generation procedure for the image content is fully supervised. This is an advantage for the proposed GAN structure in that it ensures that the synthesized images from the generator preserve the same content. Thus, we only need to focus on transferring the image style for anti-forensics purposes.

G, the generator, is a vital part of the GAN model for generating images of the raw style. Unlike most other GANs, the input to the generator is manipulated images. The anti-forensics effect can be achieved by removing traces of manipulations in manipulated images. Behind the generator, the discriminator D is introduced to act as a supervisor. The weights learned in the discriminator are assumed to be back propagated to the generator during training. Thus, the discriminator is arranged to

concatenate to the generator. We would like to investigate the architecture of the generator and discriminator in detail later.

The architecture and networks introduced above are those of the proposed prototype GAN model. The loss of G and D would gradually stabilize after training with sufficient iterations. Subsequently, the G will be capable of synthesizing images with anti-forensicsability.

5.4.2 Extra Supervision and Loss Function

As discussed above, our prototype GAN model is only supervised by a single discriminator to distinguish raw images from synthesized images. Except for the synchronized image content, this strategy restrains the generated images from only one aspect, that is, the synthesized images should be close to the raw images in terms of high-level patterns and statistics. Thus, the loss function can be designed with the formula below.

$$\mathcal{L}(G, D) = E[logD(I, G(I_m, n)] \tag{5.1}$$

where I_m is the manipulated images that are employed as inputs to the generative network, I is the raw images, which also represent target signals that supervise the generation procedure, and n is noise that should also be fed to the generator. Note that here for our anti-forensics purpose, we define the noise n as the inverse signals of the manipulated fingerprints. If it is applied to the manipulated images, the synthesized images could be images without any manipulated fingerprints. With S indicating the image style, n can be represented as

$$n = S(I) - S(I_m) \tag{5.2}$$

Consequently, the synthesized image I_g is

$$I_g = G(I_m, n) \tag{5.3}$$

Since our objective is to introduce noise to the manipulated images to reconstruct the images I_g that are close to raw images I, the loss of the generator must be minimized, while the loss of the discriminator should be maximized. Thus, the entire procedure of the GAN model can be described with the following equation.

$$\begin{aligned} T =& arg \min_G \max_D \{\mathcal{L}(G, D)\} \\ =& arg \min_G \max_D \{E[logD(I, I_g]\} \end{aligned} \tag{5.4}$$

Although the prototype model may satisfy the fundamental requirements to perform as a conditional GAN to remove manipulated fingerprints, we believe that the anti-forensicsability of the model can be improved if proper enhanced supervision can be conducted. Therefore, we designed a refined supervision system to be associated with the prototype model to boost the performance. The proposed refined model is depicted in Figure 5.3.

As seen in the two figures, the major difference is that two new discriminators, D2 and D3, are introduced in the refined structure. These two discriminators serve as Extra-Supervisions (Ex-S) for the prototype.

D2 is assumed to be trained to classify the output of the generative network from the input. Through back-propagation, the learned weights are transferred back to the generator. This strategy guarantees that the synthesized images should be far from the manipulated images in terms of high-level patterns and statistics while preserving the content.

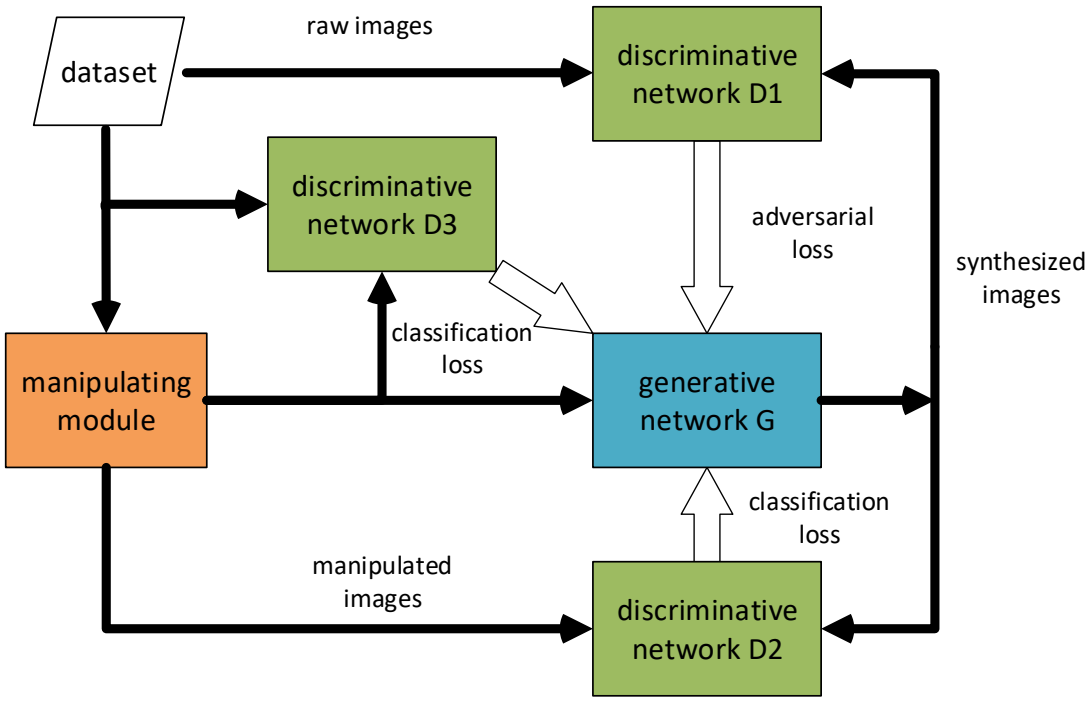


Figure 5.3 Proposed GAN structure with Ex-S.

Similar to many forensics detectors based on CNN, D3 is responsible for learning to extract features with higher efficiency to discriminate manipulated images from raw images during training. The weights learned by D3 can also be transferred to the generator. This can enhance G and allow it to make wiser choices to avoid inputting features learned in D3 into synthesized images. As a result, the synthesized images may become more difficult to distinguish from raw images. Modules with similar functions to D3 can also be found in other works. It has been proven to have a positive impact on generating desired signals [13].

During GAN training, all the discriminators are trained simultaneously, along with the generator. Nevertheless, we expect different convergence performances from each discriminator. As discussed in the prior discussion, the generative network deliberately deceives D1 to prevent it from converging. In contrast, both D2 and D3 are required to converge with high classification performance to enhance the

generation from different aspects. Therefore, the loss function for this refined model can be defined as

$$\begin{aligned}\mathcal{L}_r(G, D_1, D_2, D_3) =& E[logD_1(I, I_g)]+ \\ & E[1 - logD_2(I_m, I_g)]+ \\ & E[logD_3(I, I_m)]\end{aligned} \tag{5.5}$$

In addition, as learned from a recent report [32][55] on conditional GANs, we also deploy an $\mathcal{L}_1$ loss to enhance the performance of the generative network. This strategy has been proven to be capable of improving the quality of synthesized images. This loss can be described with the following equation.

$$\mathcal{L}_1(G) = E_{I,I_m,I_g}[||I_g - G(I_m, n)||_1] \tag{5.6}$$

We have the complete form of the loss function for the refined model as follows:

$$\mathcal{L}'(G, D_1, D_2, D_3) = \mathcal{L}_r(G, D_1, D_2, D_3) + \lambda\mathcal{L}_1(G) \tag{5.7}$$

This may lead to our final goal of minimizing the loss for G, D_2 and D_3 while maximizing the loss for D_1 during training. The procedure can be described with the following equation.

$$T' = arg \min_{(G,D_2,D_3)} \max_{D_1} \{\mathcal{L}'(G, D_1, D_2, D_3)\} \tag{5.8}$$

5.4.3 Architectures of Discriminator and Generator

With the GAN architectures and loss functions studied, the remaining task is to discuss the architectures of the discriminative network and generative networks.

To the best of our knowledge, many proposed CNNs are serving as detectors in digital forensics. Although they have been employed to solve different problems successfully, most methods use simple, single lanes of feed-forward structures, which can be considered homogeneous to AlexNet and LeNet. A similar arrangement can also be found for many discriminators in GANs. Thus, considering that the difficulty of discrimination tasks is not high, we also employ a simple structure of this type for all the discriminators in our proposed models. The architecture of our discriminators is depicted in Figure 5.4.

