Detection of computer-generated images via deep learning

Weize Quan

THÈSE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE ALPES

préparée dans le cadre d'une cotutelle entre l'*Université Grenoble Alpes* et *University of Chinese Academy of Sciences*

Spécialité : **Signal, Image, Parole, Télécoms (SIPT)**

Arrêté ministériel : 25 mai 2016

Présentée par

Weize QUAN

Thèse dirigée par **Denis PELLERIN** et **Xiaopeng ZHANG**
et co-encadrée par **Kai WANG** et **Dong-Ming YAN**

préparée au sein du **laboratoire Grenoble, Images, Parole, Signal, Automatique (GIPSA-lab)** et **National Laboratory of Pattern Recognition, Institute of Automation**

dans **l'école doctorale d'électronique, électrotechnique, automatique et traitement du signal (EEATS)** et **University of Chinese Academy of Sciences**

Detection of Computer-Generated Images via Deep Learning

Thèse soutenue publiquement le **15/12/2020**,
devant le jury composé de :

Changhe TU
Professeur, Université du Shandong, Examinateur, Président

Rongrong NI
Professeure, Université Jiao Tong de Pékin, Rapporteuse

Patrick BAS
Directeur de Recherche CNRS, CRIStAL, Rapporteur

Florent RETRAINT
Professeur, Université de Technologie de Troyes, ICD, Examinateur

Denis PELLERIN
Professeur, Université Grenoble Alpes, GIPSA-lab, Directeur de thèse

Xiaopeng ZHANG
Professeur, Académie Chinoise des Sciences, NLPR, Directeur de thèse

Kai WANG
Chargé de Recherche CNRS, GIPSA-lab, Co-encadrant, Invité

Dong-Ming YAN
Professeur, Académie Chinoise des Sciences, NLPR, Co-encadrant, Invité

UNIVERSITY OF GRENOBLE ALPES

Doctoral School EEATS

(Électronique, Électrotechnique, Automatique et Traitement du Signal)

T H E S I S

for obtaining the title of

Doctor of Science

of the University of Grenoble Alpes

Speciality: SIPT

(Signal, Image, Parole, Télécoms)

Presented by

Weize QUAN

Detection of Computer-Generated Images via Deep Learning

Thesis supervised by Denis PELLERIN and Xiaopeng ZHANG
and co-supervised by Kai WANG and Dong-Ming YAN

prepared at
Grenoble - Images, Parole, Signal, Automatique Laboratory (GIPSA-lab)
and National Laboratory of Pattern Recognition, Institute of Automation
presented on 15/12/2020

Jury:

President:	Changhe TU	-	Université du Shandong
Reviewers:	Rongrong NI	-	Université Jiao Tong de Pékin
	Patrick BAS	-	CNRS, CRIStAL
Examiner:	Florent RETRAINT	-	Université de Technologie de Troyes
Supervisors:	Denis PELLERIN	-	Université Grenoble Alpes, GIPSA-lab
	Xiaopeng ZHANG	-	Académie Chinoise des Sciences, NLPR
Co-Supervisors:	Kai WANG	-	CNRS, GIPSA-lab, Invited
	Dong-Ming YAN	-	Académie Chinoise des Sciences, NLPR, Invited

Acknowledgements

First and foremost I would like to express my sincere gratitude to my four thesis supervisors: Prof. Denis Pellerin and Dr. Kai Wang in Gipsa-Lab, Prof. Xiaopeng Zhang and Prof. Dong-Ming Yan in NLPR, Institute of Automation. I am very lucky to conduct a co-supervised Ph.D. thesis between the University of Grenoble Alpes and the University of Chinese Academy of Sciences, and this precious opportunity depends entirely on the support and help of my supervisors. I am also grateful to Prof. Feng Gang for his help and advice during the whole training process.

I would like to thank Prof. Pellerin for his guidance on my research work. He always makes valuable suggestions in our discussions and sometimes is a strict "reviewer". I would like to thank Prof. Zhang for his constant support. He helps me solve difficulties in my study and life, and provides convenient conditions for my research. My sincere gratitude goes to Dr. Wang for his patient guidance. He often proposes unique insights for my encountered problems, teaches me how to tell a story and polish my papers, and guides me to think about some aspects that are not easy to notice. More importantly, he always encourages me to go further when I might give up. I also would like to sincerely thank Prof. Yan for his hard work. For each of my research projects, he carefully guides every aspect of problem analysis, experiment setting, result analysis and paper writing. He teaches me how to be a researcher.

I would like to express my sincere thanks to Prof. Rongrong NI and Prof. Patrick BAS, for devoting your precious time to review my thesis manuscript. I would like to sincerely thank Prof. Changhe TU and Prof. Florent RETRAINT, for being examiners of the defense committee. I also would like to thank all the members in jury for your sacrifice about the defense time due to COVID-19.

I want to thank my colleagues, Ludovic, Ivan, Julien, Tien and Dawood in GIPSA-lab, for helping me in the work and life. I also want to thank my new friends, Jingtao, Dacheng, Jianze, Li, Yang, Bo, Yanglv, Xiaofei, Yutong, Xiaotong, Huihui, Xiaohong, Xu, Beibei, Yuan and Peng, for making my life in Grenoble much richer. They have helped me in many aspects of my daily life, taking me to the doctor, solving some administrative affairs, etc. And I cherish the days of playing basketball with some of you.

Finally, I want to thank my parents, wife and sister for your unconditional support and love. You are my strong backing forever.

Contents

List of Figures

List of Tables

Acronyms

ASM	active shape model
AutoNet	automatic network
BCP	bright channel prior
BN	batch normalization
CFA	color filter array
CI	colorized image
CNN	convolutional neural network
CG	computer graphics
CT	contourlet transform
DCP	dark channel prior
DFT	discrete Fourier transform
DWT	discrete wavelet transform
ECP	extreme channels prior
ENet	ensemble network
FC	fully-connected
FFT	fast Fourier transform
FGSM	fast gradient sign method
GAN	generative adversarial network
HTER	half total error rate
IMGSM	iterative masked gradient sign method
LBP	local binary pattern
LGS	local-to-global strategy
LRP	layer-wise relevance propagation

LSTM	long-short-term-memory
MPS	maximal Poisson-disk sampling
NcgNet	natural and computer graphics network
NI	natural image
PI	preferred inputs
PRNU	photo response non-uniformity noise
QWT	quaternion wavelet transform
ReLU	rectified linear unit
RIT	ridgelet transform
RNN	recurrent neural network
SFS	sequential floating search
SGD	stochastic gradient descent
SRM	spatial rich model
SVM	support vector machine
t-SNE	t-distributed stochastic neighbor embedding
WISERNet	wider separate-then-reunion network

Abstract

With the advances of image editing and generation software tools, it has become easier to tamper with the content of images or create new images, even for novices. These generated images, such as computer graphics (CG) image and colorized image (CI), have high-quality visual realism, and potentially throw huge threats to many important scenarios. For instance, the judicial departments need to verify that pictures are not produced by computer graphics rendering technology, colorized images can cause recognition/monitoring systems to produce incorrect decisions, and so on. Therefore, the detection of computer-generated images has attracted widespread attention in the multimedia security research community. In this thesis, we study the identification of different computer-generated images including CG image and CI, namely, identifying whether an image is acquired by a camera or generated by a computer program. The main objective is to design an efficient detector, which has high classification accuracy and good generalization capability. Specifically, we consider dataset construction, network architecture, training methodology, visualization and understanding, for the considered forensic problems. The main contributions are: (1) a colorized image detection method based on negative sample insertion, (2) a generalization method for colorized image detection, (3) a method for the identification of natural image (NI) and CG image based on CNN (Convolutional Neural Network), and (4) a CG image identification method based on the enhancement of feature diversity and adversarial samples.

Keywords: Image Forensics, Deep Learning, Computer-Generated Image, Colorized Image, Generalization, Trustworthiness

Chapter 1

Introduction, State of the Art and Objectives

Contents

1.1 Background

Digital image, because of its directness and understandability, makes it an efficient and natural communication medium. Historically, the authenticity of image data is real and reliable. For example, a photo printed in a newspaper can be widely accepted as a proof of news; or, video surveillance records are proposed as important materials in court. Today, with the low cost and simplification of acquiring devices, such as smart phones and digital cameras, almost everyone can record, store and share a large number of images/videos anytime and anywhere. In the meanwhile, lots of image editing softwares/tools also make it extremely simple to modify image content or create new images. In consequence, the possibility of tampering and forging visual content is no longer limited to experts. Digital technology has begun to weaken the degree of trust in visual content, and it is obvious that "what you see is no longer trustworthy." As shown in Figure 1.1, this is a highly-realistic image composed of 16 different photos[1]. With the advance and complexity of processing tools, all these problems become more and more urgent, which has prompted the progress of research on digital image forensics. The core and goal of image forensics are to restore some trust to digital images. Generally speaking, the main purposes of image forensics are to analyze a given digital image so as to detect whether it is a forgery,

[1]This image comes from http://commons.wikimedia.org/wiki/User:Mmxx, author: mxx.

Figure 1.1: A highly-realistic fake image composed of 16 different photos. Used software: Adobe Photoshop® [Pho].

to identify its origin, to trace its processing history, or to reveal potential details invisible to the naked eyes [Fan15].

In the past two decades, researchers have proposed various image forensic techniques. In the early stage and even in many current applications, fragile digital image watermarking is a popular way to prove the authenticity and integrity of images [Con11; Piv13]. This is essentially an *active forensics* technology. Specifically, this kind of technology actively embeds identification information (*i.e.*, digital watermark) into the image when the image is captured or before it is transmitted. In the forensic stage, the watermark information is firstly extracted: if the extracted watermark matches the embedded watermark, the authenticity of the image is proved; if the watermark extraction fails or the extracted information does not match the embedded information, the image has been tampered. The concept of trusted camera [Fri93; BF04] was proposed for the purpose of active image forensics. The digital camera is equipped with a special watermark chip. On the one hand, this process will inevitably affect the quality of the photo itself; on the other hand, it also brings some inconveniences to camera manufacturers, such as the need to develop a standard and the security of the camera itself. The high hardware cost, low detection efficiency, and non-uniform standards of the active forensics equipment limit the practical application scenarios of this solution.

Considering the limitations of active forensics, researchers in the field of multimedia security have gradually shifted their attention to a new research direction/technology, namely, *passive forensics*. Compared with active forensics, passive forensics technology does not require any prior information (such as watermarks or signatures) [LUO+07; Far09]. These techniques are usually based on the assumption that although digital forgeries may not leave any visual traces of tampering, they may change the inherent statistical properties of the image. Currently, passive forensics has become the main research paradigm in the field of image forensics, which mainly includes image identi-

fication, forgery detection and localization, image processing history recognition, and image source camera identification, etc. Figure 1.2 summarizes these research problems. Loosely speaking, for the image identification, the main purpose is to distinguish between natural image (NI) and computer graphics (CG) image[2] [FL03; LF05; Ng+05], colorized image (CI) [Guo+18; YRC19], recaptured image [Wan17; ZQY19], and the latest generated image [Li+18; ZKC19; YDF19] based on the generative adversarial network (GAN) [Goo+14] and so on. Image tampering detection and localization focuses on identifying whether the content of the image has been maliciously tampered, and at the same time locating the tampered region. Common tampering operations include copy-move [Chr+12; Li+15; LZ19], object removal [Zho+18; Bap+19], and splicing [Zha+09; Zha+10; Liu+11; Cao+15; RN16; Yao+17], etc. The recognition of image processing history mainly includes detection of median filtering [KF10; CNH13; Che+15], JPEG compression [FQ03; PF08; HHS10; Yan+14; Niu+19], resampling [PF05; FCD12; RL14], contrast enhancement [Cao+14; WQL18], etc. The research on the forensic problems of these image processing operations can assist in the detection and localization of image tampering to a certain extent. The identification of the image source camera is usually to study the camera model of the acquired image and related research [LFG06; XS12; TCC16; Bon+17; Yan+19; MS20].

This thesis mainly studies the identification of computer-generated images, including the classification of natural images (NI) and computer graphics (CG) images (referred to as CG image forensics), and the classification of natural images and colorized images (CI) (referred to as CI detection). Here, natural images refer to pictures captured by a digital camera. The former is an important and relatively long-existing research problem in the field of image forensics, and researchers have previously conducted a large amount of works [LF05; Ng+05]; the latter is an emerging forensic research problem [Guo+18; YRC19]. Figure 1.3(a) shows two high-quality CG images[3]; (b) shows two visually-realistic colorized images, which are obtained using an advanced automatic colorization algorithm [ISSI16]; (c) is the corresponding original natural images of images in (b), where (b) and (c) share same grayscale information. In fact, it is difficult for human observers to determine whether the images in (a) and (b) were captured by a camera (*i.e.*, natural images). These forensic issues have important research significance in the fields of public security, justice, and entertainment. At the same time, recently deep learning has achieved rapid development under the promotion of the industry and the extensive attention of researchers. Due to outstanding performance and simplicity of implementation, deep learning has been applied in many research fields, such as computer vision, computer graphics, natural language processing, and multimedia security.

[2] CG image refers to the image rendered by computer graphics techniques.
[3] Images come from https://area.autodesk.com/fakeorfoto/

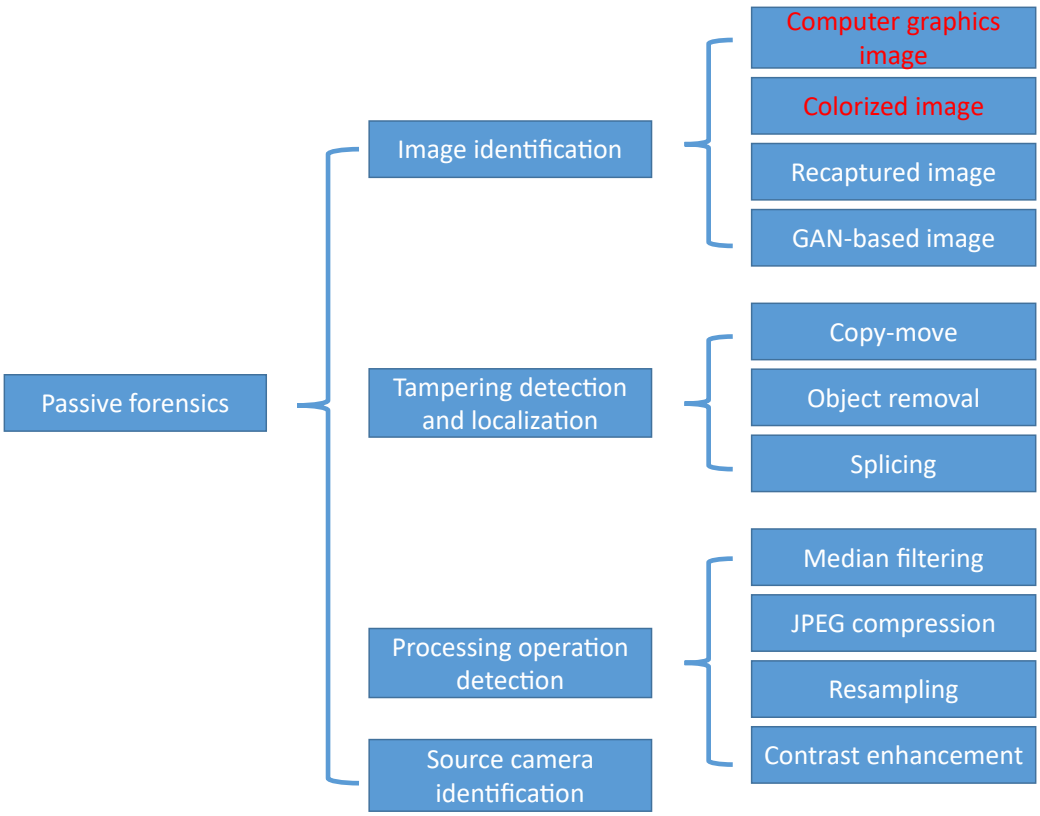


Figure 1.2: A diagram summarizing passive forensics methods.

Although researchers in the image forensics community have made big progress, there are still many difficulties and challenges in this field.