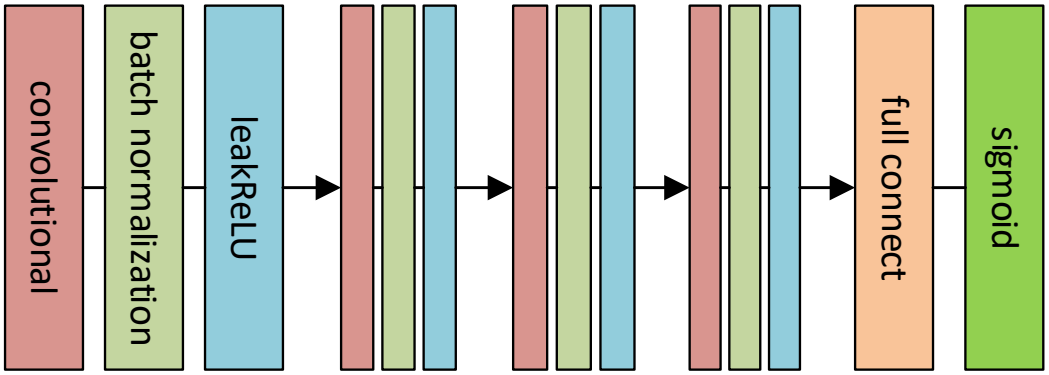


Figure 5.4 Structure of discriminative network. The kernel size in all convolutional layers is 5×5, the stride is 2. The number of filters in first convolutional layer is 64, it is always doubled in the next convolutional layer of the network. The slope for leakyReLU is 0.2.

The generator is the decisive component and can directly impact the anti-forensics performance. Therefore, we put greater effort into investigating generative networks with different architectures. Since the input signals to our generator are images and since the outputs are also images of uniform size, one of the most typical generative structures that suits this circumstance is the end-to-end model with a

downsampling network and an upsampling network. In this model, the input images are first downsampled into feature vectors in the downsampling network, and then, the feature vectors are reconstructed as images by the upsampling network. This structure is illustrated in Figure 5.4. Multiple convolutional layers are arranged in series to function as the downsampling network. Eventually, after processing by these convolutional layers, the images can be downsampled into feature vectors.

After downsampling, we employ an upsampling network to reconstruct the images from the feature vectors. The upsampling network consists of multiple deconvolutional layers in series. The upsampling network is symmetric to the downsampling network to ensure a consistent image size. This is the simplest and most fundamental structure that can be considered the basic generator for our GANs.

The upsampling here is used to restore the image to be of equal size to the input image. In most cases, deconvolution is the preferred upsampling method over linear interpolation in GANs. From the literature, it can be assumed that the input images should be downsampled by a series of convolutional layers to produce feature vectors as output. The output of the downsampling network should be the input for the upsampling network to have the image reconstructed. Thus, the upsampling network, consisting of multiple deconvolutional layers in series, should be connected behind the downsampling network to rebuild the images. This structure is shown in Figure 5.5. It is the simplest and most fundamental structure for building our desired generative network.

In addition to the structure of the basic generator, there are also other advanced end-to-end architectures. These architectures can be considered refined versions of the basic model. They can serve as optional structures for our generator. Here, we introduce two types of refined versions and evaluate them later with experiments. The first optional adjustment is inserting a transformation network between the upsampling and downsampling networks [34]. The transformation network is

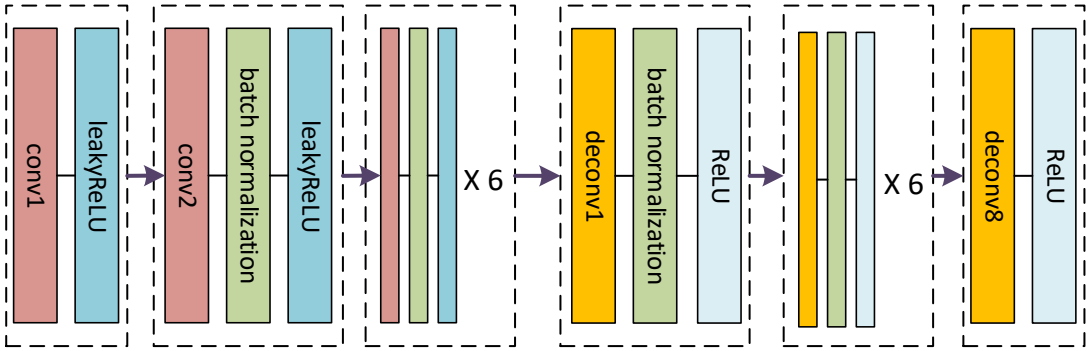


Figure 5.5 Structure of basic generative network. For all convolutional and deconvolutional layers, the kernel size is fixed to 4, the stride is 2. The number of filters is n, n=64 for conv1 and deconv8, n=128 for conv2 and deconv7, n=256 for conv3 and deconv6 and n =512 for all the others. Batch size is fixed to 1.

composed with multiple residual blocks which can be regarded as convolutional layers. It has been proved to be efficient for transforming images that can be employed as generator in GANs. We name this structure T-Net in this investigation. T-Net is employed to select the desired feature vectors during the back-propagation of the model to be upscaled and embedded in the synthesized images. This enhances the effect of the style transformation, which could be an advantage in achieving greater anti-forensibility. The second optional adjustment is inspired by U-Net [56], which was proposed in Ronneberger *et al.*'s study. It establishes channels for the symmetrically located layers in upsampling and downsampling networks to enable one-way communication in corresponding layers from the downsampling network to the upsampling network. As a result of their strategy, the deconvolutional layers in the upsampling network can reconstruct the images with the assistance of corresponding convolutional layers to improve the accuracy of the details in the synthesized images. Consequently, the synthesized images generated via this model are expected to be of higher quality than the other methods. The architectures of these advanced generative models are illustrated in Figs. 5.6 and 5.7.

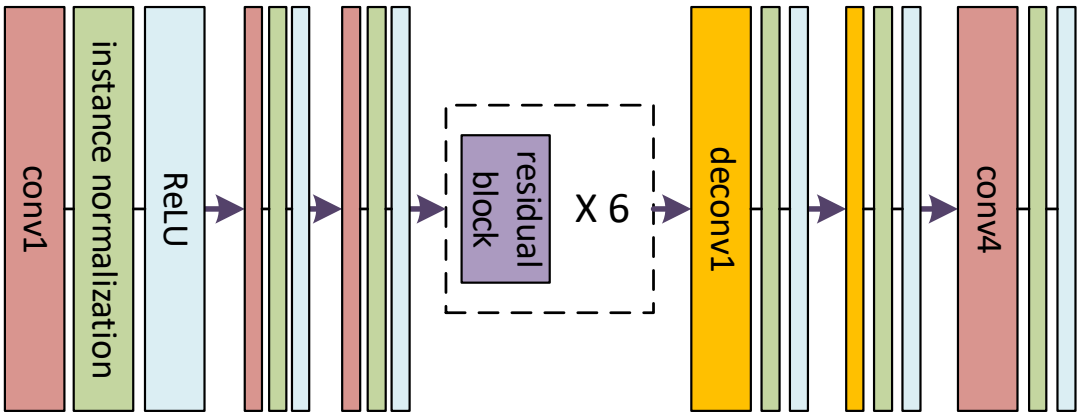


Figure 5.6 Generative network of T-Net.

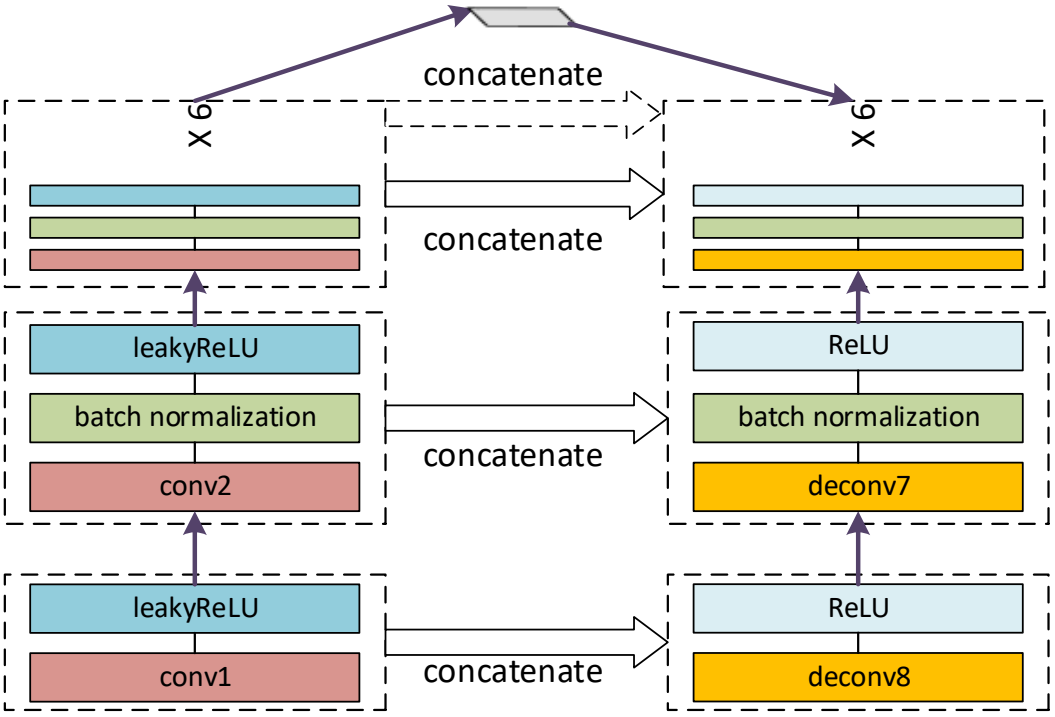


Figure 5.7 Generative network of U-Net. For all convolutional and deconvolutional layers, the kernel size is fixed to 5, the stride is 2. The number of filters is n, n=64 for conv1 and deconv8, n=128 for conv2 and deconv7, n=256 for conv3 and deconv6 and n =512 for all the others. Batch size is fixed to 1.