First of all, many previous identification methods adopt the traditional two-stage machine learning framework, namely, hand-crafted feature extraction and classifier training. This kind of method usually achieves good results on relatively simple datasets. However, their performance is often limited on more complex data, and some works try to improve the classification accuracy via a combination/fusion of multiple features which may not always be an efficient solution. In addition, how to make full use of traditional features or traditional filters is still an open problem worth studying.

Secondly, considering the limitations of the framework based on hand-crafted features, some recent research efforts have been devoted to utilizing deep learning methods, which generally achieved the state-of-the-art performance. However, some questions are worth investigating regarding the trustworthiness and understanding of such methods. For instance, what is the CNN (Convolutional Neural Network) model using as the discriminative information, *i.e.*, is it the "essential" difference between different kinds of

(a) Computer graphics images

(b) Colorized images

(c) Natural images

Figure 1.3: Examples of CG images and CIs: (a) Computer graphics images; (b) Colorized images; (c) Natural images.

images? Is the CNN just overfitting on training data in some aspects that are not the primary factors for the considered forensic problem? How can CNN generalize well on "unknown" data during the testing stage? Are there potential pitfalls behind the high performance?

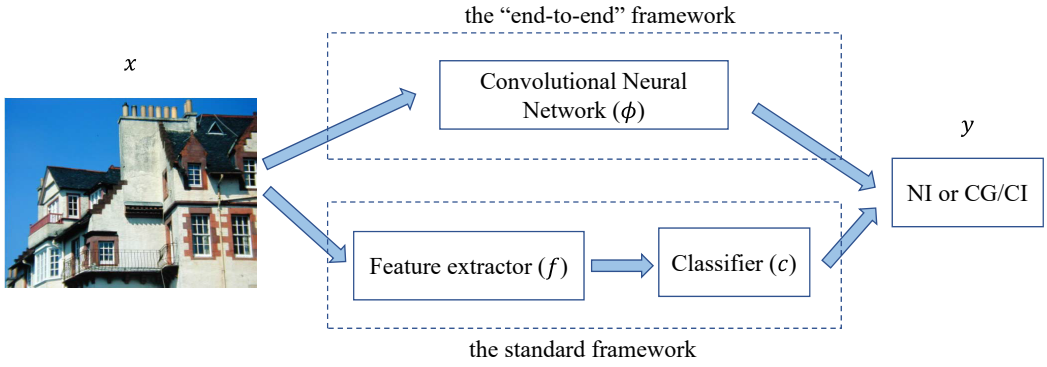


Figure 1.4: Two different frameworks for the image identification problem.

Finally, computer graphics technology, colorization technology and other generation technologies are constantly updated and developed. In particular, with the recent and popular tool of generative adversarial networks (GANs), it is increasingly easy to generate high-quality fake images that can deceive the human visual system. Therefore, how to improve the generalization (or blind detection) capability of forensic detectors has become an important and urgent research problem. Current research mainly focuses on improving the detection performance of forensic detectors (under an ideal experimental environment). However, when forensic detectors are deployed in the real-world scenarios, they will inevitably encounter the generalization problems, that is, methods/images to be detected are "unknown" to the trained detector. At present, few researches focus on such generalization problems in the field of image forensics.

1.2 Prior Art

Similar to other forensic problems, distinguishing between natural images and computer-generated images usually is modeled as a binary classification problem. Given the set of training data$\{(x^1, y^1), (x^2, y^2), \cdots, (x^N, y^N)\}$ of N samples, where x stands for the image and y corresponds to its label (1:NI, 0: generated image, *e.g.*, CG image, CI, etc), the main goal is to find a good mapping function $\phi : y = \phi(x)$ using the given training samples.

For the computer-generated image identification problem, most existing methods follow two different frameworks, as shown in Figure 1.4. The standard framework (bottom of Figure 1.4) is to find a mapping $y = c(f(x))$, where f is a well-designed feature extractor and c stands for a classifier, such as support vector machine (SVM), or in a larger sense, it can also stand for thresholding for scalar features. This framework is a classical two-stage model, and its core is the feature extractor. However, hand-crafted features are

often time-consuming and tedious to design and not necessarily the most adequate ones, as in general it is difficult to extract and organize the discriminative information from the data [BCV13]. A generic "end-to-end" framework (top of Figure 1.4), such as CNN, becomes the main tendency of current research. Given a testing image, a well-trained CNN can directly and accurately predict its label in an "end-to-end" manner.

1.2.1 Detection of Colorized Images

This section will first review the representative colorization algorithms and then present the existing detection methods of colorized images.

1.2.1.1 Image Colorization

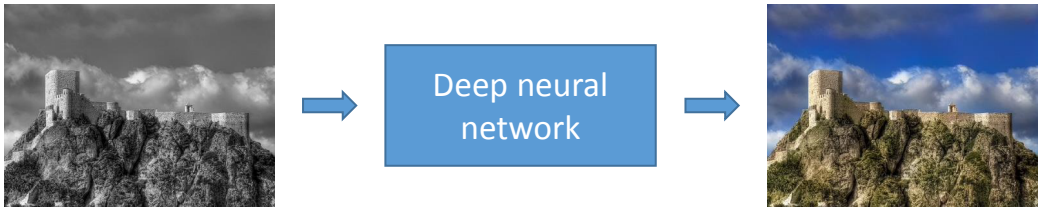
Image colorization adds color to a monochrome image and obtains a realistic color image. Existing colorization algorithms mainly consist of three categories: scribble-based [LLW04; Lua+07; Xu+09; Che+12; Pan+13], reference-based [WAM02; ICOL05; Gup+12; He+18a], and fully automatic [ISSI16; LMS16; ZIE16] approaches. Figure 1.5 illustrates the main pipeline of these three kinds of methods.

Scribble-based methods require user-specific scribbles and propagate the color information to the whole grayscale images. This is based on a simple assumption: adjacent pixels with similar intensities should have similar colors. Levin et al. [LLW04] proposed an optimization-based method to complete the colorization task. Specifically, they formalized the problem with a quadratic loss function, which can be solved efficiently using standard techniques. Beside the intensity similarity, Luan et al. [Lua+07] also considered the texture similarity, and proposed a novel energy optimization framework combining the intensity continuity and texture similarity constraints, which can aggregate images as multiple coherent regions to carry out the colorization. By approximately solving the optimization problem in edit propagation, Xu et al. [Xu+09] significantly improved the efficiency of the algorithm. Chen et al. [Che+12] introduced the locally linear embedding constraint to edit propagation, whose core is to maintain the manifold structure formed by all pixels in the feature space. In addition, since each pixel is only related to a few adjacent pixels, their algorithm can achieve good operational efficiency. In general, this kind of method is usually accompanied by trail and error to obtain satisfactory results, and thus is rather time-consuming.

Reference-based (or exemplar-based) approaches mainly exploit the color information of a reference image that is (semantically) similar to the input grayscale image. The key idea is to model a matching relationship between these two images. By matching the

(a) Scribble-based colorization

(b) Reference-based colorization



(c) Fully automatic colorization

Figure 1.5: Three kinds of colorization methods: (a) scribble-based; (b) reference-based; (c) fully automatic. Images in (a) come from [LLW04], and images in (b) come from [He+18a].

brightness and texture information of the image, Welsh et al. [WAM02] transferred the entire color information from the original image to the target image. Irony et al. [ICOL05] considered spatial consistency instead of single-pixel independent decision. Specifically, their method first automatically finds for each pixel a matched image patch from the reference image, and then combines the neighborhood matching measure and spatial filtering to add an appropriate color as well as a confidence level to each pixel. Finally, the high-confidence pixel is used as scribble and the method applies the optimization framework of Levin et al. [LLW04] to complete the colorization. Gupta et al. [Gup+12] first performed super-pixel segmentation on the image, and then extracted features and carried out matching. At the same time, the voting mechanism of the image space was used to improve spatial consistency. This super-pixel-based method can speed up the colorization process. He et al. [He+18a] adopted deep neural networks to further improve the visual quality of colorized images. However, the selection of suitable reference image

Figure 1.6: From left to right: a natural image taken from ImageNet [Den+09]; three colorized images generated by the colorization method proposed in [LMS16], [ZIE16], and [ISSI16], respectively.

may be burdensome.

In contrast, recently researchers have developed fully automatic methods that do not need user interaction or example color images. Cheng et al. [CYS15] proposed the first deep neural network based image colorization method. Their method performed pixel-wise prediction, however, the input of deep model was pre-extracted hand-crafted features. Iizuka et al. [ISSI16] proposed a novel fully "end-to-end" network for the task of image colorization. The input was a grayscale image and its output was the chrominance, which was combined with the input image to produce the color image. Their network jointly learned global and local features from an image, and at the same time, they also exploited classification labels of the grayscale images to improve the performance. Different from previous methods, Larsson et al. [LMS16] proposed a deep model that predicted a color histogram, instead of a single color value, at every image pixel. Zhang et al. [ZIE16] took into account the nature of uncertainty of this colorization task and introduced class-rebalancing method to increase the diversity of color of resultant image. These CNN-based methods lead to the very high visual quality of colorized images, often plausible enough to deceive the human perception. Figure 1.6 shows a group of images, the left-most one is the original color image taken from ImageNet [Den+09], and the remaining three are colorized images produced by three state-of-the-art colorization algorithms: hereafter named as Ma [LMS16], Mb [ZIE16], and Mc [ISSI16], respectively.

1.2.1.2 Detection Methods of Colorized Images

For the colorized image detection, there mainly exist two types of methods: hand-crafted-feature-based method [Guo+18] and CNN-based method [Zhu+18; YRC19; Li+19]. Guo et al. [Guo+18] first proposed hand-crafted-feature-based methods to detect fake colorized images. On the basis of the observation that colorized images tend to possess less

saturated colors, they analyzed the statistical difference between NIs and CIs in the hue and saturation channels. In addition, they also found that there are differences in certain image priors. In practice, they exploited the extreme channels prior (ECP) [Yan+17], *i.e.*, the dark channel prior (DCP) [HST11] and the bright channel prior (BCP). They proposed two approaches, *i.e.*, histogram-based and Fisher-encoding-based, to extract statistical features, and then trained SVMs for classification. Later, Zhuo et al. [Zhu+18] greatly improved the detection performance using a CNN-based color image steganalyzer WISERNet (WIder SEparate-then-Reunion Network) [Zen+19]. Yan et al. [YRC19] designed a deep network to recognize recolorized images. The network contains three feature extraction modules with different inputs, namely, the original input image, the color channel difference image and the illuminance map based on segmentation. Accordingly, their network also has a feature fusion module. Li et al. [Li+19] first used cosine similarity to measure the similarity of the normalized histogram distribution of different channels, and then performed feature extraction. Finally, a deep neural network was used to carry out the classification.

As mentioned in the Section 1.2.1.1, the image colorization algorithm essentially reconstructs its color information from the grayscale image. Therefore, compared with natural images, colorized images will inevitably have some differences, such as under-saturated colors, statistical correlation of the three color channels, and so on. Considering these color and statistical differences, the previous works proposed the hand-crafted-feature-based and CNN-based approaches, and achieved good detection performance.

However, there is a limitation in the existing works. When the training image and the testing image come from different colorization algorithms, the performance of the CNN-based method [Zhu+18] and the hand-crafted-feature-based method [Guo+18] in general decreases. This thesis defines it as a blind detection scenario, that is, no training sample is available from "unknown" colorization methods that we may encounter during the testing phase of forensic detectors. Hereafter, we call this blind detection performance as *generalization* performance. Take Figure 1.6 as an example, the second and fourth images (produced by Ma [LMS16] and Mc [ISSI16]) are misclassified as NI by a CNN model trained on NIs and CIs generated by Mb [ZIE16]. Chapters 2 and 3 will focus on this issue. Although not being very rigorous, in the following, the term "classification accuracy/performance" refers to the detection performance on testing data in which CIs are generated by a same colorization method known by the training stage.

1.2.2 Identification of Natural Images and Computer Graphics Images

For the discrimination of natural versus CG images and videos, there mainly exist two lines of research, namely, (1) subjective, perceptual studies and (2) objective studies.

Subjective studies involve performing a series of psychophysical experiments to study the effects of image properties and cognitive characteristics of human observers on the discrimination between photorealistic and photographic images. Objective studies usually depend on the statistical or intrinsic properties of natural and CG images or videos and design efficient algorithms to separate them. For the objective studies, two types of methods have been proposed: hand-crafted-feature-based methods and CNN-based methods. The former follows the two-stage framework composed of feature extraction and classifier training, the latter adopts the data-driven "end-to-end" framework. In the following, we review and summarize these existing methods.

1.2.2.1 Subjective Study

In 1996, the United States Congress passed The Child Pornography Prevention Act which, in part, prohibited any image that appears to be or conveys the impression of someone under 18 engaged in sexually explicit conduct. This law made illegal computer generated pictures that only appear to show minors involved in sexual activity. In 2002, however, the United States Supreme Court struck down this law, and said that language in the 1996 child pornography law was unconstitutionally vague and far-reaching [LF05]. This ruling led law enforcement agencies, such as judges, lawyers, and juries, to determine whether the image is rendered, but there is no data to show that they can do this reliably. To test the ability of human observers to distinguish between computer graphics images and natural images, Farid and Bravo [FB07] collected 180 high-quality CG images, which contain human, artificial or natural content. At the same time, they also collected 180 natural images matching the content of CG images. They reported that human observers have the ability to distinguish NI from CG images. It is also found that the recognition rate will decrease for CG images created in 2006.

Based on this earlier work, Farid and Bravo [FB12] focused exclusively on images of people, and explored the impact of the variations in image quality that arise in real-world settings: (1) resolution, (2) JPEG compression, and (3) color vs. grayscale. In the meanwhile, they updated the CG images to include images rendered between 2007 and 2010. The people depicted in these images vary in age, gender, race, pose, and lighting. The experimental result shows that observers consistently perform better at one-half resolution, and the recognition ability will drop for lower or higher resolution. One reason is that the fine details in computer generated images with high resolution are very accurate and observers take their presence as evidence of a photographic image. When the JPEG compression is stronger, the human observation ability is worse. In addition, the recognition accuracy of RGB images by human observers is higher than the gray-scale version.

In the same year, Fan et al. [Fan+12b] studied the effects of the observer's cognitive characteristics and image attributes (*i.e.*, color and shadow). Experimental results show that visual realism depends not only on image attributes, but also on the cognitive characteristics of the observer. Shadows are essential for visual realism. In addition, the performance of experts is better than that of non-professionals, but only for gray-scale images. Holmes et al. [HBF16] found that human observers have a certain degree of deviation in recognizing photography and CG portraits, and the observers are more likely to choose the latter. However, this bias can be greatly reduced by conducting a small amount of training before the main experiment. On the basis of previous studies, Mader et al. [MBF17] described a series of experiments that revealed how to improve the recognition ability of the observer. This work demonstrates that introducing appropriate training, feedback and incentive measures can further enhance the observer's classification ability.

1.2.2.2 Hand-crafted-feature-based Method

The traditional hand-crafted-feature-based methods for detecting CG images can mainly be divided into two categories: spatial domain method [Ng+05; GC08; PCH09; SZY09; ZWN12; LYS13; PZ14; PLL14; Wan+14; Pen+17] and transform domain method [LF05; CSX07; Che+09; OA11; Fan+12a; Wan+17]. Table 1.1 summarizes the detailed information, such as the key idea and the dimension of features.

The spatial domain method mainly analyzes some statistical differences and texture details in the image space, and uses some geometric methods as well. Inspired by the generation process of natural and computer-rendered images, specially object model, light transport, and acquisition differences, Ng et al. [Ng+05] proposed geometry-based features aided by fractal and differential geometry. To assess this approach, they created an open dataset, *i.e.*, Columbia Photographic Images and PRCG (PhotoRealistic Computer Graphics) Dataset [Ng+04], comprising: (1) 800 PRCG images from 40 3D graphic websites (PRCG), (2) 800 NIs from the authors' personal collections (Personal), and (3) 800 photographic images from Google Image Search (Google). Gallagher and Chen [GC08] detected traces of demosaicing of original camera images to distinguish camera images from computer graphics and reported a good forensic performance. However, this method may be sensitive to postprocessing operations, such as resizing, which can remove the demosaicing interpolation structure [NC13]. From the point of view of image perception, Pan et al. [PCH09] captured the difference in color perception and coarseness between CG images and natural images. In details, the fractal dimensions are derived from the hue and saturation channels of image, the generalized dimensions are calculated on hue component gradient, and then combining these to construct fea-

Table 1.1: The hand-crafted-feature-based methods for CG image forensics.