5.5 Experimental Results and Discussion

In order to serve as a general anti-forensics tool, The evaluation for the proposed GAN model is based on three datasets: the BOSS, RAISE and UCID datasets. The BOSS image dataset contains 10000 grayscale images of size 512×512. RAISE is a relatively new image dataset released in 2015. It consists of 8156 high-resolution raw images of size 4288×2848 or 4928×3264 and is intended for study on digital forensics. UCID includes 1338 uncompressed color images of size 384×512. The BOSS and RAISE datasets are our training datasets, while the UCID dataset is the validation set. All images are randomly cropped to a uniform size of 256×256. In addition, all color images are converted to grayscale images for our experiments. All experiments are simulated with TensorFlow 1.1.0 and CUDA 8.0.

In our experiment, we would like to assess the anti-forensicsability of the GAN models with some common image editing manipulations to serve as a general anti-forensics tool. Hence, the following manipulations were selected in the manipulation module for our GAN models: Gaussian filtering, median filtering, average filtering, USM sharpening, adding Gaussian noise and JPEG compression. The parameters applied for each manipulation in our experiments can be found in Table 5.1.

Table 5.1 Employed Manipulations With ID and Related Parameters

Editing manipulations (EM)	Parameters
Gaussian filtering (GF)	3×3 window size, $\sigma = 0.8$
Median filtering (MF)	3×3 window size
Average filtering (AF)	3×3 window size
USM sharpening (US)	$\sigma = 1, \lambda = 1$
Gaussian noising (AGN)	$\sigma = 0.01$
JPEG compression (JC)	$Q = 50$

5.5.1 Study of Ex-S and Generator Structure

First, we study the structures of the generator and the proposed Ex-S enhanced supervision system. To achieve this goal, we employ the prototype GAN model α

with different generator structures introduced in chapter 5.4: the encoder-decoder E, the U-Net U and the Transformation-Net T as options for assessment. In addition, we conducted several experiments in ablation studies of Ex-S. In this case, individual D2, D3 and Ex-S are tested along with the prototype model α. The generator in α is fixed with E to ensure the ablation study is professional. Each model is trained for 100 epochs with learning rates of 0.0002, after which validation images are synthesized from the raw images by the trained models.

To evaluate the anti-forensicsability, it is necessary to employ a forensics tool as a benchmark. In this experiment, we choose the constrained CNN (CCNN) [4] to play this role. Although many famous classifiers can be employed as forensics detectors, the constrained CNN [4] proposed in 2018 is generally considered the state-of-the-art detector in digital forensics. The reported results outperform almost all digital forensics tools proposed in past years. In addition, the CCNN covers a wide range of image editing manipulations and can also serve as a universal tool. Given all the advantages, our ideal validation tool should be the CCNN. Therefore, several CCNNs are trained against the manipulations listed in Table 5.1. The observed detection performance reported in Table 5.2 demonstrates that it is an effective and reliable tool as a benchmark.

Table 5.2 Detection Accuracy of Trained Constrained CNN

EM	Detection accuracy
GF	99.12%
MF	99.67%
AF	99.52%
US	99.25%
AGN	99.93%
JC	99.34%

The synthesized images are then validated via corresponding detectors. For each manipulation, the ratio of synthesized images classified as manipulated images is listed in Table 5.3.

Table 5.3 Results of Ablation Study for Models With Different Generators and Supervision Modules

EM	α + E	α + U	α + T	α + D2	α + D3	α+Ex-S
GF	13.64%	13.59%	14.38%	**9.13%**	10.21%	9.16%
MF	1.93%	1.66%	3.47%	1.02%	1.58%	**0.29%**
AF	0.25%	0.57%	0.96%	**0.01%**	0.06%	**0.01%**
US	22.33%	20.51%	26.21%	15.46%	17.01%	**12.89%**
AGN	0.07%	0.01%	0.19%	0.03%	0.01%	**0.00%**
JC	19.51%	19.63%	23.70%	16.32%	17.30%	**13.17%**

From Table 5.3, we can see that each extra discriminator boosts the anti-forensibility of the GAN model. Nevertheless, it can also be observed that the model with the joint supervision of two discriminators achieves the best anti-forensics performance of all the models. Although the structure of the generator can have a certain effect on the anti-forensibility of the GAN models, this effect is not prominent. Therefore, it is difficult to determine which structure is more advanced, at least from the perspective of anti-forensibility.

We examine the qualities of the synthesized images via 3 benchmarks: PSNR, SSIM and VIF. PSNR and SSIM are two famous benchmarks in image processing. VIF is an image quality assessment method proposed in 2006 [60]. Unlike the other methods, VIF evaluates the image quality in a perceptually consistent manner that matches the human vision system. The quality assessment can be found in Tables 5.4, 5.5 and 5.6.

Table 5.4 The Average PSNR for Images Synthesized by Different Models

EM	α + E	α + U	α + T	α + D2	α + D3	α + Ex-S	I_m
GF	31.03	**32.72**	25.31	31.07	30.88	30.97	29.89
MF	27.86	**30.62**	25.27	27.99	27.86	27.85	28.35
AF	27.69	**29.31**	23.57	27.72	27.75	27.76	27.68
US	35.13	**36.39**	26.77	35.08	35.10	35.12	35.38
AGN	24.55	**26.90**	16.77	24.60	24.53	24.64	19.87
JC	31.80	**33.77**	23.98	31.65	31.71	31.74	33.09

Table 5.5 The Average SSIM for Images Synthesized by Different Models

EM	α + E	α + U	α + T	α + D2	α + D3	α + Ex-S	I_m
GF	0.913	**0.941**	0.850	0.916	0.918	0.916	0.883
MF	0.827	**0.903**	0.726	0.828	0.825	0.826	0.836
AF	0.887	**0.922**	0.715	0.900	0.902	0.892	0.817
US	0.978	**0.988**	0.852	0.979	0.976	0.981	0.975
AGN	0.617	**0.731**	0.387	0.638	0.620	0.652	0.389
JC	0.919	**0.933**	0.735	0.922	0.920	0.920	0.923

Table 5.6 The Average VIF for Images Synthesized by Different Models

EM	α + E	α + U	α + T	α + D2	α + D3	α + Ex-S	I_m
GF	0.809	**0.861**	0.576	0.806	0.801	0.800	0.612
MF	0.493	**0.547**	0.408	0.495	0.492	0.492	0.526
AF	0.680	**0.759**	0.406	0.667	0.666	0.671	0.552
US	0.898	0.945	0.788	0.895	0.908	0.902	**0.947**
AGN	0.182	**0.278**	0.075	0.191	0.180	0.193	0.247
JC	0.670	**0.714**	0.437	0.675	0.681	0.668	0.712

The quality assessments indicate that the structure of the generator has a strong impact on the quality of the synthesized images. On the other hand, the anti-forensics performance is less relevant to it, as we cannot tell which structure can boost the anti-forensicsability over the others. Thus, the image quality is our only concern in making the decision for the generator. Based on the quality assessments and the ablation study for supervision, we choose U-Net as the generator in association with Ex-S as our proposed GAN model.

5.5.2 Evaluation of the Proposed GAN Model

Since the structure of the proposed GAN model is determined via the justifications above, we thoroughly evaluate the model by conducting more experiments. First, we trained the proposed model. Then, a validation set with patch size of 256×256 was synthesized via the trained model. Some examples of the synthesized images and the corresponding raw images and manipulated images are illustrated in Figure 5.8. Additionally, we investigated the effect of the patch size for a deeper study.

Therefore, in addition to a patch size of 256×256, validation images of size 128×128 and 64×64 are also synthesized following the identical pipeline of removing the manipulated fingerprints in the manipulated images I_m. The average PSNR, SSIM and VIF values of the synthesized images and manipulated images with their corresponding raw images are calculated as the quality assessments. The results are displayed in Table 5.7

Table 5.7 Quality Assessment for Image of Different Sizes Synthesized Via Ex-S GAN

EM	256×256			128×128			64×64		
	PSNR	SSIM	VIF	PSNR	SSIM	VIF	PSNR	SSIM	VIF
GF	32.77	0.938	0.864	32.67	0.931	0.865	32.66	0.933	0.865
MF	30.71	0.898	0.559	30.70	0.900	0.562	30.67	0.903	0.560
AF	29.33	0.919	0.763	29.36	0.917	0.760	29.37	0.915	0.760
US	36.41	0.989	0.948	36.38	0.991	0.948	36.40	0.990	0.947
AGN	26.92	0.742	0.290	26.73	0.726	0.288	26.34	0.721	0.281
JC	33.72	0.937	0.715	33.79	0.935	0.717	33.80	0.935	0.714

The results of the quality assessment demonstrate that the quality of the synthesized image is quite high. It is well known in traditional anti-forensics that attacking images produces distortions. Consequently, the image quality is always sacrificed to enhance the anti-forensicsability. However, this rule does not apply to our proposed method. Surprisingly, in contrast, summarized from the observed results, the synthesized images tend to have higher quality than the attacked images I_m. After removing the artifacts of the manipulations, the synthesized images are visually closer to the manipulations-free images. This can be considered a tremendous advantage for anti-forensics based on GANs. In addition, it can also be observed that the patch size has little impact on the quality of the synthesized images.

We conducted several experiments to investigate the anti-forensibility of the proposed model. In most conventional cases, for anti-forensics, an attack is successful if an attacked image can be falsely determined as untouched image by a binary classifier. Therefore, since there were only two categories for classification, an

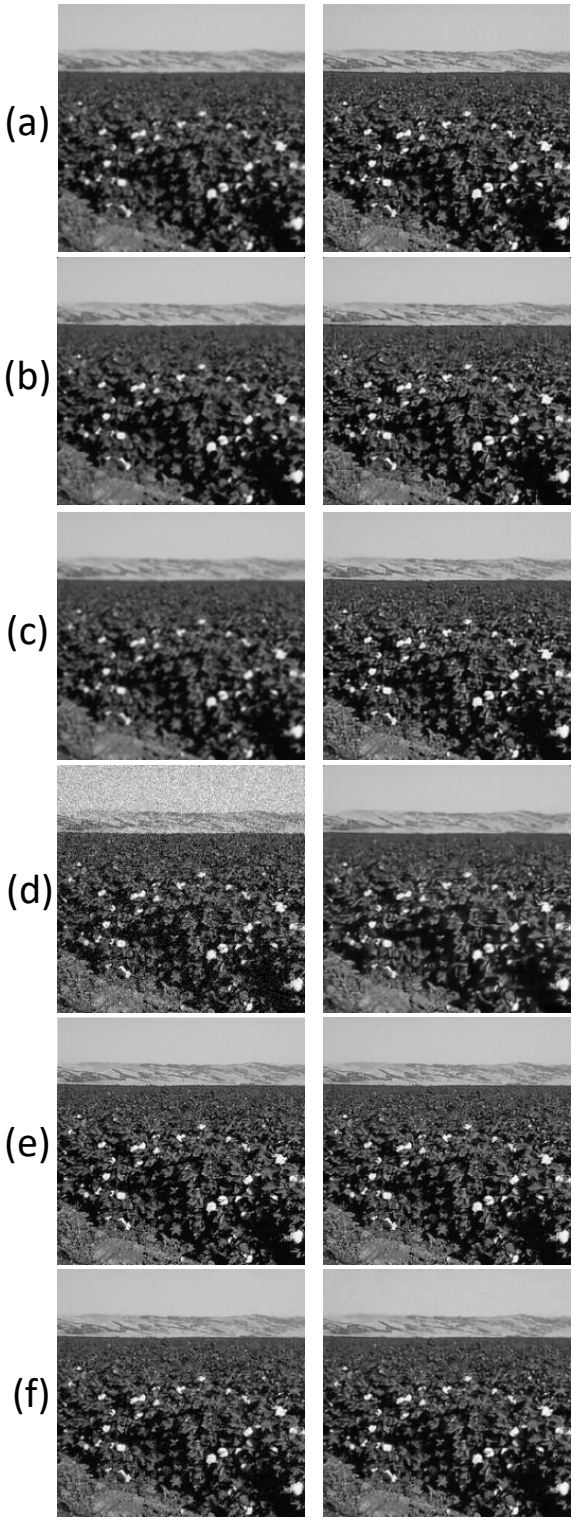


Figure 5.8 Sample images. (a) Gaussian filtering, (b) median filtering, (c) average filtering, (d) Gaussian noising, (e) USM sharpening, (f) JPEG compression. The images in the left column are the manipulated images, the ones in the right column are the synthesized images.

anti-forensics method can be regarded as higher performance if less attacked images are classified as manipulated images.

Here, we employ more forensics detectors to fulfill the task. Along with the CCNN introduced in above subsection, the VGG16 and the rich model are also chosen for validation. VGG16 is a famous CNN model for classification and detection [61]. The rich model is a non-CNN forensics tool proposed in 2012. Although the original purpose of the rich model was steganalysis, it has been proven by many researchers to be a successful algorithm in revealing manipulations in images [24]. We employ it along with ensemble learning for multi-class classification.

In our first experiment, we followed the similar process of CCNN to train the rich model and VGG16 with untouched images and manipulated images as binary classifiers to act as standard forensics detectors. Both the rich model and VGG16 can achieve detection rates of over 99% for all manipulations after training. Then, we test the two models as well as the CCNN with the images synthesized by our proposed model. In this experiment, we also tested the impact of patch size. The detection rate representing the ratios of images that are detected as 'manipulated' on images of different patch sizes are reported in Table 5.8.

Table 5.8 Anti-forensics Assessment for Image of Different Sizes Synthesized Via Ex-S GAN

EM	256×256			128×128			64×64		
	Rich model	VGG16	CCNN	Rich model	VGG16	CCNN	Rich model	VGG16	CCNN
GF	3.50%	7.72%	5.34%	3.03%	7.17%	5.45%	1.59%	6.85%	6.09%
MF	0.01%	0.97%	0.35%	0.00%	0.23%	0.52%	0.00%	0.68%	0.39%
AF	0.55%	2.15%	0.82%	0.20%	2.23%	1.67%	0.00%	1.86%	1.34%
US	22.25%	23.68%	14.03%	17.52%	25.16%	18.59%	12.77%	24.39%	17.21%
AGN	0.00%	0.01%	0.01%	0.00%	0.03%	0.01%	0.00%	0.05%	0.07%
JC	7.21%	20.89%	9.27%	4.19%	23.26%	12.79%	3.45%	22.63%	11.52%

Observed from the table, most synthesized images were falsely judged as untouched images. Considering these detectors were trained to be nearly perfect with

accuracies about 99%, most attacks launched via our proposed model can deceive the forensics detectors. We can also notice that the impact of patch sizes is not prominent. Hence, the proposed enhanced supervision system along with the GAN structure has been justified to be a successful general anti-forensics tool.

After the evaluation based on conventional binary classification detector, we extend the evaluation with thorough investigation to test universal forensics tools. For universal detectors, the most important feature is that they can be employed against a wide range of attacks. Thus, these three forensics detectors were trained with untouched images I_o and all kinds of manipulated images I_m to be powerful multi-class classifiers that can achieve an overall classification accuracies over 90%. Then, the validation set were classified by these general forensics detectors.

The confusion matrix based on classification for images with a patch size of 256×256 of each detector is shown in Tables 5.9, 5.10 and 5.11. Additionally, regardless the patch size, there are images that were correctly classified even after being attacked by our GAN model. The overall detection accuracy which represents the ratios of these images is reported in Table 5.12.