Type	Author	Year	Method	#Feature
Spatial domain	Ng et al. [Ng+05]	2005	Geometry-based	192
	Pan et al. [PCH09]	2009	Fractal and generalized dimension	30
	Sanker et al. [SZY09]	2009	Combined features	557
	Zhang et al. [ZWN12]	2012	Visual vocabulary; local image edges	256
	Li et al. [LYS13]	2013	YCbCr; local binary patterns	236
	Peng and Zhou [PZ14]	2014	CFA; PRNU	9
	Peng et al. [PLL14]	2014	Statistical and textural Features	31
	Wang et al. [Wan+14]	2014	Homomorphic filtering; statistical	70
	Peng et al. [Pen+17]	2017	Multi-fractal and regression analysis	24
Transform domain	Lyu and Fraid [LF05]	2005	DWT; higher order	216
	Chen et al. [CSX07]	2007	DWT; DFT; HSV	234
	Chen et al. [Che+09]	2009	Fractional lower order statistics	243
	Özparlak et al. [OA11]	2011	RIT; CT; SFS	768
	Fan et al. [Fan+12a]	2012	HSV, CT	16
	Wang et al. [Wan+17]	2017	QWT	576

ture vector. The former describes the global color distribution, and the latter measures the detailed texture difference. Finally, a SVM is trained for classification with the grid searching. On the basis of several previous studies, Sankar et al. [SZY09] proposed a set of combined features, including periodic-correlation-based feature [PF05], color histogram feature [IVR03], moment-based statistical feature in the YCbCr color space [CSX07], and local patch statistics [Ng+05]. Zhang et al. [ZWN12] proposed a method that analyzed the statistical property of local image edge patches. First, a visual vocabulary on local image edges was constructed with the aid of Voronoi cells. Second, a feature vector was formed with a binned histogram of visual words. Finally, an SVM classifier was trained for image classification. Li et al. [LYS13] explored the statistical difference of uniform gray-scale invariant local binary patterns (LBP) to distinguish CG image from photographic images. Their method selected YCbCr as the color model, the JPEG coefficients of Y and Cr components, and their prediction errors are used for LBP calculation. These LBP features are finally used for SVM classification. Considering the generation process of natural images, Peng et al. [PZ14] studied the impact of color filter array (CFA) interpolation on the local correlation of photo response non-uniformity noise (PRNU), and extracted histogram features from the local variance histograms of PRNU for identification. In addition, Peng et al. [PLL14] captured statistical features (the mean and variance of the relative frequency of gray-scale images) and texture features for classification. Based on the detail difference between NI and CG images, Wang et al. [Wan+14] conducted homomorphic filtering for input image, computed texture similarity from the

difference matrix of original image and filtered image, and then extracted the statistical information from the difference matrix of Contourlet decompositions of these two images. These two discriminative features were used for classification. Recently, Peng et al. [Pen+17] used a linear regression model to extract the residual of a Gaussian low-pass-filtered image and combined the histogram statistics and multi-fractal spectrum of the residual image with the fitness of the regression model as a feature to discriminate between NIs and CG images.

The transform domain method is mainly to transform the image space to the frequency space, thereby exposing some forensic traces for classification tasks. Lyu and Farid [FL03; LF05] proposed a feature combining the first four order wavelet statistics, *i.e.*, mean, variance, skewness, and kurtosis (computed from first three level of wavelet coefficient and the first two level of prediction error). Besides, they both tested the identification performance of linear discrimination analysis and SVM classifier. Chen et al. [CSX07] proposed discrete wavelet transform (DWT)-based and discrete Fourier transform (DFT)-based forensic method, which conducted in the HSV color space. More specifically, they separately calculated DFT of the histogram of wavelet coefficients and prediction error, and then extracted moment statistics as discriminative features. SVM is used for final classification. Chen et al. [Che+09] built the alpha-stable distribution model to characterize the wavelet decomposition coefficients, and used fractional lower order moments to construct features. The experimental results showed that this proposed method performs better than the previous higher-order statistical approaches. Instead of using DWT, Özparlak et al. [OA11] extracted features from the ridgelet transform (RIT)-based and contourlet transform (CT)-based image model. In addition, they introduced the sequential floating search (SFS) to select feature, and further improved identification performance. Later, Fan et al. [Fan+12a] used different contourlet wavelet models and HSV color model. Considering the fact that the DWT feature for forensics suffers from some drawbacks of DWT, *i.e.*, oscillations, shift-variance, and lack of directionality, and two drawbacks of CWT, *i.e.*, discontinuity of local phase and directionality redundancy, Wang et al. [Wan+17] introduced the quaternion wavelet transform (QWT) to solve these issues. Compared with DWT and CWT, QWT includes not only the magnitude which encodes the frequency information but also three phases which indicate richer edge and texture information. Those three phases contain extra information that is not included in high frequency subbands of DWT and CWT, and thus have obtained better performance.

Besides the above general cases of CG image forensics (the image contains various scenes, *e.g.*, indoor, outdoor, etc), there are some specific scenarios, especially CG character identification. For this kind of problem, a simple and popular strategy is to find a class-sensitive quantity and select an appropriate threshold for classification. Synthetic

expressions usually contain some repetitive patterns, and in natural human faces, the same expressions are usually produced in a similar but not equal way. Based on this discrepancy, Dang-Nguyen et al. [DNBDN12b] distinguished CG characters from real ones by analyzing variations in facial expressions. This method contains five steps: human faces extraction, facial expression recognition, active shape model (ASM) extraction, normalized face computation, and variation analysis. An appropriate threshold was finally used for classification. Conotter et al. [Con+14] identified CG faces in videos by detecting a physiological signal resulting from human pulse, which was absent in videos of CG faces. The features extracted by such methods can also be combined with classifier training. Another approach is based on face symmetry. On the one hand, if a given face presents a high symmetric structure, this could be considered as a hint that it is generated via computer. On the other hand, although human faces are symmetric, there does not exist a perfectly symmetrical face. Having observed these two points, Dang-Nguyen et al. [DNBDN12a] proposed an asymmetry-information-based method to discriminate between natural and CG human faces. This method contains three main steps: shape normalization, illumination normalization and asymmetry estimation. At last, a specific threshold is used to carry out classification. Besides, this feature can be added to other feature sets to improve their performance by using SVM binary classification. To distinguish between natural and CG faces in videos, Dang-Nguyen et al. [DNBDN15] examined the spatial-temporal variation of 3D face models, and defined a metric that can be used to measure the diversity in animation patterns. The underlying idea is that the variations in real faces are more complex than those in CG faces. The latter often follows repetitive or fixed patterns. More specifically, this method associates a 3D model to the face to be analyzed and maps various instances of the face in the video to the model. Then, it computes a set of parameters associated to the relevant deformation patterns. Finally, it estimates the variation of the geometric distortion parameters along time to achieve a measure of the diversity, thus leading to the classification of the face as synthetic or natural.

1.2.2.3 Deep-learning-based method

Inspired by the notable success of CNN in the field of computer vision and pattern recognition, some recent works also applied CNN to solve the CG image forensic problem [Rah+17; Yao+18; He+18b; NYE19; BT+19].

By analyzing the difference between deep network and traditional hand-crafted-feature-based methods, Rahmouni et al. [Rah+17] used convolutional layers to replace the traditional filter layer, and designed a specific pooling layer to replace the maximum pooling layer to extract the statistical information of the convolved image, including the

mean, variance, maximum and minimum, as feature vector. Finally a multi-layer perceptron is used to complete the classification task. Experimental results show that these simple statistics are better than more complex histogram statistics. Yao et al. [Yao+18] proposed a method based on sensor pattern noise and CNN to solve this task. In their method, they used several high-pass filters, on the one hand to remove low-frequency signals (image content), on the other hand to enhance residual signal as well as sensor pattern noise introduced by the digital camera devices. At the same time, they found that the performance of using three sets of high-pass filters is better than using one set of filters. He et al. [He+18b] mainly focused on two important forensic clues, color and texture, to detect CG images. An input image is first converted from RGB to YCbCr, and then the Schmid filter bank is used to enhance the texture information of the luminance component Y. After that, the color component and the brightness component are respectively fed into the two convolutional networks to learn the joint feature representation of the local image blocks. The recurrent neural network (RNN) [Shu+16] uses these features as input to model local and global statistics, thereby achieving classification. This work is also the first attempt to introduce RNN to CG image detection task. Considering the capability of capsule network to model image spatial information, Nguyen et al. [NYE19] first tried to use capsule network [SFH17] to solve the problem of CG image forensics. They used the first half of VGG-19 [SZ14] to extract hidden layer features, and then took these features as the input of the capsule network. The entire capsule network contains 3 main capsules and 2 output capsules (one corresponds to the true image and the other corresponds to the fake image). In addition, in the training stage, they slightly improved Sabour et al.'s training method [SFH17] by adding Gaussian random noise to the 3D weight tensor and used an additional squash function. This can reduce overfitting and stabilize network training. Most previous methods based on hand-crafted features or CNN uniformly process the pixels of the entire input image, and these technologies usually require a large computational cost. Therefore, Bhalang Tarianga et al. [BT+19] proposed a recursive model based on the attention mechanism to classify computer graphics images and natural images. At each time step, the model selectively processes an image region, then uses a small CNN network to extract features, and updates the internal state of the recurrent network (stacked long-short-term-memory unit [LSTM]). Afterwards it uses another CNN network to predict the location of the next image area. In the testing phase, the model gradually combines multiple steps of information to obtain the final prediction result.

From the above presentation, we can see that the method based on subjective experiments mainly studies the discrimination ability of human observers on CG images, and useful research results have been obtained. For example, proper training and feedback will enhance the discrimination ability of human observers. In addition, from the perspective of computer graphics, these quantitative measurements of realism also provide

valuable information and guidance for enhancing the fidelity of rendering technology. In the meanwhile, these studies also show that the discrimination ability of human observers is limited, and cannot quickly and massively identify CG images. Therefore, research and development of advanced computational methods is a more appropriate choice. The approach based on hand-crafted features is mainly to extract statistical information in the spatial domain or the transform domain to distinguish natural images from CG images. The basic process is that the researcher first analyzes the problem or data in hand, designs some possible statistical discriminative information, and then uses mathematical models for abstraction and modeling, so it is interpretable and understandable to some degree. In addition, this type of method has some advantages when the data scale is small. However, methods based on hand-crafted features also have some shortcomings. First, most methods only obtain one aspect of features, and cannot fully reflect the difference between natural images and computer graphics images. Since the content of the image is rich, if only specific aspects of the feature are considered, the amount of obtained information will be relatively limited. Secondly, the feature extraction in traditional methods mainly relies on manually designed extractors, which require professional knowledge and a complicated parameter tuning process. Meanwhile, each method is for specific applications and has limited generalization capability and robustness. Finally, computer rendering technology is constantly evolving and updating, and the realism and diversity of CG images are also constantly enhanced, which also increases the difficulty of manually designing features.

Taking into account the limitations of the hand-crafted-feature-based methods, recently researchers have used deep learning methods to solve the identification problem of CG images. Although CG image identification is modeled as a binary classification problem, it is fundamentally different from the general object classification task. One important point is that object classification tasks pay more attention to semantic information. The classic CNN network usually has limited forensic performance, so some customized modules are needed to better solve forensic problems. This is also generally recognized by researchers in the field of multimedia security [Che+15; BS18; YNY17]. Therefore, the network design of most of the above methods is more or less inspired by traditional image forensic methods, implicitly or explicitly using customized filters to extract forensically relevant signals. However, we notice that the majority of existing methods use deep learning as a technical tool, without comprehensive and in-depth analysis. For example, some interesting questions remain unanswered: How is the performance of transferring the pre-trained model of the classic image classification task in the computer vision to the CG image identification problem? What kind of information the deep neural network actually uses as the discriminative information? More importantly, although these deep-learning-based methods can usually achieve very good forensic performance, the existing methods have ignored a very important research problem, that is,

the blind detection problem (or the so-called generalization problem). When using CG images from "known" computer rendering technology and NI to train a deep model, the trained model usually has high detection accuracy on test data of same source. However, when the model is tested on CG images generated from "unknown" rendering technology, the classification accuracy sometimes drops significantly. This problem is studied for the first time in the literature in Chapter 5 and an effective solution is proposed.

1.3 Objectives and Contributions

Keeping in mind the current problems and challenges in the field of computer-generated image identification, the study of this thesis mainly focuses on the following four aspects: (1) For the generalization of colorized image detection, we use feature visualization to understand the potential underlying causes, and introduce a novel enhanced training procedure based on negative samples to improve generalization capability; (2) For the trustworthiness of CNN forensic detectors, we take colorized image detection as an example to study the impact of data preparation and CNN's first layer on forensic performance; (3) For the problem of CG image forensics, we comprehensively study the CNN-based solutions, including network design, training strategies, visualization and understanding; (4) For the improvement of CG image forensic performance, we combine the first three research work, design a new network, collect new datasets, and improve the detection accuracy and generalization capability. More precisely, the main research contents and contributions of this thesis include the following four points:

1. Colorized image detection based on negative sample insertion. In view of the limited generalization of existing hand-crafted-feature-based or CNN-based detectors in challenging blind detection scenario, this thesis proposes a CNN enhanced training method to improve the generalization capability of the detector. Here, the blind detection means that during the testing phase, the test samples are generated by "unknown" colorization methods. This is a frequently encountered situation, in which not any samples from the "unknown" colorization methods encountered during the testing phase have been used during the training phase. This blind detection performance is also called the generalization performance of forensic detectors. We first analyze the potential reasons for the limited generalization performance of neural networks by means of feature visualization, and then design an enhanced training method based on negative sample insertion. Specifically, negative samples are automatically constructed through linear interpolation of paired natural images and colorized images, and these samples have the same label as colorized images. The constructed negative samples are added into the original training dataset iteratively, then enhanced training is performed, and finally the model is chosen through a simple threshold-based method. This approach is validated on multiple

datasets and different CNNs, and the results show that the proposed enhanced training can significantly improve the generalization performance.

2. A method with improved generalization based on studies about the impact of data and network on the performance of CNN-based forensics. Recently, deep learning methods have achieved good performance in many fields, and the field of image forensics is no exception. Many researchers have introduced CNN methods to solve forensic problems, and CNN-based methods usually achieve the best forensic performance. However, high performance may conceal some potential problems or pitfalls. Therefore, this thesis carries out a study on the trustworthiness of CNN-based forensic detectors. Specifically, we attempt to study and answer several questions that are closely related to detector's trustworthiness, such as the suitability of the discriminative features automatically extracted by the CNN model and the generalization capability to "unknown" data in the testing phase. Taking colorized image detection as an example, this thesis investigates these issues and obtains some useful hints. Moreover, inspired by the idea of ensemble learning, we propose a simple and effective method to obtain the final prediction results by combining the decision results from CNN models with different settings at the network's first layer. Experimental results show that this method can effectively improve the generalization performance of colorized image detection.

3. A comprehensive study on the identification of natural images and CG images based on CNN. Motivated by the observation of the limited classification performance of traditional methods based on hand-crafted features, especially when dealing with more complex multi-source datasets, this thesis designs and implements a generic identification framework, which contains three groups of networks to process input image patch of different sizes. We first fine-tune the CNN model pre-trained on ImageNet, and then design an improved CNN network with cascaded convolutional layers for this forensic problem. Experimental results show that the two CNN-based solutions are superior to the state-of-the-art methods based on hand-crafted feature extraction and classifier training. More importantly, our method shows good classification capability on a challenging public dataset comprising images of heterogeneous origins (very close to the real-world application), and demonstrates strong robustness against several post-processing operations, including resizing and JPEG compression. Our work was one of the first deep-learning-based methods for detecting CG images. In addition, unlike the existing methods of applying CNN to image forensic problems, we use advanced visualization analysis tools, including fast Fourier transform (FFT), layer-wise relevance propagation (LRP), and "preferred inputs" (PI), to understand what our CNN has learned about the differences between NIs and CG images.

4. CG image identification based on feature diversity enhancement and adversarial examples. The forensic performance of existing CNN-based detectors can be further im-

proved, in particular the generalization capability of the trained detector on "unknown" test datasets is limited. To solve this problem, we make efforts in two aspects of CNN: network architecture design and network training. Another contribution is that for the first time in the literature we propose to study the generalization of CG image forensics. In order to study this challenging problem, we collect four high-quality CG image datasets. For the network architecture, we design a two-branch CNN. The first layer of the two branches of the network uses different initialization methods, namely, Gaussian random initialization and a set of high-pass residual filters. The purpose is to enrich the diversity of deep features. For the network training, we propose a novel model-centric method to generate more difficult negative samples (comparing with the so-called data-centric method, *i.e.*, same as the negative sample generation for colorized image detection presented earlier in this subsection, which is based on interpolation of a pair of NI and CI). Afterwards, enhanced training is performed to further improve the generalization capability of CNN which makes use of the generated negative samples. Experimental results on multiple datasets show that our proposed method can obtain better classification accuracy and generalization performance.

In summary, this thesis mainly considers four research tasks. The first two tasks focus on colorized image detection. The former solves the generalization problem, and the latter studies the impact of data and network architecture on forensic performance (especially the generalization performance). The last two tasks focus on the identification of CG images. For this forensic problem, we first propose a generic CNN-based framework and perform visualization analysis and understanding. On the basis of all our previous studies, we then improve the network architecture design and negative sample generation to achieve improved generalization capability of CG image identification.

1.4 Outline

The remainder of this thesis is organized as follows.

Chapter 2 presents colorized image detection based on negative sample insertion. With the help of visualization tools, the potential causes of the generalization degradation of the deep model are analyzed. The negative samples are constructed by linear interpolation of the paired natural and colorized images, and then the original training dataset and the automatically generated negative samples are combined for enhanced training to improve the generalization capability of the network.

Chapter 3 introduces the generalization improvement method based on the impact of data and network on CNN forensic performance. Take the CNN-based colorized image

detection as an example, some questions regarding the trustworthiness of CNN forensic detectors are analyzed, including for example the appropriateness of the discriminative information automatically extracted by CNN and the generalization performance on "unseen" data during the testing phase. A simple and effective combination strategy is proposed to improve the generalization performance of CNN.

Chapter 4 presents the CNN-based identification of natural images and CG images. Considering the design complexity of traditional hand-crafted features and the limited performance on challenging datasets, a generic framework based on CNN is proposed by carrying out comprehensive studies from network fine-tuning to the design of new networks, and its robustness against typical post-processing operations is analyzed as well. Adequate and advanced visualization tools are used to understand what the CNN has learned about the differences between NIs and CG images.

Chapter 5 describes the identification of CG images based on feature diversity enhancement and adversarial examples by considering both network architecture design and network training. We design a novel two-branch neural network, which can learn more diverse features, so as to obtain better classification performance and generalization. We also propose a new gradient-based negative sample generation method to further enhance the generalization capability with enhanced training. In the meanwhile, the four high-quality CG image datasets are collected and made publicly available to facilitate the relevant research.

Chapter 6 concludes this thesis, summarizing the contributions and proposing several perspectives about the future research work on the detection of computer-generated images.

Chapter 2

Colorized Image Detection via Negative Sample Insertion

Contents

Image colorization consists in adding artificial but plausible color information to grayscale images to obtain highly realistic color images. For the colorized image detection, the previous methods based on hand-crafted features or CNN have achieved good detection performance. However, these methods do not cope well with the challenging blind detection scenario. This chapter proposes a simple and effective method to solve this problem. Specifically, the negative samples are constructed by performing linear interpolation on the paired natural images (NIs) and colorized images (CIs) in the training dataset, and then iteratively added to the original training dataset for additional enhancement training. The whole process is completely automatic, and high generalization performance can be consistently obtained.

The main contributions of this chapter are summarized as follows:

- For the blind detection problem of colorized image identification, this chapter proposes an enhanced training framework that can improve the generalization capability of the CNN model.
- Based on the analysis of the insufficient generalization of CNN, this chapter uses linear interpolation to construct negative samples as a proxy for "unknown" test samples to assist CNN training. The process is completely automatic.

- This chapter experimentally validates the enhanced training method on multiple datasets and networks. Experimental results show that this method is suitable for a variety of network architectures and can further improve the generalization performance of the network.

The work presented in this chapter was published and orally presented at the 2019 International Symposium on Image and Signal Processing and Analysis [Qua+19b].

2.1 Network Architecture and Enhanced Training

For the colorized image detection, to our knowledge there is no existing work that considers the generalization capability for CNN-based methods. In fact, this is a highly challenging scenario because no training samples of the "unknown" colorization algorithms are available. In other words, we want the trained network to be able to successfully detect colorized images generated by new colorization methods that remain unknown during the training of CNN. This is a very realistic situation which can be commonly encountered after deploying a forensic detector in practical applications. We solve this challenging generalization problem through a simple yet effective approach, *i.e.*, inserting additional negative samples that are automatically constructed from available training samples, in order to carry out an enhanced training of CNN and thus to obtain an appropriate decision boundary for this classification problem. Besides considering the CNN model proposed in the recent work of [Zhu+18], in this chapter, we also construct a different CNN model so as to validate and show that our enhanced training can work well on different networks.

2.1.1 Architecture of Networks

In this subsection, we describe the architecture of considered networks, as shown in Figure 2.1. WISERNet [Zen+19] was originally designed to solve the problem of steganalysis of color images and achieved good results. The core idea is to replace the channel summation operation of traditional convolution with the separable channel convolution, that is, the three color channels are convolved separately to suppress the relevant image content. At the same time, to increase the signal-to-noise ratio (the ratio of steganographic noise to image content), they used a high-pass filter bank (spatial rich model, SRM [FK12]) to initialize the weights of these separable convolutions. It is believed that for natural color images, the intensity values at the same position of the three color channels show a strong and specific internal relationship. The existing automatic colorization algorithms reconstruct the three color channels of red, green, and blue from a single grayscale value

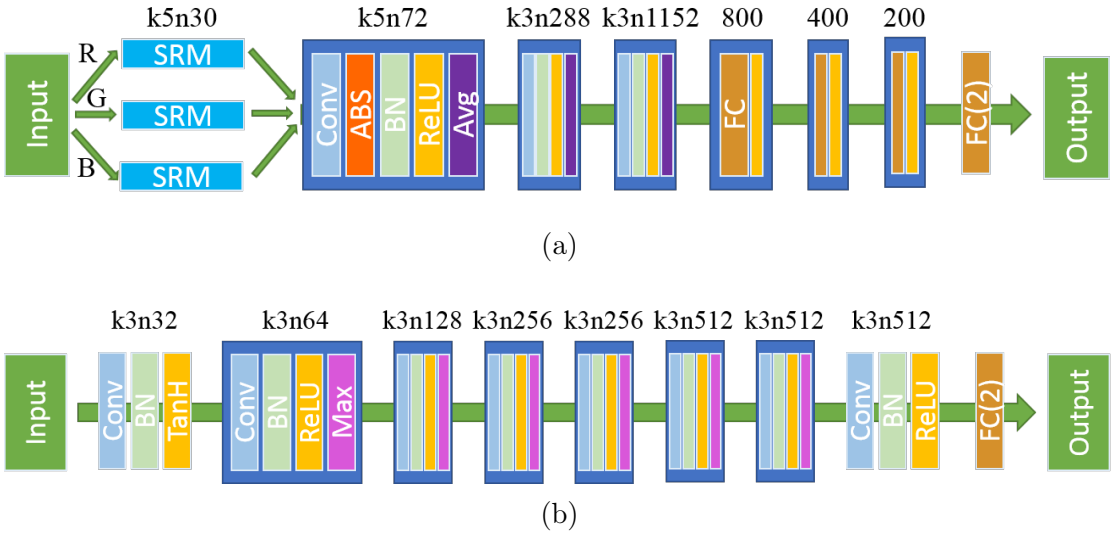


Figure 2.1: Network architecture: (a) WISERNet [Zhu+18]; (b) our designed AutoNet.

(*i.e.*, an ill-posed problem), which will inevitably introduce forensic traces into the statistical properties of the three color channels. Meanwhile, the steganalysis methods of color images are attempting to detect similar traces. Therefore, Zhuo et al. [Zhu+18] introduced WISERNet to the colorized image detection problem. Let Ck(M or A) denote a Convolution-BatchNorm-ReLU(-MaxPool or -AveragePool) layer with k filters. Fk(R) denotes a fully-connected layer with k neurons (and with ReLU). The architecture of WISERNet is SRM-C72A-C288A-C1152A-F800R-F400R-F200R-F2, where SRM refers to channel-wise convolution where the convolutional kernels are fixed as the thirty 5×5 SRM filters borrowed from [FK12]. Figure 2.2 shows part of the weights of SRM[1].

To verify the generality of the enhanced training method proposed in this chapter, that is, it is applicable to a variety of network architectures, this chapter introduces another deep network and names it AutoNet (Automatic Network). In [Zhu+18], the first layer (with so-called SRM) of WISERNet is untrainable, while the first layer of our designed network AutoNet uses common convolution, and all weights of AutoNet are trainable. The architecture of AutoNet is C32-C64M-C128M-C256M-C256M-C512M-C512M-C512-F2. All convolutional kernel sizes in AutoNet are 3×3. For layers 1-7, each convolutional layer (conv) is with the zero-padding of 1, and all max-pooling layers in AutoNet have the same kernel size of 3×3 and a stride of 2. For conv1, we use TanH as activation. In Section 2.2.2, this chapter also analyzes two other common choices in the field of image forensics, namely, no activation and rectified linear unit function (ReLU) [Hah+00; NH10]. The input and output relationship of ReLU is $\text{relu}(x) = \max(0, x)$; that of TanH is $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$.

[1]All weights of SRM can be observed at https://github.com/tansq/WISERNet.

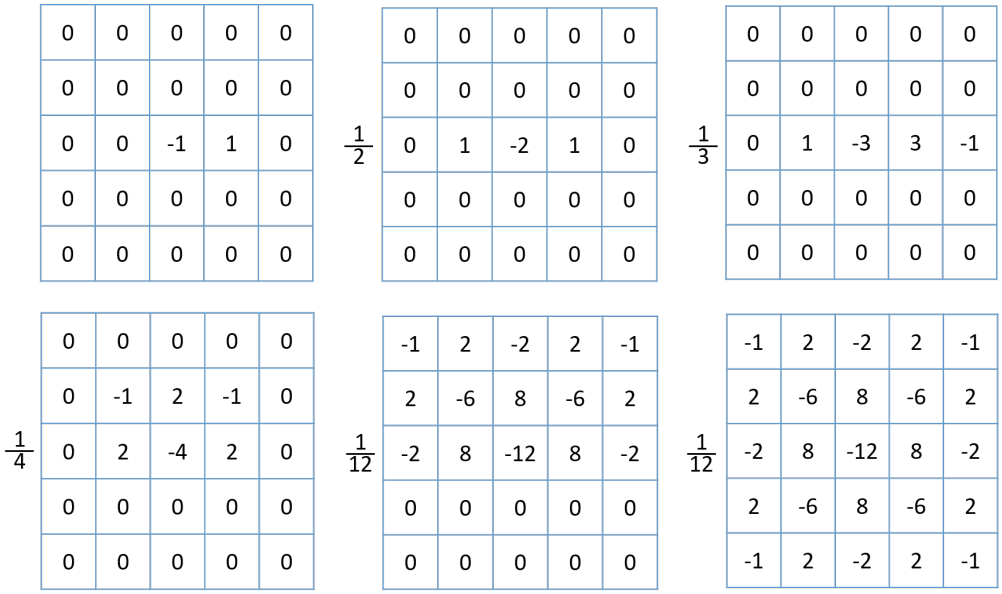


Figure 2.2: Part of the weights of SRM filters.

2.1.2 Negative Sample Insertion

According to our observation, there is a certain degree of performance decrease in the challenging blind detection scenario, not only for traditional hand-crafted-feature-based methods [Guo+18], but also for CNN-based approaches (AutoNet and WISER-Net [Zhu+18]), although the latter has better performance. In details, for a traditional or CNN-based model trained on dataset constructed by one specific colorization algorithm, the test performance on datasets constructed by other colorization algorithms is sometimes limited for colorized images. The possible reason of this performance drop is that CIs produced by a specific colorization algorithm tend to be equipped with a particular internal property, but CIs of different colorization algorithms are very likely to have different properties.

To clearly illustrate the encountered problem with an example, we train the AutoNet on the dataset constructed by colorization method Mb [ZIE16], and test on the datasets constructed by Ma [LMS16] and Mc [ISSI16], respectively. It should be noted that Ma and Mc are the "unknown" colorization algorithms, and thus the corresponding samples of Ma and Mc are not used in the training process. We use t-distributed stochastic neighbor embedding (t-SNE) [MH08] to project the high-dimensional deep features (the output of conv8 of AutoNet, and its dimension is 512) of testing data constructed by above three colorization methods onto the two-dimensional map, and detailed visualization results are shown in Figure 2.3. Comparing Figure 2.3(a), (b) and (c), we find that the distributions of NIs (red squares) are relatively stable with a rather high intra-class

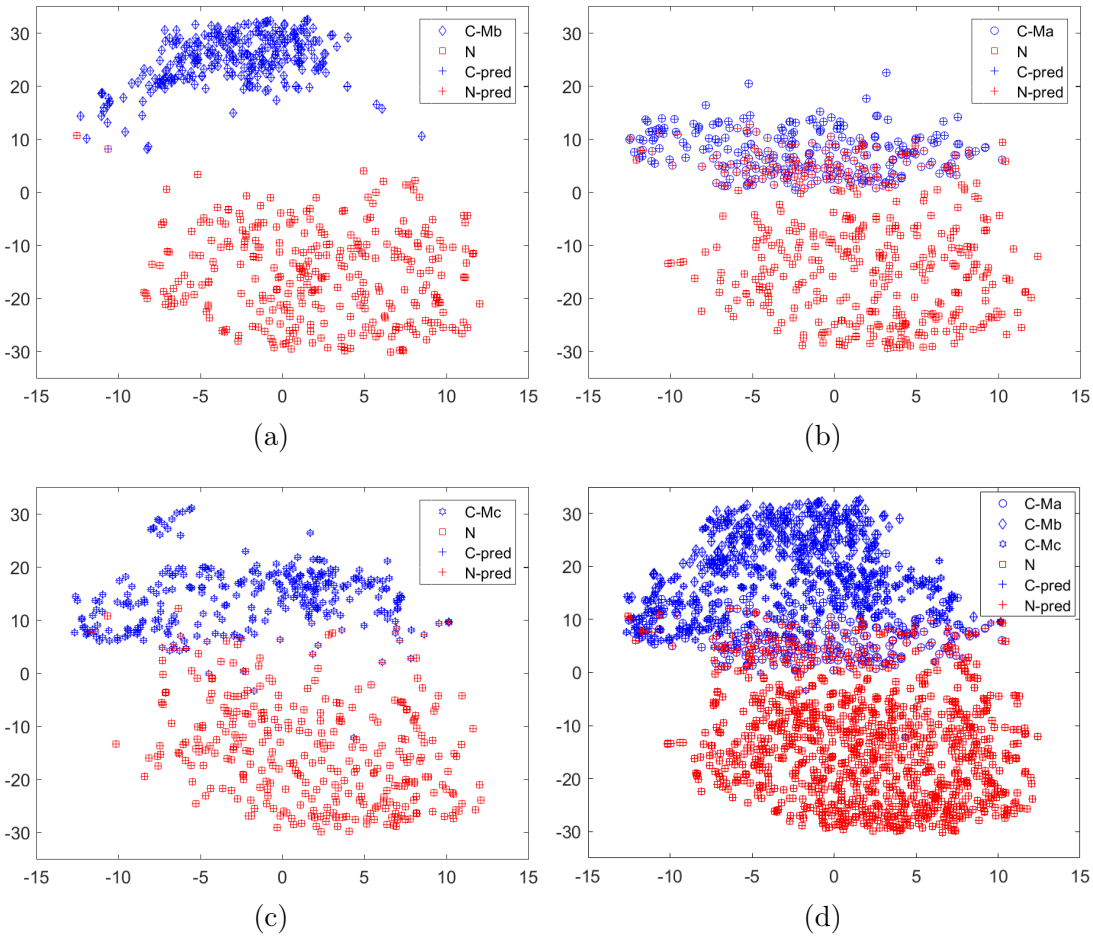


Figure 2.3: The deep feature visualization with t-SNE [MH08]. The model is trained on the original dataset where CIs are generated by Mb. "C" means colorized images and "N" means natural images. "C-X" means the colorized images produced by X colorization method, for example, "C-Ma" corresponds to CIs generated by Ma colorization algorithm. "Y-pred" means that the predicted label of CNN is Y. We randomly select 900 natural images from validation dataset splitting them into three equal subsets of 300 images, and then we construct corresponding colorized images using Mb, Ma, and Mc for every 300 images. The deep feature is the output of conv8 of AutoNet, and the dimension is 512. (d) is the combination of (a) [Mb], (b) [Ma], and (c) [Mc].

variation, which is somehow expected; in the meanwhile, CIs (blue symbols) are more tightly clustered for each colorization algorithm but their locations change a lot for different methods [please compare the CIs in (a), (b) and (c), which correspond to Mb, Ma and Mc, respectively]. This is reasonable because the different colorization methods tend to have not exactly the same internal characteristics and hence the corresponding CIs have different locations in the feature space. When the features of CIs produced

by "unknown" colorization algorithms (here Ma and Mc whose samples are not used for training) are near the decision boundary of the CNN (which is trained by using NIs and CIs produced by a "known" colorization algorithm, here Mb), and at the same time the decision boundary is relatively close to colorized images, there are high probabilities to misclassify the "unknown" CIs. For instance, many CIs in Figure 2.3(b) (blue circles with red + in the figure) are wrongly predicted as NIs.