Table 5.9 Confusion Matrix of Rich Model; Prediction (Rows) vs Ground Truth (Columns)

Manipulations	GF	MF	AF	US	AGN	JC	I_o
GF	1.39%	0.00%	6.28%	31.55%	0.00%	12.64%	48.14%
MF	0.01%	0.01%	0.36%	29.44%	0.00%	4.15%	66.03%
AF	0.00%	0.00%	9.72%	27.66%	0.00%	1.72%	60.90%
US	0.00%	0.00%	0.10%	19.18%	0.01%	6.39%	74.42%
AGN	0.00%	0.00%	0.00%	39.37%	0.00%	5.11%	55.52%
JC	0.83%	0.12%	0.03%	26.32%	0.00%	6.91%	65.79%

It can be observed that most synthesized images are falsely detected as original images. However, unlike the binary classifications, for multi-class classification, it is also highly potential that synthesized images are falsely classified as images manipulated by other manipulations. For example, the sharpened images could be classified

Table 5.10 Confusion Matrix of VGG16; Prediction (Rows) vs Ground Truth (Columns)

Manipulations	GF	MF	AF	US	AGN	JC	I_o
GF	7.66%	0.52%	0.03%	4.15%	0.00%	5.47%	82.17%
MF	1.82%	0.55%	0.27%	4.98%	0.00%	6.25%	86.13%
AF	0.67%	0.58%	1.43%	3.73%	0.00%	4.08%	89.51%
US	0.03%	0.00%	0.01%	20.21%	0.00%	16.32%	63.43%
AGN	0.03%	0.12%	0.01%	29.16%	0.00%	10.75%	59.94%
JC	1.73%	0.03%	0.00%	12.45%	0.00%	18.57%	67.22%

Table 5.11 Confusion Matrix of Constrained CNN; Prediction (Rows) vs Ground Truth (Columns)

Manipulations	GF	MF	AF	US	AGN	JC	I_o
GF	5.69%	0.07%	0.12%	6.61%	0.00%	6.12%	81.39%
MF	0.03%	0.22%	0.38%	5.96%	0.03%	5.40%	88.00%
AF	1.15%	0.79%	0.03%	6.22%	0.00%	7.78%	84.03%
US	0.12%	0.03%	0.03%	9.69%	0.00%	19.73%	70.40%
AGN	0.01%	0.03%	0.00%	26.63%	0.03%	9.82%	63.48%
JC	1.12%	0.66%	0.01%	20.57%	0.00%	8.33%	69.31%

Table 5.12 The Classification Accuracy of Different Multi-class Classifiers on Images of Different Sizes

Classifiers	256×256	128×128	64×64
Rich model	6.20%	3.85%	2.52%
VGG16	8.07%	8.76%	8.68%
CCNN	4.18%	4.39%	4.15%

as compressed images after being processed by the GAN model. Nevertheless, this phenomenon can also be considered as high anti-forensics performance because the original manipulations cannot be identified by detectors.

Summarizing the above experiments, the proposed model is found to be a reliable anti-forensics tool that can deliver images with high quality while maintaining satisfactory anti-forensicsability.

5.5.3 Comparisons with Prior Study

Recall that there are existing anti-forensics approaches for median filtering and JPEG compression. In the following experiments, we compare the performance of the proposed model with these prior works.

For median filtering, Kim *et al.*'s method [36] is reported as a state-of-the-art median anti-forensics model. Their study is also based on supervised training of the GAN model. Therefore, their model is an ideal approach to be compared with. Validation images are generated with their model from the identical dataset as ours to guarantee that the comparison is fair. Here, the CCNN is trained with median filtered images and untouched images as binary classifiers for validation. For comparison, we also considered the effect of different parameters for median filtering. Hence, the comparison results for window sizes 3 and 5 are displayed in Table 5.13 with the detection accuracy and image quality. The detection accuracy here is the ratio of synthesized images that are classified as median filtered images.

Table 5.13 Comparison with Kim *et al.*'s Method [36]

Window size	Methods	PSNR	SSIM	VIF	Accuracy
3×3	[36]	28.45	0.870	0.553	4.07%
	Proposed	**30.71**	**0.898**	**0.559**	**0.35%**
5×5	[36]	24.52	0.741	0.312	4.52%
	Proposed	**26.19**	**0.763**	**0.347**	**0.50%**

We follow the same pipeline to implement comparisons with Stamm *et al.*'s method [64] and Luo *et al.*'s method [43] as JPEG compression anti-forensics models. For JPEG compression, quality factors of 30 and 70 are also considered for comparison. We still employ the trained CCNN as our validation tool. The detection accuracy is still the ratio of synthesized images that are classified as compressed images. The performance, including detection accuracy and image quality, can be found in Table 5.14

Table 5.14 Detection Accuracy and Image Quality Comparison with Stamm *et al.*'s Method [64] and Luo *et al.*'s Method [43] [36]

Quality factor	Methods	PSNR	SSIM	VIF	Accuracy
30	[64]	26.74	0.811	0.598	**0.13%**
	[43]	30.94	0.901	**0.676**	17.43%
	Proposed	**31.15**	**0.904**	0.670	13.28%
50	[64]	27.72	0.840	0.616	**0.27%**
	[43]	32.90	0.920	0.696	13.21%
	Proposed	**33.72**	**0.937**	**0.715**	9.27%
70	[64]	28.55	0.859	0.641	**0.00%**
	[43]	34.94	**0.958**	0.728	6.55%
	Proposed	**35.47**	0.956	**0.730**	2.26%

As observed from the experimental results, Stamm *et al.*'s method can achieve the best anti-forensicsability in that the compression artifact left in compressed images can be completely removed. However, this merit comes at the price of sacrificing image quality, as we mentioned above. For images synthesized with GANs, most of them can deceive the detectors. The detection accuracy is lowered to below 20%, which is acceptable. Although the anti-forensibility is relatively low in contrast to [64], the image quality can be satisfactory. For the performance of two GAN models, our proposed model outperforms Luo *et al.*'s method, with slight improvements in both anti-forensicsability and image quality in most cases. We can also find that when the quality of the compressed image is higher, it is easier to generate images than can deceive the detector. This is because there are fewer artifacts left in the high-

quality compressed images, which may be easier for GANs to clean. Consequently, the synthesized images tend to be higher quality.

5.5.4 Limitations of the Proposed GAN Model

Despite the effectiveness of the proposed model was justified, during our experiments, there were also some noticeable issues. Here, we discuss these problems based on our observation to provide a thorough evaluation for employing GANs as anti-forensics tools. The first is the checkerboard artifact as we mentioned in Section 5.5. A GAN-generated sample with checkerboard artifacts is displayed in Figure 5.9(a). The checkerboard artifact is clear enough to be visible to human eyes.

The checkerboard artifact has been proved as the trace left by transposed convolution (also known as 'deconvolution') which is widely used in generating networks of GANs. These artifacts are generated when transposed convolution layers upsample images. Technically, the checkerboard artifacts are not relevant to the training process of GANs. However, since transposed convolution is now mainly applied in GANs to construct images, the checkerboard artifacts are regarded as the traces of GANs. If the artifacts are visible in synthesized images, it would make the images dubious that the anti-forensics attacks can be considered as failures. Hence, it is necessary to prevent the generation of checkerboard artifacts in our proposed model.

So far, several works have been reported to avoid generating checkerboard artifacts in GANs [45][66]. Among them, the most simple and effective method is adjusting the kernel sizes and strides in transposed convolution layers. When the kernel size can be divided by stride without remainders, it could prominently ease the checkerboard artifacts. In our experiment, we applied this strategy to synthesize images without checkerboard artifacts as shown in Figure 5.9(b). Both the two samples are generated from the same Gaussian filtered image.

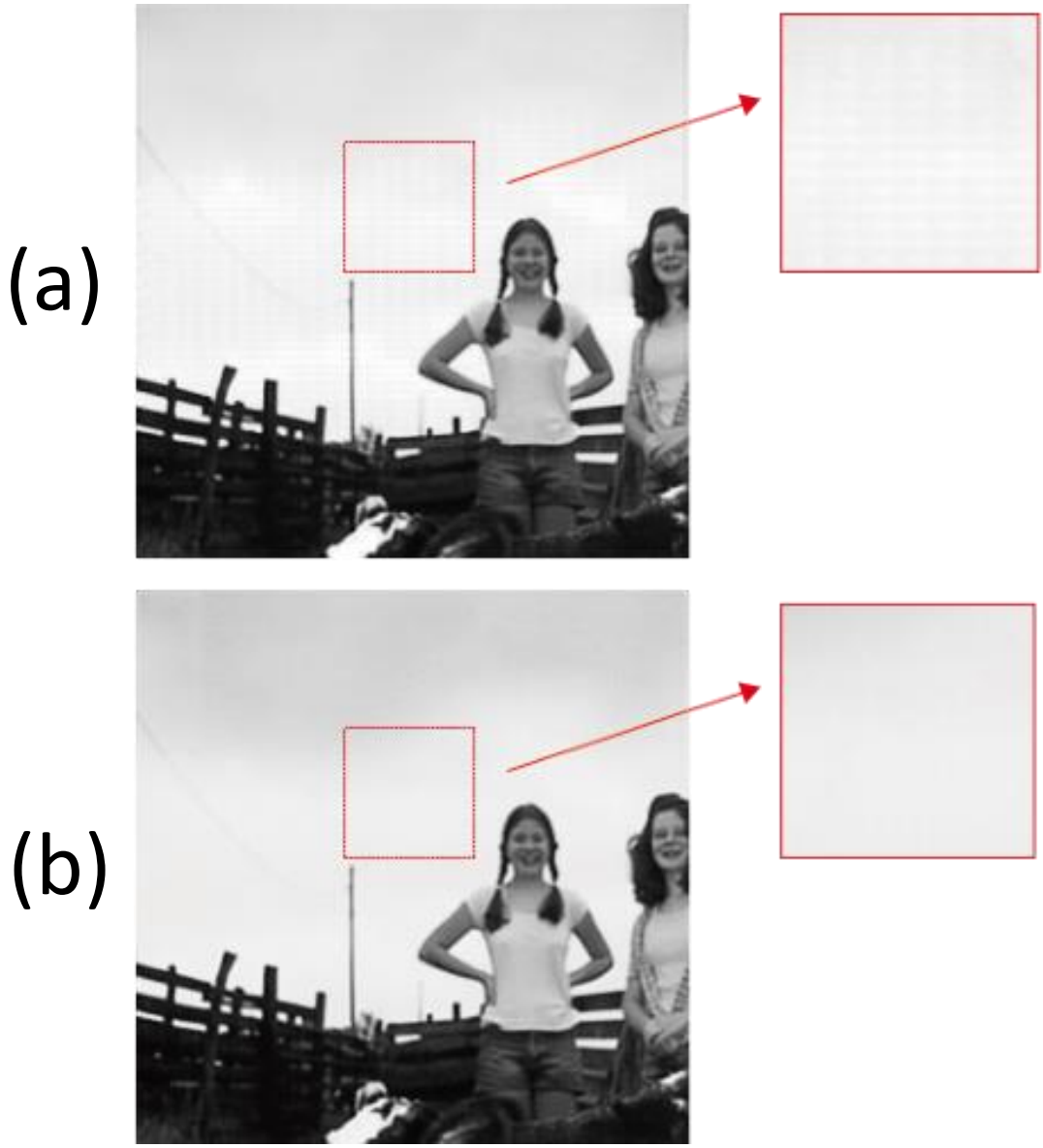


Figure 5.9 Samples generated from Gaussian filtered image, (a) generated image with checkerboard artifacts, (b) generated image without checkerboard artifacts.

Besides, we can also found random flaws in some synthesized images. This is absolutely intolerable for anti-forensics as images with such flaws could be easily identified as unnatural images. However, given the nature of deep learning, GANs remain un-interpretable. Subsequently, the training process of GANs is beyond our control. Hence, if flaws appear, it is feasible to re-start the training process untill the model is adequate for the target images. Although it could be time-consuming, it is necessary to launch a pinpoint attack on target images. Some samples are displayed in Figure 5.10. After flaws are discovered in the left column of Figure 5.10, we re-trained the same model once respectively toward Gaussian filtering and average filtering to generate the images on the right column.

Additionally, we also noticed that it is difficult for the proposed model to generate highly convincing data from images that are intensively manipulated or polluted by noise. Generally, it is also quite challenging to restore the lost information in such images. Some details may not be accurately restored in these images. Although some of these images can be employed to fool human eyes in blind detection, it still cannot be defined as successful attacks as not all details are restored. In particular, when intense Gaussian noise is added in images, the image contents are severely distorted, the proposed model completely failed in restoring contents for all test images. Several failure cases are displayed in Figure 5.11.

5.5.5 Summarization

After all the experiments and evaluations, several points can be summarized as follows.

1) Our experiments demonstrate that GANs can achieve high anti-forensics performance toward many common image editing manipulations. The synthesized images can be used to deceive and attack forensics detectors regardless of whether the detector is based on CNNs.

2) Our proposed model has also been justified as an ideal anti-forensics tool. The Ex-S system and the U-Net are the key components for the model to achieve success

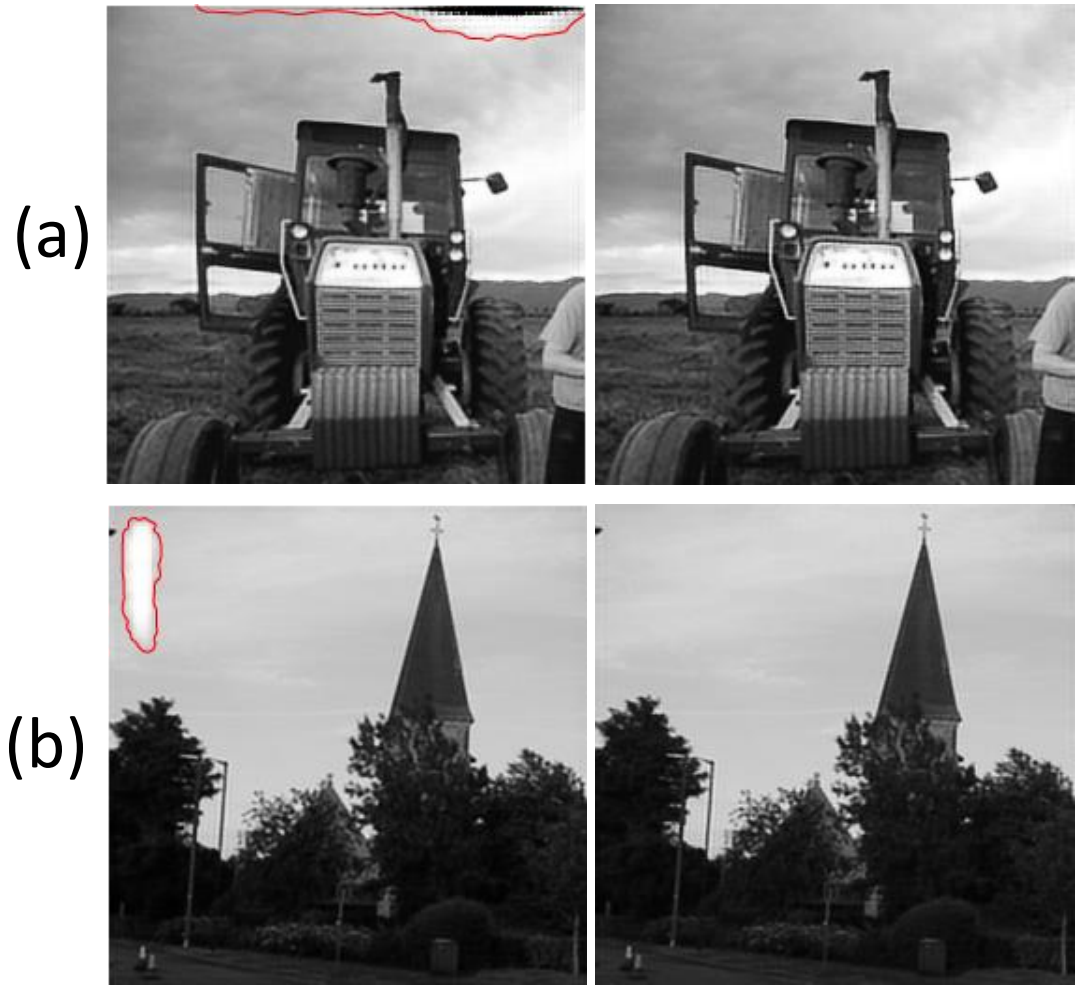


Figure 5.10 Samples generated by the identical model trained with different performance (a) images generated from Gaussian filtered image, left: sample with visible flaws; right: sample without visible flaw (b) images generated from average filtered image, left: sample with visible flaws; right: sample without visible flaw.