We would like to find a simple yet effective method to solve the encountered problem. The idea is that we make use of the available training samples (and only these samples) to construct an appropriate decision boundary which can lead to better generalization performance. A feasible and intuitive solution is to add negative samples (with same labels as CIs) near the initial decision boundary of the CNN, so as to make the CNN be more "strict" about the predictions of CIs and somehow push the classification boundary towards NIs. As such, it is expected that the "unknown" CIs located close to the initial decision boundary [*e.g.*, those shown in Figure 2.3(b)] have more chance to be correctly classified with the new classification boundary which would be closer to NIs. More precisely, we construct negative sample through linear interpolation between *paired* NI and CI which share the same grayscale version and only differ in chrominance components. The corresponding formulation is shown below:

$$I_{NS} = \alpha \cdot I_N + (1 - \alpha) \cdot I_C, \tag{2.1}$$

where I_{NS} is the negative sample, I_N is the natural image, I_C is the corresponding colorized image, and $\alpha \in \{0.1, 0.2, 0.3, 0.4\}$ is the interpolation factor. This actually makes sense, as negative samples are in fact forensically negative (*i.e.*, considered as CIs), especially for our chosen weight values among $\{0.1, 0.2, 0.3, 0.4\}$ (*i.e.*, negative samples are closer to CIs than NIs). Figure 2.4 shows some images of negative samples constructed through linear interpolation. It can be observed that, when α increases, the negative samples are progressively getting closer to the natural images and it is expected that the decision boundary is further moving towards NIs after enhanced training.

As analyzed above, adding negative samples and conducting additional training will push the classification boundary towards NIs. Thus, the classification accuracy on the NIs will gradually decrease as more and more negative samples are inserted. The classification accuracy of network on validation dataset also slightly decreases because the "known" CIs are almost all correctly classified and this accuracy mainly depends on the classification accuracy on the NIs. However, in the meanwhile the CIs constructed by "unknown" colorization algorithms are expected to be classified more correctly, implying a better generalization capability. Obviously, there is a trade-off between the classification accuracy (on data similar to the training samples) and generalization performance (mainly on "unknown" CIs) for the network. Therefore, without being able to directly measure the generalization during training of network, we consider the classification ac-

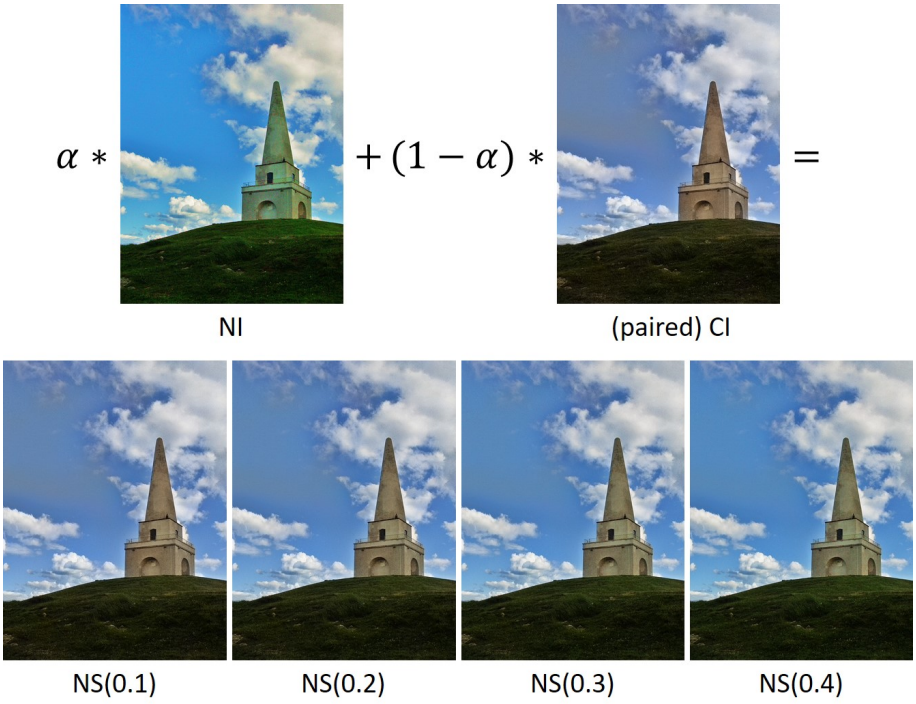


Figure 2.4: Negative sample generation via linear interpolation. The second row is negative samples, from left to right: $\alpha = 0.1, 0.2, 0.3, 0.4$.

curacy on NIs (on the so-called *natural validation dataset* $\mathcal{V}$) as a measure to select the final model in the process of additional training with negative sample insertion. In our work, we design a threshold-based model selection criterion. This threshold (θ) essentially determines the degree of final classification accuracy that can be accepted by user or current task. Generally speaking, larger θ means that the selected model has less high classification accuracy, but better generalization performance. Basically, we set $\theta = \beta \cdot error\_rate$, where β is a user defined parameter and $error\_rate$ is the classification error rate (in %, measured on natural validation dataset $\mathcal{V}$) of the CNN model trained with the original training dataset $\mathcal{D}$ before negative sample insertion. This criterion simply defines the maximum tolerable value of the relative increase of error rate on $\mathcal{V}$ induced by enhanced training. In our experiments, we set $\beta = 2$. One exception is that when $error\_rate$ is very small (less than 1%), we set $\theta = 2\%$, meaning that we can slightly relax the constraint on classification error rate to obtain relatively large improvement of generalization performance.

Algorithm 1 illustrates the training process with negative sample insertion. It is worth noting that we only use CIs of a "known" colorization method but in a better way to construct a more appropriate decision boundary. In our experiments, this insertion is an iterative process with four iterations, *i.e.*, the α is increased from 0.1 to 0.4 with

Algorithm 1 Enhanced training of CNN model with negative sample insertion

Input: $\mathcal{M}$, lr^0, S, $\mathcal{V}$, $\mathcal{D}$ and the set of corresponding natural and colorized image pairs $\mathcal{P}$ constructed from $\mathcal{D}$.

Output: final model after enhanced training.

Initialization: current learning rate $lr = lr^0$, set of negative samples $\mathcal{N} = \emptyset$, set of error rates on $\mathcal{V}$ of candidate CNN models $\mathcal{R} = \emptyset$.

1: compute $error\_rate$ of $\mathcal{M}$.
2: compute θ.
3: **for all** $\alpha \in \{0.1, 0.2, 0.3, 0.4\}$ **do**
4: construct negative samples from $\mathcal{P}$ using Eq. (2.1) and insert them into $\mathcal{N}$.
5: update training dataset: $\mathcal{D} = \mathcal{D} \cup \mathcal{N}$.
6: update the parameters of $\mathcal{M}$ for S epochs. In the second half of training process, compute error rate on $\mathcal{V}$ for each model, and insert this value at the end of $\mathcal{R}$.
7: **for all** $I_{NS} \in \mathcal{N}$ **do**
8: **if** I_{NS} is misclassified **then**
9: remove corresponding pair from $\mathcal{P}$.
10: **end if**
11: **end for**
12: set $\mathcal{N} = \emptyset$.
13: update current learning rate: $lr = lr \cdot 0.1$.
14: **end for**
15: select i-th model which satisfies $\max_i \{r_i | r_i \in \mathcal{R}, r_i < \theta\}$.

step of 0.1. Given a CNN model $\mathcal{M}$ trained by using original dataset $\mathcal{D}$, and some basic settings for CNN training, such as initial learning rate lr^0 and S epochs for each insertion, we first compute $error\_rate$ on $\mathcal{V}$ and then the threshold θ, which is used for final model selection. For each round of negative sample insertion, we construct negative samples and insert them into the dataset $\mathcal{D}$. Then, we update the parameters of model $\mathcal{M}$ using new training dataset, and compute the error rate on $\mathcal{V}$ starting from the second half of training process (*i.e.*, from $\lceil \frac{S}{2} \rceil$-th epoch for each insertion, where $\lceil . \rceil$ is the integer ceiling operator), because from that time the model becomes relatively stable. After each insertion, we test the negative samples produced by previous iteration. If a negative sample is misclassified, *i.e.*, the predicted label is NI and not consistent with its ground-truth label, then we stop using the corresponding pair to construct negative sample (*i.e.*, we remove corresponding pair from $\mathcal{P}$ as described in line 9 of Algorithm 1). In fact, this operation can slightly reduce the amount of negative samples, and does not weaken the performance of the network. After four iterations of insertion, we select the final CNN model. It is worth mentioning that when $\alpha \geqslant 0.5$, the negative samples will

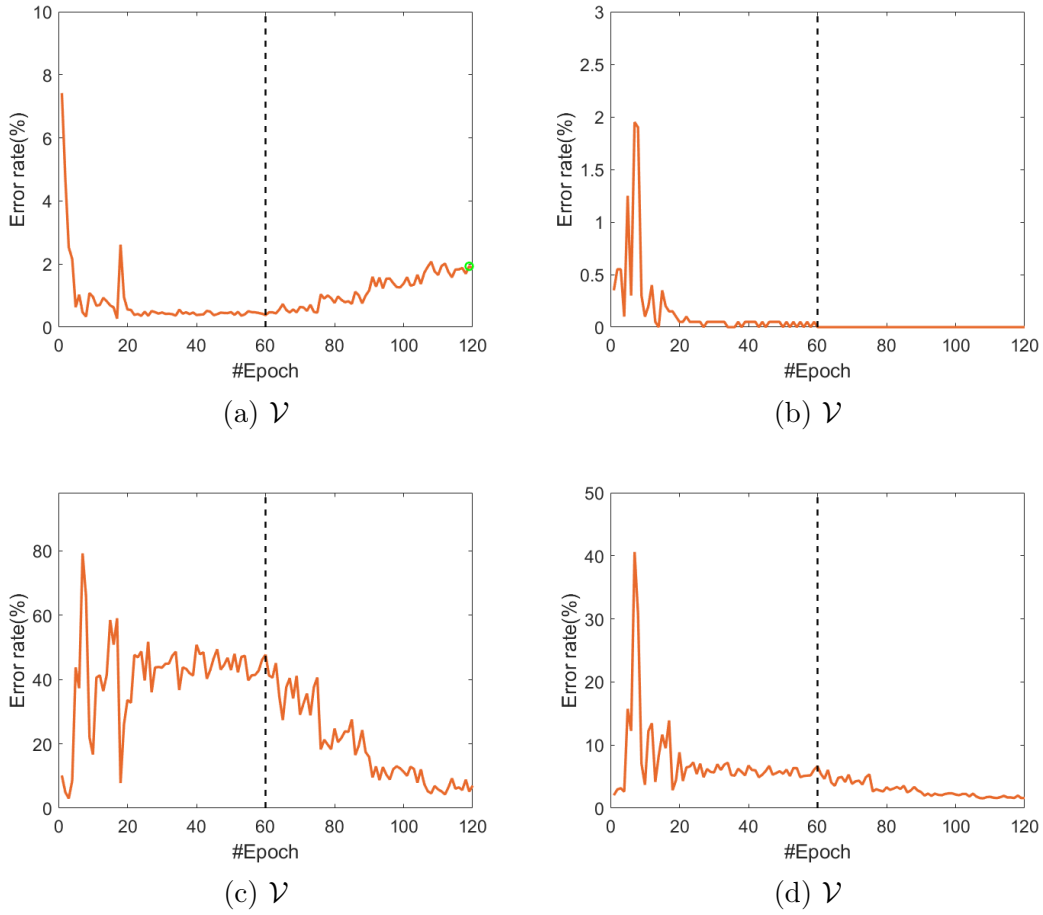


Figure 2.5: Error rate curves of a complete training of AutoNet. The network is trained on Mb [ZIE16], and tested on Ma [LMS16] and Mc [ISSI16]. The error rates (in %) on CIs produced by these three methods are shown in (b), (c), and (d), respectively. The error rate on $\mathcal{V}$ is shown in (a). Black dotted line separates the two stages of normal training (60 epochs) and enhanced training (4×15=60 epochs). The green circle in (a) stands for the final selected model.

be close to NIs, and this is likely to have more impact on the classification of NIs. Here we take a conservative and experimentally effective approach, *i.e.*, stopping the negative sample insertion process after four iterations.

The complete training process of CNN model includes two stages: (1) using the original training dataset to train the deep model from scratch until convergence (normal training); (2) iteratively adding new negative samples into the original training dataset and continuing to train the model as summarized in Algorithm 1 (enhanced training). Figure 2.5 shows the error rate curves of a complete training process of AutoNet. In

the first stage, the error rates on $\mathcal{V}$ and CIs produced by Mb obviously decrease in the first 20 epochs and the network reaches the stability after about 50 epochs, as shown in Figure 2.5(a) and (b). With the negative sample insertion, the error rate on $\mathcal{V}$ slightly increases, which can be found from the second part of Figure 2.5(a). However, the generalization performance of network has a significant improvement on CIs produced by Ma [Figure 2.5(c)] and a small improvement on Mc [Figure 2.5(d)]. More numerical and visual results (including t-SNE visualization after enhanced training) are given in the next section.

2.2 Experimental Results

2.2.1 Parameter Settings

All images in our experiments are resized to 256×256 using bicubic interpolation, and for each image, we convert its pixel values to $[-1, 1]$ (we first rescale the pixel values from the range $[0, 255]$ to the range $[0.0, 1.0]$, and then subtract these values by 0.5 and divide by 0.5). Stochastic gradient descent (SGD) with a minibatch of 20 is used to train AutoNet. Each minibatch contains 10 natural images and 10 colorized images. We randomly shuffle the order of training dataset after each epoch. For SGD optimizer, the momentum is 0.9 and the weight decay is 1e-4. The base learning rate is initialized to 1e-4. For the normal training (only using original training dataset) of AutoNet, we divide the learning rate by 10 every 20 epochs, and the training procedure stops after 60 epochs. For the normal training of WISERNet, we follow the setting described in [Zhu+18]. As shown in line 13 of Algorithm 1, for the enhanced training of AutoNet and WISERNet, we adopt the same strategy about learning rate: the learning rate is divided by 10 every 15 epochs (it is enough to guarantee the convergence after new negative sample insertion), and the training procedure stops after 60 epochs, *i.e.*, 4 iterations of negative sample insertion.

Following [Guo+18] and [Zhu+18], we also employ the *half total error rate* (HTER) to evaluate the performance of the proposed method. The HTER is defined as the average of misclassification rates (in %) of NIs and CIs. In this work, all reported results are the average of 7 runs.

2.2.2 Comparison and Analysis

Before evaluating the proposed method, we provide the details of datasets used in our experiments. Following [Guo+18] and [Zhu+18], three state-of-the-art colorization algorithms, Ma [LMS16], Mb [ZIE16], and Mc [ISSI16] are adopted for producing CIs. NIs

Table 2.1: The performance (HTER, in %, lower is better) of AutoNet with different activations: No activation, TanH, and ReLU, on ImageNet validation dataset [Den+09].

Dataset	No activation	TanH	ReLU
Ma	0.66	**0.56**	0.63
Mb	0.32	**0.19**	0.26
Mc	0.87	**0.72**	0.77

come from ImageNet dataset [Den+09]. We use 10,000 natural images from ImageNet validation dataset to construct training dataset and validation dataset, and the ratio is 4:1. The exact indexes of these images are shared by the authors of [LMS16]. Then, we remove the 899 grayscale images and 1 CMYK (cyan, magenta, yellow, and black) image from the remaining 40,000 images of ImageNet validation dataset (the total number of images in this dataset is 50,000), and obtain 39,100 natural images to construct testing dataset. Note that, the magnitude of testing dataset is far larger than the settings reported in [Guo+18] and [Zhu+18]. We employ the three colorization methods mentioned above to produce the corresponding colorized images.

Regarding the activation function of the first layer of AutoNet, this chapter analyzes three commonly used choices in the field of image forensics: no activation, TanH, and ReLU. Table 2.1 reports the classification performance of different activation functions on three datasets. We can find that AutoNet with TanH in the first layer has the best classification performance (lowest HTER), while AutoNet without activation has the highest HTER. Two possible reasons are: (1) The non-linearity of TanH and ReLU helps to increase the approximation/learning capability of the network; (2) Different from ReLU, TanH keeps the sign of features which may provide useful information for classification of NIs and CIs.

In this chapter, we propose negative sample insertion to improve the generalization performance of CNN-based detectors. As described in Section 2.1.2, this enhanced training uses natural validation dataset $\mathcal{V}$ to select the final model, and we randomly select 20,000 NIs from ImageNet test dataset [Den+09] to construct $\mathcal{V}$. Table 2.2 reports the performance of AutoNet and WISERNet before (*i.e.*, the rows of "AutoNet" and "WISERNet") and after (*i.e.*, the rows of "AutoNet-i" and "WISERNet-i") negative sample insertion. We do not present the results of hand-crafted-feature-based methods proposed in [Guo+18] because as shown in [Zhu+18] and also verified by our experiments, CNN-based method has significantly better performance in terms of both accuracy and generalization. The difference between the results of the row of "WISERNet" and those reported in [Zhu+18] is probably due to the differences in the generation of experimental data and the number of testing images (we use much more testing data). It is worth

Table 2.2: The performance (HTER, in %, lower is better) of the two CNN-based methods (AutoNet and WISERNet [Zhu+18]) on ImageNet validation dataset [Den+09]. For the sake of clarity, the generalization performance results are presented in italics.

Method	Ma			Mb			Mc		
	Ma	Mb	Mc	Ma	Mb	Mc	Ma	Mb	Mc
AutoNet	0.56	*10.57*	*10.62*	*31.65*	0.19	*6.16*	*13.93*	*1.91*	0.72
AutoNet-i	1.02	*6.94*	*5.12*	*5.13*	0.94	*1.92*	*3.33*	*1.75*	1.14
AutoNet-mixup	0.89	*12.45*	*15.35*	*20.68*	0.34	*10.04*	*8.42*	*2.25*	0.76
WISERNet	0.29	*2.21*	*10.74*	*33.30*	0.16	*7.88*	*5.80*	*0.59*	0.36
WISERNet-i	0.98	*1.22*	*2.29*	*4.74*	0.94	*2.04*	*2.46*	*1.08*	0.98

mentioning that here we focus on the generalization improvement after applying our proposed enhanced training for the two networks (*i.e.*, AutoNet and WISERNet), rather than the performance difference between them. We will study later in this manuscript the architecture comparison and the design of CNN of better generalization. From Table 2.2, we can see that the effect of negative sample insertion, *i.e.*, improving the generalization of network, is consistently stable for these two networks (except for one case, trained on Mc and tested on Mb for WISERNet, but with a very low final error rate of 1.08%). The negative sample insertion leads to slight decrease of the classification accuracy, however, the generalization performance of network usually has apparent improvement. For example, the initial generalization error of WISERNet trained on Mb and tested on Ma is 33.30%, and then reduces to 4.74% after enhanced training using negative samples, with a slight increase of classification error from 0.16% to 0.94%. This is also consistent with previous analysis (Section 2.1.2) that there is a compromise between the accuracy and the generalization performance, and our negative sample insertion method can achieve a satisfying trade-off.

In addition, we also compare our method with a recently proposed "mixup" learning principle [Zha+18] which regularizes the neural network and encourages the trained model to behave linearly in-between training examples. Although the linear interpolation is also used, there is an essential difference: "mixup" results in the linearly-transitioned decision boundary, while our method pushes the decision boundary towards NI. Based on the respective standing point, for the linear interpolation itself, [Zha+18] uses the interpolation factor in the range of $[0, 1]$ to combine pair of raw inputs and their labels, whereas our method uses that of $\{0.1, 0.2, 0.3, 0.4\}$ (forensically negative) and sets the label of new generated image as CI (the so-called negative sample). In addition, "mixup" is a form of data augmentation that implicitly affects the generalization of network, whereas our enhanced training explicitly controls the decision boundary and then improves the generalization of CNN-based detectors. In order to compare the "mixup"

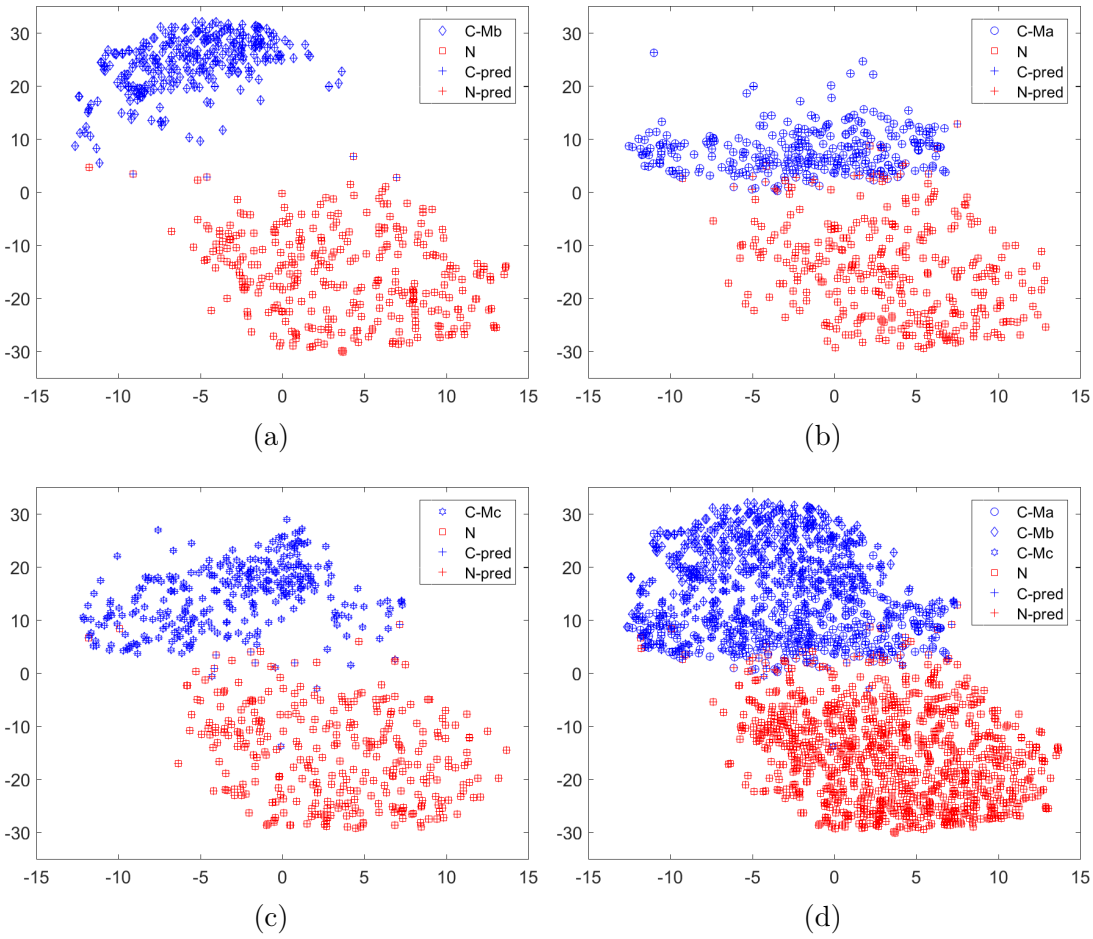


Figure 2.6: The deep feature visualization of AutoNet-i with t-SNE [MH08]. The model is obtained through enhanced training of the previously trained model (used in Figure 2.3). The meaning of symbols is same as that of Figure 2.3. It is worth noting that in t-SNE the transformation used for dimension reduction and the obtained visualization depend on the input data. Therefore, transformation and visualization in this figure are different from those of Figure 2.3.

and our method, we train the model with "mixup" where the learning rate schedule is exactly the same as the normal training of AutoNet and the results are shown in Table 2.2 (the row of "AutoNet-mixup"). We set the "mixup" hyper-parameter $\alpha = 0.4$ as recommended in [Zha+18]. Obviously, the generalization of our enhanced training based on negative sample insertion is significantly better than that of "mixup", only with a slight decrease of the classification performance (please compare the rows of "AutoNet-i" and "AutoNet-mixup").

At last, we visualize deep features of AutoNet-i using t-SNE [MH08], and the results are shown in Figure 2.6. Here, deep features are the output of conv8 of AutoNet-i, and

its dimension is 512. The corresponding visualizations of the model before negative sample insertion are shown in Figure 2.3. The testing data is also the same in Figure 2.6 and Figure 2.3. By comparing the border of correctly classified CIs, *i.e.*, blue symbols with a blue + inside, in Figure 2.3(d) and Figure 2.6(d), we can find that the latter has fewer misclassified CIs, and the classification boundary is pushed towards NIs. The CIs generated by "unknown" colorization algorithms, especially Ma [LMS16], are in consequence less misclassified, and this can be clearly observed by comparing Figure 2.3(b) with Figure 2.6(b). This confirms that our negative sample insertion scheme can push the decision boundary towards NIs to some extent and accordingly improve the generalization performance.

2.3 Summary

For the colorized image detection problem, this chapter studies the challenging blind detection scenario (*i.e.*, the generalization capability of CNN-based methods). The potential reasons for the degradation of network generalization are analyzed by feature visualization, and an enhanced training method based on negative sample insertion is proposed to improve the generalization capability of CNN-based detectors. Although the classification accuracy has decreased slightly, the generalization performance of the network has been noticeably and consistently improved. The corresponding source code can be obtained from `https://github.com/weizequan/NIvsCI`.

Chapter 3

Generalization Study for Colorized Image Detection

Contents

Recently, CNN has obtained notable success in computer vision and pattern recognition. An important reason is that CNN attempts to automatically learn hierarchical representation from available data in an "end-to-end" manner. Inspired by this success, many researchers have proposed CNN-based approaches for image forensics. For example, CNNs have been used to identify camera model [Bon+17], to expose image forgery [Zho+18], and to detect synthetic images [Qua+18; YRC19]. These CNN-based forensic methods in general work better than traditional hand-crafted-feature-based approaches. Despite this, some questions hidden behind the high performance are worth studying and answering, including the following ones: What is the CNN model using as the discriminative information, *i.e.*, is it the "essential" difference between different kinds of images? Is the CNN just overfitting on training data in some aspects that are not the primary factors for the considered forensic problem? How can CNN generalize well on "unknown" data during the testing stage? These questions are closely related to the trustworthiness and the practical applicability of CNN-based forensics.

This chapter still studies the colorized image detection, but the research focus is different from the previous Chapter 2. This chapter mainly focuses on CNN-based detectors to study issues related to trustworthiness, specifically, the impact of data and the first

layer of the network on the performance of image forensics, especially generalization capability. Zhuo et al. [Zhu+18] achieved the state-of-the-art detection performance on the experimental database shared by [Guo+18], by making use of an advanced CNN-based color image steganalyzer called WISERNet [Zen+19]. Note that, WISERNet is particularly good at detecting weak signals in images and its first layer has thirty 5×5 residual filters borrowed from the well-known hand-crafted steganalytic filters called SRM (spatial rich model) [FK12]. As shown later in this chapter, we find that data preparation and setting of CNN's first layer can have big impact on the forensic performance of WISERNet, especially the *generalization* capability.

The main contributions of this chapter are summarized as follows:

- To our knowledge, this chapter focuses on and studies, for the first time in the literature, the trustworthiness of the CNN-based image forensic detector. We try to understand the impacting factors and the potential pitfalls which are behind the high performance of CNN. Concretely, through a lot of experimental design and studies, we analyze the impact of data preparation and the setting of CNN's first layer on the performance of image forensics, especially the generalization capability.
- Inspired by the idea of ensemble learning, we propose a simple yet effective combination strategy that can further improve the generalization performance of the CNN-based detector. The effectiveness of this method is verified on multiple datasets. Though we take the colorized image detection as example in our studies, our work may be useful and inspiring for other image forensic tasks.

This work was published and orally presented at the 2019 IEEE/WIC/ACM International Conference on Web Intelligence [Qua+19a].

3.1 Data Preparation and Network

For this specific forensic problem of colorized image (CI) detection, we have some observations about data and network: CIs shared by the authors of [Guo+18] and used in [Guo+18; Zhu+18] are in a *lossless* format without compression on the artificially generated color information, and NIs (natural images) from ImageNet are in the *lossy* JPEG format; in the meanwhile, the weights of first layer of WISERNet are initialized with SRM residual filters [FK12] and untrainable. Therefore, it is natural to raise the following question: Does WISERNet, as used in [Zhu+18], rather capture the difference of processing history between NIs and CIs (*i.e.*, JPEG compressed or not), or the desired "essential" color difference? In order to answer this question, in this work, we experimentally study the impact of two important but until now ignored and underestimated

factors on CNN's forensic performance as follows: (1) we study the impact of data by constructing two datasets in which CIs are with/without JPEG compression, and (2) we study the impact of network by adopting two different strategies for the setting of the first layer of WISERNet.

3.1.1 Dataset Construction

To study the impact of different setting of datasets on the forensic performance of CNN-based detector, we construct two sets of data and the only difference is whether CIs are JPEG compressed or not. Following [Guo+18] and [Zhu+18], three state-of-the-art colorization algorithms (Ma [LMS16], Mb [ZIE16], and Mc [ISSI16]) are adopted for producing CIs. NIs come from ImageNet dataset [Den+09]. We use 10,000 natural images from ImageNet validation dataset to construct training and validation dataset (with the ratio 4:1). The exact indexes of these images can be found from [LMS16]. Then, as in the previous chapter, we remove 899 grayscale images and 1 CMYK image from the remaining 40,000 images of ImageNet validation dataset (the total number of images in this dataset is 50,000), and obtain 39,100 NIs to construct testing dataset. Note that, the magnitude of testing dataset is far larger than the settings reported in [Guo+18] and [Zhu+18]. We employ the three colorization methods mentioned above to produce the corresponding colorized images. In addition, we construct another dataset where we only replace the CIs (the original output of colorization algorithms) with a JPEG compressed version. In details, the compressed CI is generated in the following way: given an original CI, we first obtain the quantization table of the corresponding NI (*i.e.*, the NI which shares the same grayscale version) using the Matlab JPEG Toolbox [Sal03]. Then, we estimate the quality factor from the above quantization table of NI using the method proposed in [Cog18] and compress the CI with estimated quality factor. Hereafter, for ease of presentation, dataset with/without JPEG compression means that CIs in this dataset are with/without JPEG compression.