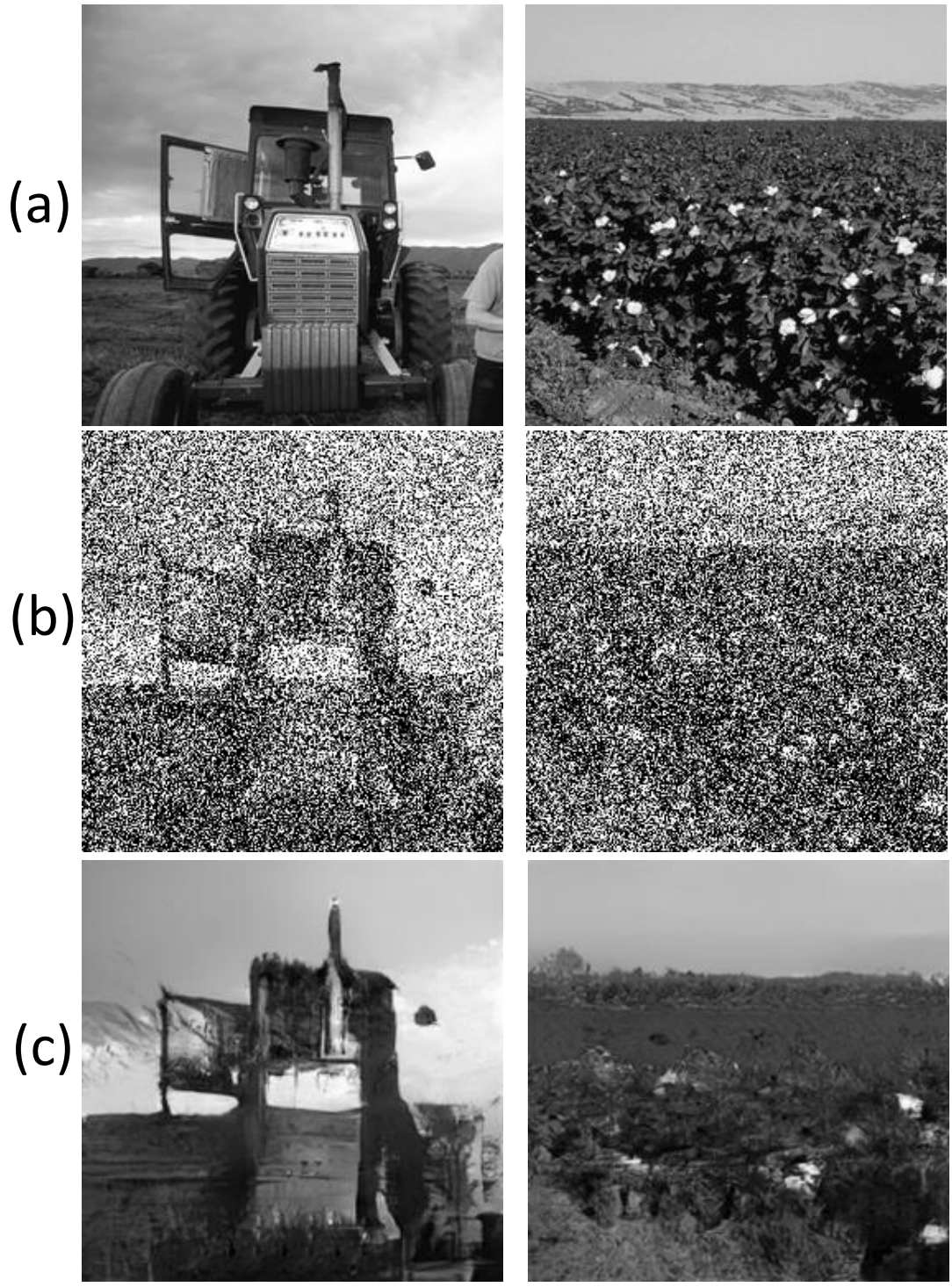


Figure 5.11 (a) Untouched images; (b) manipulated images with intense Gaussian noise; (c) failed images synthesized from noising images.

for anti-forensics.

3) Unlike traditional mathematical modeling for anti-forensics, attacks based on GANs do not require image quality sacrifices. Consequently, as anti-forensics tools, GANs have more practical value than traditional mathematical models. Besides, Minimal expertise is required to generate these images if the GAN model is prebuilt and trained.

4) GANs could produce checkerboard artifacts and distortions in generated images. The production of checkerboard artifact can be avoided by applying proper parameters in transposed convolution layers. The distortions can be also avoided by training multiple times.

5) It is not suitable to apply GANs to generate images from heavily distorted images, especially when intense Gaussian noise is added.

CHAPTER 6

SUMMARY

6.1 Major Contributions

In this dissertation, machine learning based (ML-based) methods have been developed to solve problems of image forensics and anti-forensics.

In Chapter 2, an efficient CNN structure is proposed to detect images processed by unsharp masking (USM). The proposed CNN structure contains 16 layers in convolutional module to generate 64-D features. To aim at sharpening detection, which is an edge enhancement algorithm, the Max pooling is taken to process down sampling. In addition, the ReLU layers have speed up the processing and greatly reduced the experimental time. Finally, the experimental results derived from two large image data sets have shown that the proposed CNN structure outperforms the existing method (EPTC) for sharpening detection in term of detection accuracy.

In Chapter 3, a method for image recoloring forensics is proposed. It is based on CNN structure that consists of 12 layers. The performance of the proposed CNN is excellent regardless if it is binary or multiple label classification. It is capable of detecting recoloring as well as identify which recoloring algorithm is applied. The proposed method can reach accuracies over 90% under all circumstances. In addition, we also discussed about the impact of the network with different depths and activation strategies for the problem. Besides, the oscillation can be observed for the convergence of training procedure during our experiment.

In Chapter 4, an anti-forensics method is proposed. Our method is capable of generating images, which have sharpening effect that aims to deceive the state-of-art sharpening detector. The proposed GAN model consists of a generator and a discriminator. It features generating images with sharpening effect while preserving the image contents. Observed from the experimental results, the generated images

can be successfully classified as sharpened images although they have never been sharpened via any traditional sharpening manipulation. In addition, the quality of the generated images is also better with higher PSNR when compare with the real sharpened images. This can be considered as a tremendous success that it shows the potential of GAN models in generating images.

In Chapter 5, we investigate the capability of GANs to perform as anti-forensics tools. Discussions are made on this topic after proposing a GAN model as a universal anti-forensics tool. The proposed GAN model can remove the fingerprints of manipulations, which makes it a black-box anti-forensics method. We have proven by our experiments that most synthesized images are undetectable by forensic detectors regardless of whether they are based on CNNs. In addition, the images synthesized by GANs also have higher quality when compared with the traditional anti-forensics approach. Some discussion are also made to avoid generating checkerboard artifacts and distortions in synthesized images. We further contribute to this topic through an ablation study and comparison of our proposed model with prior study. Images generated by our proposed model show superior performance, with higher anti-forensibility and improved quality. Besides reporting the achievements we made during our investigation, we also discuss the drawbacks of GANs as anti-forensics tool, such as generating checkerboard artifacts and random distortions, failure to restore heavily distorted images, and so on.

6.2 Future Work

Most anti-forensics algorithms proposed in the past relies on specialists with expertise to build corresponding anti-forensics models towards different manipulations. However, unlike the traditional approaches, this process is now significantly simplified that ordinary people without any professional training can easily build their own attackers based on GANs by organizing proper data. This makes GANs

more dangerous than any anti-forensics methods in the past. Given the current circumstance, despite there are limitation for GANs, it can be foreseen that the prospects of GANs are promising. More advanced GAN models can be developed in the future, with significantly improved performance. Antiforensics via GANs could be a potential huge threat to information security. The development of forensic tools should be encouraged against this situation.

REFERENCES

[1] Mauro Barni, Luca Bondi, Nicolò Bonettini, Paolo Bestagini, Andrea Costanzo, Marco Maggini, Benedetta Tondi, and Stefano Tubaro. Aligned and non-aligned double jpeg detection using convolutional neural networks. *Journal of Visual Communication and Image Representation*, 49:153–163, 2017.

[2] Patrick Bas, Tomáš Filler, and Tomáš Pevnỳ. " break our steganographic system": the ins and outs of organizing boss. In *International Workshop on Information Hiding*, pages 59–70. Berlin, Heidelberg: Springer, 2011.

[3] Belhassen Bayar and Matthew C Stamm. A deep learning approach to universal image manipulation detection using a new convolutional layer. In *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*, pages 5–10, 2016.

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

[5] Sue Becker and Yann Le Cun. Improving the convergence of back-propagation learning with second order methods. In *Proceedings of the 1988 Connectionist Models Summer School*, pages 29–37, 1988.

[6] Rainer Böhme and Matthias Kirchner. Counter-forensics: Attacking image forensics. In *Digital Image Forensics*, pages 327–366. Springer, 2013.

[7] Mehdi Boroumand and Jessica Fridrich. Deep learning for detecting processing history of images. *Electronic Imaging*, 2018(7):213–1, 2018.

[8] John Francis Canny. Finding edges and lines in images. Technical Report ADA130824, Massachusetts Inst of Tech Cambridge Artificial Intelligence Lab, 1983.

[9] Gang Cao, Yao Zhao, and Rongrong Ni. Detection of image sharpening based on histogram aberration and ringing artifacts. In *IEEE International Conference on Multimedia and Expo*, pages 1026–1029, 2009.

[10] Gang Cao, Yao Zhao, Rongrong Ni, and Alex C Kot. Unsharp masking sharpening detection via overshoot artifacts analysis. *IEEE Signal Processing Letters*, 18(10):603–606, 2011.

[11] Chen Chen and Matthew C Stamm. Camera model identification framework using an ensemble of demosaicing features. In *IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6, 2015.

[12] Chen Chen, Xinwei Zhao, and Matthew C Stamm. Mislgan: an anti-forensic camera model falsification framework using a generative adversarial network. In *25th IEEE International Conference on Image Processing (ICIP)*, pages 535–539, 2018.

[13] Chen Chen, Xinwei Zhao, and Matthew C Stamm. Generative adversarial attacks against deep-learning-based camera model identification. *IEEE Transactions on Information Forensics and Security*, 2019.