To justify the JPEG compression of CIs mentioned above we need to prove that the artificial color information in CIs before compression is indeed considered as uncompressed. To this end, we quantitatively analyze the JPEG blocking artifacts of the two datasets with the forensic measure of K_F [FQ03]. The measure K_F is formulated as:

$$K_F = \sum_k |H_I(k) - H_{II}(k)|, \tag{3.1}$$

where $H_I(k)$ and $H_{II}(k)$ are normalized histograms of pixel differences across block boundaries and within the block, respectively. Larger K_F means stronger JPEG blocking artifacts. We analyze the blocking artifacts of NIs, as well as CIs with or without JPEG compression, in the color space of YCbCr, which partitions images into luminance

Table 3.1: K_F of validation dataset in color space of YCbCr. "X-C" means the JPEG compressed version of colorized images produced by X colorization method.

Channel	NI	Ma	Mb	Mc	Ma-C	Mb-C	Mc-C
Y	0.3491	0.3227	0.3121	0.3144	0.3650	0.3667	0.3650
Cb	0.6757	0.0661	0.0596	0.0434	0.6552	0.6318	0.6741
Cr	0.7023	0.0653	0.0625	0.0489	0.6185	0.6548	0.6811

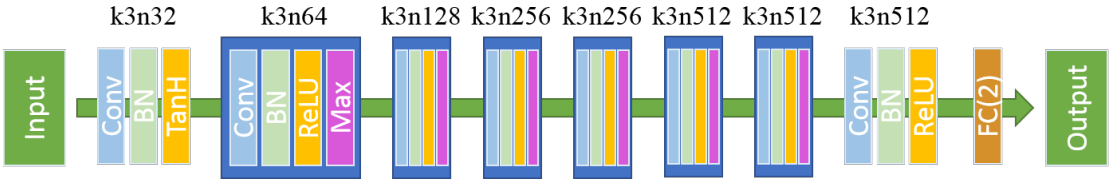


Figure 3.1: Network architecture of WISERNet

and chrominance and which is the space adopted by JPEG standard. We calculate K_F of validation dataset (2,000 images for each class) and report the average values in Table 3.1. Compared with the column of "NI", K_F of "Y" of the columns of "Ma", "Mb" and "Mc" are very similar, while that of "Cb", "Cr" have a large gap. This is an experimental proof that the color information in CIs without JPEG compression is in fact considered as forensically uncompressed. Furthermore, this gap is significantly decreased after compressing the original CIs with the same quality factor as that of the corresponding NIs (compare K_F of "Cb", "Cr" of the column of "NI" with that of the columns of "Ma-C", "Mb-C" and "Mc-C"). This implies that the difference between NIs and JPEG compressed CIs becomes very small in terms of JPEG compression trace, and that this trace seems quite obvious before compression which may impact the colorized image detection, *e.g.*, as in the experimental setting of [Guo+18; Zhu+18] where CIs are not compressed.

3.1.2 Network Settings

Figure 3.1 illustrates the network architecture of WISERNet. The first layer of WISERNet used in [Zhu+18] is a channel-wise convolutional layer where the convolutional kernels are fixed as the thirty 5×5 SRM residual filters borrowed from [FK12]. For each convolutional layer, k is the kernel size and n is the number of feature maps, and "FC(2)" stands for a fully-connected classifier layer with a 2-dimensional output of class scores. To study the sensitivity and impact of the first layer of WISERNet on the different datasets, *i.e.*, CIs with/without JPEG compression, we adopt another setting in which WISERNet's first layer is initialized in a conventional way with Gaussian random distribution and is trainable (denoted by WISERNet-Gauss). We expect that under

these two settings, the network focuses on different information in NIs and CIs to carry out the classification. In the next section, we present the experimental results related to the two datasets and the two settings of the CNN's first layer, as well as our proposed simple combination method to improve generalization.

3.2 Results and Analysis

3.2.1 Experimental Settings

All the experiments are implemented with PyTorch [Pyt]. The GPU version is GeForce® GTX 1080Ti of NVIDIA® corporation. For each image, we crop the 256×256 image patch in the upper left corner to construct the dataset.

We train the network according to the experimental setting described in [Zhu+18]. SGD with patch size of 32 is used for training all networks, the initial learning rate is set as 1e-3, the adopted scheduler of learning rate is "inv" (power: 0.75; gamma: 0.0001; weight_decay: 0.0005), and the moment is 0.9. We use the early stopping strategy to select the optimal model, *i.e.*, when the accuracy on the validation dataset does not increase after 200 epochs, the training process is terminated, and the model with the highest validation accuracy is selected as the final model. Following [Guo+18] and [Zhu+18], the *half total error rate* (HTER) is employed to evaluate the detection performance. The HTER is defined as the average of misclassification rates (in %) of NIs and CIs. In this chapter, all reported results are the average of 5 runs.

3.2.2 Impact of Data and Network

In this subsection, we first study the impact of CIs with/without JPEG compression and setting of WISERNet's first layer on the classification accuracy (*i.e.*, trained and tested on CIs of same colorization algorithm) and generalization (*i.e.*, trained and tested on CIs of different colorization algorithms). Then, we propose a simple yet effective method to improve the generalization of WISERNet. Table 3.2 reports the performance of WISERNet and WISERNet-Gauss trained on dataset without JPEG compression and tested on dataset without/with compression. On the contrary, Table 3.3 reports the performance of these two networks trained on dataset with JPEG compression and tested on dataset with/without compression. In addition, we also tested a variant of WISERNet with trainable first layer initialized with SRM filters, so-called WISERNet-T. As reported in Table 3.2 and 3.3, the performance of WISERNet and WISERNet-T is in many cases similar, therefore, we mainly analyze the results related to WISERNet

Table 3.2: The performance (HTER, in %, lower is better) of WISERNet [Zhu+18] and WISERNet-Gauss trained on dataset without JPEG compression. For each row, "X" (*e.g.*, "WISERNet") means testing on dataset without JPEG compression and "X-cro" (*e.g.*, "WISERNet-cro") means cross-testing on dataset with JPEG compression. The generalization performance results are presented in italics (same in Table 3.3).

Method	Ma			Mb			Mc		
	Ma	Mb	Mc	Ma	Mb	Mc	Ma	Mb	Mc
WISERNet	0.34	*4.49*	*3.67*	*3.27*	0.23	*0.30*	*3.15*	*0.98*	0.21
WISERNet-cro	20.56	*40.06*	*40.65*	*36.99*	22.97	*32.47*	*34.82*	*28.37*	26.31
WISERNet-T	0.31	*9.67*	*6.67*	*16.44*	0.41	*0.89*	*3.28*	*1.14*	0.33
WISERNet-T-cro	15.02	*40.28*	*42.38*	*34.49*	22.00	*35.23*	*33.48*	*27.97*	30.37
WISERNet-Gauss	0.58	*20.90*	*25.93*	*24.98*	0.43	*2.03*	*23.39*	*1.41*	0.46
WISERNet-Gauss-cro	0.89	*27.96*	*31.70*	*28.48*	10.08	*27.95*	*22.18*	*14.37*	10.53
WISERNet-Ensemble	0.60	*3.86*	*3.61*	*2.82*	0.38	*0.43*	*2.68*	*0.82*	0.38
WISERNet-Ensemble-cro	0.96	*26.44*	*29.40*	*25.79*	8.99	*23.92*	*19.68*	*13.58*	9.66

Table 3.3: HTER (in %, lower is better) of different networks trained on dataset with JPEG compression. For each row, "X" (*e.g.*, "WISERNet") means testing on dataset with JPEG compression and "X-cro" (*e.g.*, "WISERNet-cro") means cross-testing on dataset without JPEG compression.

Method	Ma			Mb			Mc		
	Ma	Mb	Mc	Ma	Mb	Mc	Ma	Mb	Mc
WISERNet	0.78	*18.90*	*24.28*	*9.35*	0.93	*2.98*	*4.78*	*2.79*	0.89
WISERNet-cro	0.69	*13.78*	*17.40*	*10.31*	0.67	*1.08*	*4.40*	*1.29*	0.66
WISERNet-T	0.80	*29.37*	*26.38*	*7.39*	0.87	*3.37*	*3.46*	*3.65*	1.08
WISERNet-T-cro	0.72	*25.42*	*23.40*	*8.62*	0.73	*2.01*	*3.09*	*1.94*	0.89
WISERNet-Gauss	0.76	*25.52*	*27.97*	*9.03*	0.89	*5.26*	*4.05*	*3.53*	0.96
WISERNet-Gauss-cro	0.72	*22.97*	*29.27*	*12.53*	0.74	*3.86*	*5.49*	*2.44*	0.84
WISERNet-Ensemble	0.82	*16.00*	*20.43*	*6.08*	0.93	*2.10*	*2.44*	*2.14*	1.00
WISERNet-Ensemble-cro	0.80	*11.72*	*15.81*	*6.72*	0.85	*1.10*	*2.30*	*1.40*	0.95

in the following.

For the analysis of the impact of JPEG compression and CNN's first layer on the forensic performance, we first analyze the different performance of WISERNet on datasets with/without JPEG compression, then we analyze the influence of the different settings of the first layer of WISERNet on network performance.

Compared with the row of "WISERNet" in Table 3.2, we can find that the classification and generalization error rate of the row of "WISERNet" in Table 3.3 both

increases. In other words, when only replacing CIs with corresponding compressed version, the forensic performance (especially generalization) obviously drops. Meanwhile, the detection performance significantly decreases when we train WISERNet on dataset without JPEG compression and test it on dataset with compression (compare the rows of "WISERNet" and "WISERNet-cro" in Table 3.2), whereas this phenomenon does not exist in the case of training on dataset with JPEG compression and testing on dataset without compression (compare the rows of "WISERNet" and "WISERNet-cro" in Table 3.3). These results indicate that WISERNet takes the trace of JPEG compression as the important discriminative feature and thus has good generalization when trained and tested both on dataset without JPEG compression of CIs (the row of "WISERNet" in Table 3.2).

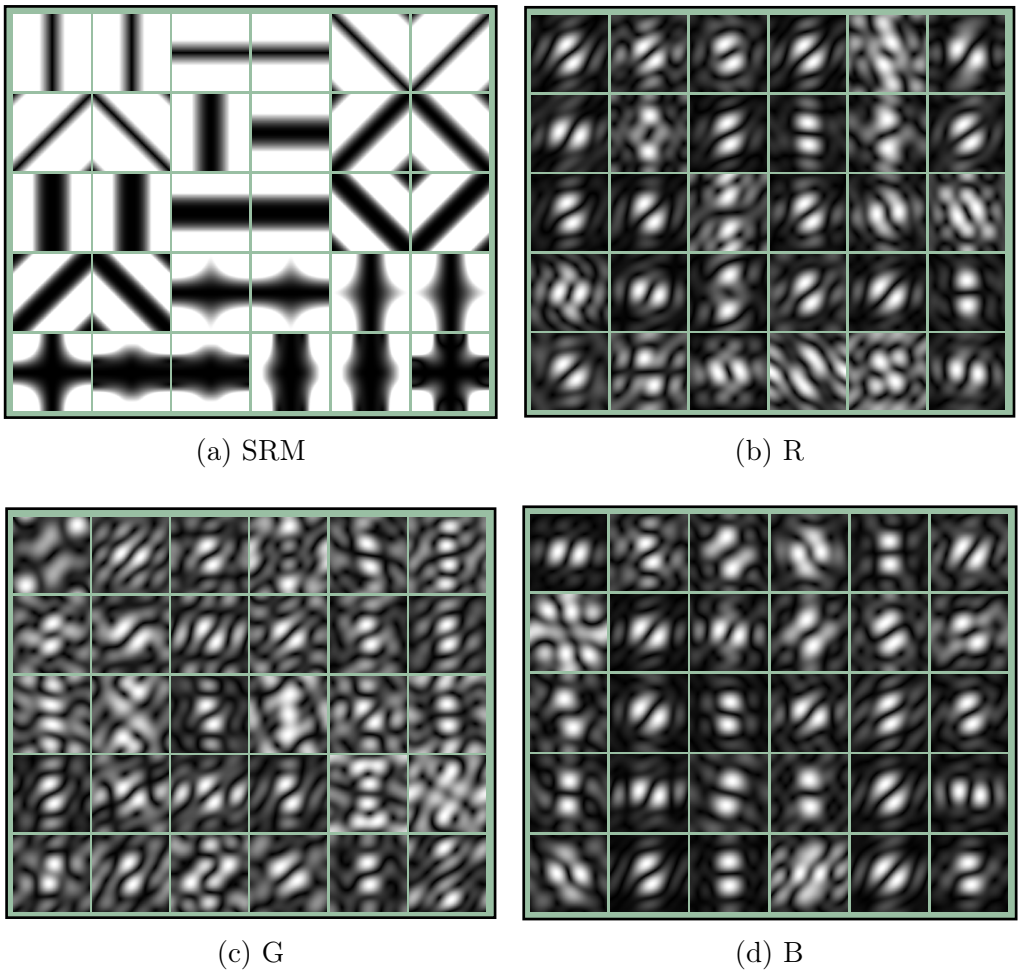
In addition, in Table 3.2, when the first layer of WISERNet is initialized with Gaussian random distribution and trainable (the so-called WISERNet-Gauss), the generalization is not as good as the original WISERNet (compare the rows of "WISERNet" and "WISERNet-Gauss"). On the contrary, the rows of "WISERNet" and "WISERNet-Gauss" in Table 3.3 are relatively close, where the networks are trained and tested on datasets with JPEG compression of CIs. This further implies that the SRM filters can strongly capture the trace of JPEG compression.

To summarize, when the training datasets have a pitfall, *i.e.*, the CIs are without JPEG compression, the WISERNet can achieve very good detection performance, especially generalization, because this model uses SRM filters in the beginning of network which coincidentally and *mistakenly* detects the trace of JPEG compression. This is however not desirable and leads to the dramatic performance drop in the row of "WISERNet-cro" in Table 3.2. In the meanwhile, when the training database is carefully prepared as in Table 3.3, the original WISERNet is indeed a good choice for this forensic task, providing better overall performance than WISERNet-Gauss and WISERNet-T.

3.2.3 Generalization Performance Improvement

During the above research, we notice that the detection performance differs for WISERNet and WISERNet-Gauss, *e.g.*, the rows of "WISERNet" and "WISERNet-Gauss" in Table 3.3. The only difference between these two networks is in the first layer. Intuitively, this difference may, to some extent, guide the two networks to extract different discriminative features. We qualitatively analyze this difference, and visualize the FFT (fast Fourier transform) of the first-layer kernels of these two networks after training, and the corresponding results are shown in Figure 3.2.

As shown in Figure 3.2(a), many kernels in the first layer of WISERNet have an ap-



(a) SRM (b) R

(c) G (d) B

Figure 3.2: Visualization of FFT of the first-layer filters of WISERNet [(a)] and WISERNet-Gauss [(b), (c), and (d)]. From left to right: SRM, filters in R, G, and B channel, respectively. Note that, filters for R, G, and B in WISERNet are all SRM.

parent high-pass response, whereas the kernels of WISERNet-Gauss [Figure. 3.2(b), (c) and (d)] mainly capture the band-pass frequency information. Based on these observations, we introduce a simple yet effective method to further improve the generalization, by somehow borrowing idea from ensemble learning. Specifically, we combine the predictions of these two networks to obtain the final prediction according to *a simple criterion*: the final prediction is CI when the prediction of either of two networks is CI, otherwise the image is NI. The rationale behind this criterion is that we trust the (different) discriminative features of both networks which are used to determine whether an image is CI. The corresponding results are reported in rows of "WISERNet-Ensemble" and "WISERNet-Ensemble-cro" in Table 3.2 and 3.3. Obviously, the generalization can be improved by this method while decreasing very slightly the classification accuracy (compare the rows

of "WISERNet", "WISERNet-Gauss", and "WISERNet-Ensemble" in Table 3.2 and 3.3). Despite of its simplicity, to the best of our knowledge, this ensemble strategy with different initialization methods (SRM and Gaussian) is for the first time used for improving the generalization of CNN-based image forensic detector. The success is probably due to the high diversity of the two initializations, more diverse than the conventional way of using only one initialization method (*e.g.*, Gaussian) with different random sampling.