[14] Feng Ding, Weiqiang Dong, Guopu Zhu, and Yun-Qing Shi. An advanced texture analysis method for image sharpening detection. In *International Workshop on Digital Watermarking*, pages 72–82. Springer, 2015.

[15] Feng Ding, Hanzhou Wu, Guopu Zhu, and Yun-Qing Shi. Meteor: Measurable energy map toward the estimation of resampling rate via a convolutional neural network. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(12):4715–4727, 2020.

[16] Feng Ding, Guopu Zhu, Weiqiang Dong, and Yun-Qing Shi. An efficient weak sharpening detection method for image forensics. *Journal of Visual Communication and Image Representation*, 50:93–99, 2018.

[17] Feng Ding, Guopu Zhu, and Yun Qing Shi. A novel method for detecting image sharpening based on local binary pattern. In *International Workshop on Digital Watermarking*, pages 180–191. Springer, 2013.

[18] Feng Ding, Guopu Zhu, Jianquan Yang, Jin Xie, and Yun-Qing Shi. Edge perpendicular binary coding for usm sharpening detection. *IEEE Signal Processing Letters*, 22(3):327–331, 2014.

[19] Hany Farid. Digital image forensics. *Scientific American*, 298(6):66–71, 2008.

[20] Hany Farid. Image forgery detection. *IEEE Signal Processing Magazine*, 26(2):16–25, 2009.

[21] Marco Fontani and Mauro Barni. Hiding traces of median filtering in digital images. In *Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, pages 1239–1243. IEEE, 2012.

[22] A Jessica Fridrich, B David Soukal, and A Jan Lukáš. Detection of copy-move forgery in digital images. In *Proceedings of Digital Forensic Research Workshop*. Citeseer, 2003.

[23] Jessica Fridrich. Digital image forensics. *IEEE Signal Processing Magazine*, 26(2):26–37, 2009.

[24] Jessica Fridrich and Jan Kodovsky. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 7(3):868–882, 2012.

[25] Simson Garfinkel. Anti-forensics: Techniques, detection and countermeasures. In *2nd International Conference on i-Warfare and Security*, volume 20087, pages 77–84, 2007.

[26] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 315–323. JMLR Workshop and Conference Proceedings, 2011.

[27] Rafael Gonzalez and Richard Woods. *Digital Image Processing*. Upper Saddle River, NJ: Prentice-Hall, Inc., 2002.

[28] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *ArXiv Preprint ArXiv:1406.2661*, 2014.

[29] David Güera and Edward J Delp. Deepfake video detection using recurrent neural networks. In *15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6, 2018.

[30] Mohammad F Hashmi, Aaditya R Hambarde, and Avinash G Keskar. Robust image authentication based on hmm and svm classifiers. *Engineering Letters*, 22(4), 2014.

[31] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer vision and Pattern recognition*, pages 770–778, 2016.

[32] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1125–1134, 2017.

[33] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM International Conference on Multimedia*, pages 675–678, 2014.

[34] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016.

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

[36] Dongkyu Kim, Han-Ul Jang, Seung-Min Mun, Sunghee Choi, and Heung-Kyu Lee. Median filtered image restoration and anti-forensics using adversarial networks. *IEEE Signal Processing Letters*, 25(2):278–282, 2017.

[37] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25:1097–1105, 2012.

[38] Lu Laijie, Yang Gaobo, and Xia Ming. Anti-forensics for unsharp masking sharpening in digital images. *International Journal of Digital Crime and Forensics (IJDCF)*, 5(3):53–65, 2013.

[39] Yann LeCun. Generalization and network design strategies. *Connectionism in Perspective*, 19:143–155, 1989.

[40] 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. *Neural Computation*, 1(4):541–551, 1989.

[41] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[42] Haoliang Li, Shiqi Wang, and Alex C Kot. Image recapture detection with convolutional and recurrent neural networks. *Electronic Imaging*, 2017(7):87–91, 2017.

[43] Yingmin Luo, Hanqi Zi, Qiong Zhang, and Xiangui Kang. Anti-forensics of jpeg compression using generative adversarial networks. In *26th European Signal Processing Conference (EUSIPCO)*, pages 952–956. IEEE, 2018.

[44] Siwei Lyu and Hany Farid. How realistic is photorealistic? *IEEE Transactions on Signal Processing*, 53(2):845–850, 2005.

[45] Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 1(10):e3, 2016.

[46] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51–59, 1996.

[47] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[48] Cecilia Pasquini and Giulia Boato. Jpeg compression anti-forensics based on first significant digit distribution. In *IEEE 15th International Workshop on Multimedia Signal Processing (MMSP)*, pages 500–505, 2013.

[49] Lionel Pibre, Jérôme Pasquet, Dino Ienco, and Marc Chaumont. Deep learning is a good steganalysis tool when embedding key is reused for different images, even if there is a cover sourcemismatch. *Electronic Imaging*, 2016(8):1–11, 2016.

[50] Alessandro Piva. An overview on image forensics. *International Scholarly Research Notices*, volume 2013, 2013.

[51] Yinlong Qian, Jing Dong, Wei Wang, and Tieniu Tan. Deep learning for steganalysis via convolutional neural networks. In *Media Watermarking, Security, and Forensics 2015*, volume 9409, page 94090J. International Society for Optics and Photonics, 2015.

[52] Zhenxing Qian and Xinpeng Zhang. Improved anti-forensics of jpeg compression. *Journal of Systems and Software*, 91:100–108, 2014.

[53] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *ArXiv Preprint ArXiv:1511.06434*, 2015.

[54] Yuan Rao and Jiangqun Ni. A deep learning approach to detection of splicing and copy-move forgeries in images. In *IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6, 2016.

[55] Krishna Regmi and Ali Borji. Cross-view image synthesis using conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3501–3510, 2018.

[56] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-assisted intervention*, pages 234–241. Springer, 2015.

[57] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, and Michael Bernstein. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[58] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *ArXiv Preprint ArXiv:1606.03498*, 2016.

[59] Gerald Schaefer and Michal Stich. Ucid: An uncompressed color image database. In *Storage and Retrieval Methods and Applications for Multimedia 2004*, volume 5307, pages 472–480. International Society for Optics and Photonics, 2003.

[60] Hamid R Sheikh and Alan C Bovik. Image information and visual quality. *IEEE Transactions on Image Processing*, 15(2):430–444, 2006.

[61] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *ArXiv Preprint ArXiv:1409.1556*, 2014.

[62] Kulbir Singh, Ankush Kansal, and Gurinder Singh. An improved median filtering anti-forensics with better image quality and forensic undetectability. *Multidimensional Systems and Signal Processing*, 30(4):1951–1974, 2019.

[63] Matthew C Stamm, W Sabrina Lin, and KJ Ray Liu. Forensics vs. anti-forensics: A decision and game theoretic framework. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1749–1752, 2012.

[64] Matthew C Stamm and KJ Ray Liu. Anti-forensics of digital image compression. *IEEE Transactions on Information Forensics and Security*, 6(3):1050–1065, 2011.

[65] Matthew C Stamm, Steven K Tjoa, W Sabrina Lin, and KJ Ray Liu. Undetectable image tampering through jpeg compression anti-forensics. In *IEEE International Conference on Image Processing*, pages 2109–2112, 2010.

[66] Yusuke Sugawara, Sayaka Shiota, and Hitoshi Kiya. Super-resolution using convolutional neural networks without any checkerboard artifacts. In *25th IEEE International Conference on Image Processing (ICIP)*, pages 66–70, 2018.

[67] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern recognition*, pages 1–9, 2015.

[68] Amel Tuama, Frédéric Comby, and Marc Chaumont. Camera model identification with the use of deep convolutional neural networks. In *IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6, 2016.

[69] David Vázquez-Padín, Fernando Pérez-González, and Pedro Comesana-Alfaro. A random matrix approach to the forensic analysis of upscaled images. *IEEE Transactions on Information Forensics and Security*, 12(9):2115–2130, 2017.

[70] Dongping Wang, Tiegang Gao, and Yuan Zhang. Image sharpening detection based on difference sets. *IEEE Access*, 8:51431–51445, 2020.

[71] Zhung-Han Wu, Matthew C Stamm, and KJ Ray Liu. Anti-forensics of median filtering. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3043–3047, 2013.

[72] Guanshuo Xu, Han-Zhou Wu, and Yun-Qing Shi. Structural design of convolutional neural networks for steganalysis. *IEEE Signal Processing Letters*, 23(5):708–712, 2016.

[73] Jingyu Ye, Zhangyi Shen, Piyush Behrani, Feng Ding, and Yun-Qing Shi. Detecting usm image sharpening by using cnn. *Signal Processing: Image Communication*, 68:258–264, 2018.

[74] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pages 818–833. Springer, 2014.

[75] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2223–2232, 2017.