3.2.4 Discussion

Unlike traditional hand-crafted-feature-based forensic methods, recent CNN-based approaches are relatively difficult to understand concerning what is the discriminative information used by CNN, and sometimes this information can be surprising and misleading. Taken the above CNN-based colorized image detection as an example, when within the dataset there is an apparent difference in JPEG compression, the CNN with SRM filters captures to some extent this trace and takes it as part of the discriminative information. Consequently, the high performance of CNN model benefits from and covers up the potential pitfall existing in the dataset. As far as we know, there is no existing work that considers and studies this kind of phenomenon for CNN-based image forensics. From this case study, we get some useful hints: 1) reducing as much as possible the impact of image generation and processing history (this information is not relevant to the task at hand), so we need to carefully prepare the data; 2) carefully using some existing filters (*e.g.*, SRM filters) in the beginning of CNN because these filters have strong capacity of capturing image processing history and thus are risky to be used if the dataset has not been properly prepared.

3.3 Summary

This chapter considered the CNN-based colorized image detection as an example and studied the impact of image generation pipeline and CNN's first layer on the classification accuracy and generalization. From this case study, we learned some lessons related to the trustworthiness of CNN-based forensics, which until now have been ignored among the community but would be helpful for researchers in the field to avoid biased data preparation and network design. We think that our new study in this direction can be useful for other forensic tasks, *e.g.*, detection of computer graphics images and GAN-generated fake images where similar problems and pitfalls may exist. It is interesting to continue our work on improving the generalization of deep-learning-based detectors, either based on an ensemble of classifiers or other approaches.

Chapter 4

Identification of Natural and CG Images based on CNN

Contents

Chapters 2 and 3 focus on the forensic problem of colorized image detection, especially the generalization capability. Besides the fake colorized images, computer graphics images are also a common and important type of computer-generated images. Starting from this chapter, we will study the identification of computer graphics images, mainly based on CNN.

At the time when we carried out the studies presented in this chapter, prevalent methods for distinguishing between NIs (natural images) and CG (computer graphics) images with various scenes and contents followed the classical pipeline of machine learning, which consisted of two separate phases: (1) designing sophisticated, discriminative and hand-crafted features (almost always multidimensional features); (2) training classifiers (*e.g.*, SVM, ensemble classifier). This pipeline usually performs well on relatively simple datasets, such as those in which NIs are acquired by only one or two digital cameras. However, such methods often exhibit limited performance (to be shown later in this

chapter) on complex datasets comprising images of heterogeneous origins. An example is the challenging setting of the Columbia dataset [Ng+04], in which we want to differentiate between NIs collected from Google Image Search (Google) and photorealistic computer graphics (PRCG) images downloaded from various websites of CG image collections. In general and as argued by other image forensic researchers [Che+15; BS18], discriminative hand-crafted features are tedious to design, and the designed features are not necessarily the most adequate for a given forensic problem, especially for complex and challenging datasets. To this end, we propose a new data-driven, CNN-based framework to distinguish between NIs and CG images. The proposed framework is different from the traditional pipeline of almost all existing methods (at the time when we conducted this work) with two separate steps of feature extraction and classifier training. The proposed framework is "end-to-end" and does not require designing features by hand.

Our study is one of the first attempts in the literature on deep-learning-based detection of computer graphics images. We pay special attention to the completeness of our study, including fine-tuning popular network, new network design, comprehensive experimental evaluation, visualization and understanding, etc. Our contributions are summarized as follows:

- We introduce a generic framework that uses CNN to identify NIs and CG images. This framework can be easily adjusted to handle different sizes of input image patches.

- We fine tune a pre-trained CNN and then design and implement a new and improved CNN. Both CNN-based solutions outperform state-of-the-art methods that combine hand-crafted feature extraction and classifier training.

- Our method exhibits good forensic performance in the challenging dataset of Google versus PRCG comprising images of heterogeneous origins and is thus close to the real-world application. Our method also demonstrates strong robustness against several post-processing operations, including resizing and JPEG compression.

- Unlike previous attempts to use CNNs for other image forensic problems, we attempt to understand what our CNN has learned about the differences between NIs and CG images by using advanced visualization tools, which provide interesting observations and insights for future studies.

The work presented in this chapter was published by the international journal "IEEE Transactions on Information Forensics and Security" [Qua+18].

4.1 Proposed CG Image Identification Framework

For this CG image forensic problem, the traditional two-stage classification models have limited performance on complex data with heterogeneous origins, and a reason is that these hand-crafted features are tedious to design and not necessarily the most adequate ones. A generic "end-to-end" framework, such as CNN, may be a better option. Given a testing image, a well-trained CNN can directly and accurately predict its label. To this end, we introduce a suitable CNN model for our framework. We consider three different methodologies in our approach: (1) following the existing network architecture and training it from scratch, (2) fine-tuning an "off-the-shelf" network that has been pre-trained on another dataset and/or for another task, and (3) designing a new network and training it from scratch. Before providing details on these methodologies, we present our general strategy adopted when using CNN for classifying NIs and CG images.

4.1.1 Local-to-Global Strategy

In view of computational cost, diversity of image size, and specific requirement of image forensics, we adopt the *local-to-global* strategy (LGS) of training on small patches and classifying full-sized images using the simple majority voting rule. This strategy is partly based on the concept of data augmentation, which is a commonly used technology to expand training data, especially for deep learning [KSH12; SZ14]. Krizhevsky et al. [KSH12] randomly altered the intensities of the RGB channels of each training image using principal component analysis. The motivation behind this scheme is that object identification in digital images should be invariant to changes in the pixel intensity and color of the illumination. Simonyan et al. [SZ14] resized each training image, with the length of its shorter edge as an integer randomly sampled from the range of $[256, 512]$ for scale augmentation.

For our classification problem, on the one hand, local decisions, *i.e.*, high accuracies on small image patches, are important and generally desirable in many image forensic applications. On the other hand, a small patch cropped from a CG image is still CG, and this is also true for NI. Therefore, we apply *patch augmentation*, that is, we crop a certain number of image patches of a fixed size from each training image to augment the training dataset and try to obtain an accuracy as high as possible on patches.

This strategy is flexible for local and global forensic decisions. The direct result of such a strategy is high classification accuracy on patches, and the strategy can thus be used for the case of local decisions without any modification. For global decisions, merely conducting majority voting of multiple local decisions can lead to good performance,

which is a natural result of the high accuracy on patches. In practice, we randomly crop a fixed number of patches from each training image using Maximal Poisson-disk Sampling (MPS) [Qua+16] to construct the training set. Unlike random sampling, cropping with MPS can completely cover the entire image and thus retains the original information as much as possible. In the testing phase, we crop a certain number of patches from each testing image and take the label (0: CG image and 1: NI) of patches with a higher number as the prediction result of this image. As shown later, this strategy can also enhance the performance of existing approaches that are based on manually designed features.

4.1.2 Fine-Tuning

Fine-tuning, a technique based on the concept of transfer learning, is pervasive in the field of deep learning. Yosinski et al. [Yos+14] analyzed the transferability of neurons in each layer of a deep CNN. For similar datasets, they found that initializing the weights of almost any number of layers from a well-trained network on an original dataset can improve the generalization performance after fine-tuning to the new dataset. Such transferability generally declines as the dissimilarity between the source and target task/data increases. However, fine-tuning pre-trained CNN models from computer vision tasks has in general been omitted by the multimedia security community, at least at the time of this work. We are curious about and want to verify the transferability of such pre-trained models when applied to image forensic problems, although our available data and classification task are somewhat different from those of the pre-trained CNN model.

A well-known reference network for visual recognition is CaffeNet [Jia+14], which is trained on 1.3 million images with 1,000 categories. CaffeNet has eight layers (or group of layers): two convolutional groups, each of which includes one convolutional layer, one max-pooling layer and one local response normalization layer; three cascaded convolutional layers, followed by a max-pooling layer; and three fully-connected (FC) layers. In CaffeNet, each convolutional layer consists of linear multidimensional convolutional kernels and rectified linear unit (ReLU) activation [Hah+00; NH10]. We successively fine-tune the first N layers, where $N = 1, 2, \cdots, 7$. We always need to change the number of neurons in the last output layer from 1,000 (for the 1,000 classes of ImageNet [Den+09]) to 2 (binary classification of NIs and CG images). The detailed results are reported in Section 4.2.3, where we show that fine-tuning CaffeNet leads to satisfactory results that are better than those of state-of-the-art methods.

Next, we decide to design and implement our own CNN that can cope even better with the classification of NIs and CG images. We present its architecture and energy function in the next two subsections. The design of the new CNN is motivated by the

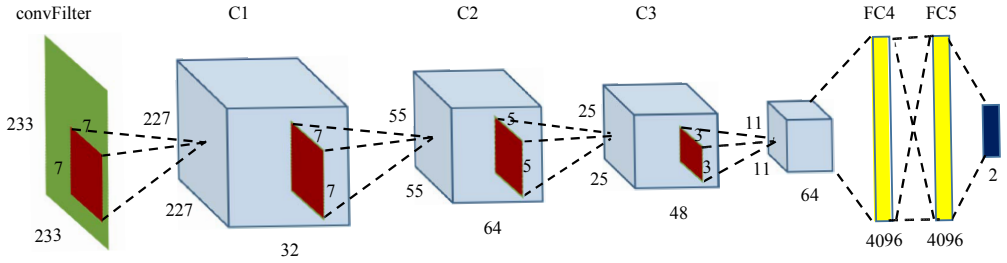


Figure 4.1: Architecture of NcgNet. The network input is a 233×233 RGB image, which is represented by a green square for simplicity. A red square stands for a convolutional kernel, and the numbers close to it denote the kernel size. For example, the first red square from the left is a 7×7 convolutional kernel. The feature maps are represented by shaded cuboids. No padding exists in NcgNet.

following observations and intuitions. First, our task and the corresponding dataset are more or less dissimilar to those of CaffeNet (in particular, no CG image is used during the training of CaffeNet); therefore, transferability might not be optimal. Second, CaffeNet is a relatively complex CNN designed for advanced and complicated computer vision tasks, but our task on hand is a less complicated two-class classification problem. Thus, a less deep and less complex network would suffice to solve our problem. Third, the fixed architecture of CaffeNet prevents us from easily adapting the network to accommodate different sizes of input patches.

4.1.3 Proposed Network - Architecture

Figure 4.1 shows the architecture of our network, and we denote it as NcgNet (natural and computer graphics network). The input of our network is an image patch, and the output is a binary label. One image patch is abstracted step by step through nonlinear mapping (*i.e.*, linear convolution and nonlinear activation) and down-sampling. A powerful high-level reasoning is then applied. The informative and highly abstracted vector is converted into the probability vector of the label. As illustrated by Figure 4.1, the entire network is made up of the so-called *convFilter* layer, three convolutional groups, two FC layers and a softmax layer. Before explaining each layer, we mention one detail about the relationship between the patch and input sizes of CNN. We actually follow the common way in the field of computer vision [KSH12; He+16]. The patch size (240×240) is slightly larger than the input size of the network (233×233) shown in Figure 4.1, which can increase the space of training samples and is thus useful in suppressing potential over-fitting. During each iteration of network training, every 233×233 training sample is randomly cropped from a 240×240 patch. In the testing stage, the network extracts five patches

of 233×233 pixels (the center and four corner patches) from a testing sample, flips these patches in the left-right direction (*i.e.*, horizontal reflection), and finally averages the predictions of total 10 patches as the final result.

The convFilter layer consists of a few convolutional kernels (32 in NcgNet). In multimedia security, such as steganalysis, a common operation is applying a group of filters on an input image/patch prior to the execution of the main algorithm [PBF10; FK12]. The convFilter layer cannot be simply regarded as "pre-processing" like fixed and manually designed filters in previous steganalytic methods because our layer is trainable without any constraint. In addition, these kernels are not explicitly required to have high-pass properties, such as in several previous methods that use CNN for steganalysis and forensics [Qia+15; Che+15; BS18]. Technically, the convFilter layer maps an RGB image to several feature images filled in with real value elements and co-adapts to the successive convolutional group. In Section 4.2.3, we analyze the classification accuracy of our network with the convFilter layer and compare with several different configurations related to this layer.

In our network, a convolutional group includes convolutional (Conv), batch normalization (BN), ReLU activation and max-pooing layers. The Conv layer conducts multidimensional linear operations and produces multiple feature maps. BN [IS15] explicitly forces the output of Conv to take on a unit Gaussian distribution. This layer makes network training highly robust to poor initialization. The ReLU activation layer introduces nonlinearity into the network and thus enhances the mapping capacity of the model. Its form is $f(x) = \max(0, x)$. Max-pooling is a down-sampling operation, where the maximum value within a local window is taken as the output. On the one hand, this operation reduces the number of parameters to learn by reducing the spatial size of representation and thus decreases computational cost. On the other hand, this operation provides basic translation invariance to internal representation. In our network, all max-pooling layers have the same kernel size of 3×3 and a stride of 2.

The two FC layers constitute an FC two-layer neural network, where every single neuron connects to all neurons in the previous layer. This part of network conducts high-level reasoning. Many parameters of the network are located here. A simple and effective regularization technique, *i.e.*, Dropout [Sri+14], is applied to each FC layer to prevent potential over-fitting. In the training stage, each unit in the FC layers is kept active with a probability (default value is 0.5), with the interpretation of sampling the neural network and updating the weights of such sub-networks on the basis of input data. No Dropout is applied in the testing stage.

The softmax layer maps the high-level feature vector (output of FC layers) to the probability vector of class labels. Therefore, the dimension of its output is equal to the

number of classes, and the sum of its output is 1.

To accommodate multiple input sizes, we design three groups of network. We do not change the depth of the network and the number of kernels in each layer during the adjustment of network architecture to maintain the structural stability of our CNN. For a small input size, a new network can be rapidly built by simply reducing the kernel size and removing the striding of the first few layers. This minor adjustment also ensures that experimentally the input flow can propagate to the last FC layers with a sufficient amount of useful information.

4.1.4 Proposed Network - Loss Function with Regularization

CNN models are usually trained by minimizing a well-designed loss function with the aid of back propagation. A loss function is often composed of a data loss term and a regularization term. The data loss term evaluates the compatibility between a prediction (*e.g.*, the class scores in a classification problem) and the ground-truth label, and the regularization term on model weights is designed to prevent the over-fitting of trained models. In our method, we use multinomial (binomial in our case) logistic loss (also known as cross-entropy loss) with softmax, that is

$$J(\theta)^{(data)} = -\frac{1}{N}[\sum_{i=1}^{N}\sum_{j=1}^{K}\mathbb{1}\{y^i = j\}\log\frac{e^{a_j^i}}{\sum_{j=1}^{K}e^{a_j^i}}], \tag{4.1}$$

where N is the number of training samples, K is the number of categories, $\mathbb{1}\{\cdot\}$ is the indicator function so that $\mathbb{1}\{True\} = 1$ and $\mathbb{1}\{False\} = 0$, $\frac{e^{a_j^i}}{\sum_{j=1}^{K}e^{a_j^i}}$ is the softmax function (normalized exponential function) that converts the network output into the probability of the class label, $a^i = \phi(x^i, \theta)$ is the 2D output vector of the network parameterized by θ, and $K = 2$ in our case.

We select L_1 regularization to be added to the loss function to reduce the complexity of model and prevent over-fitting. The total loss is

$$J(\theta) = -\frac{1}{N}[\sum_{i=1}^{N}\sum_{j=1}^{K}\mathbb{1}\{y^i = j\}\log\frac{e^{a_j^i}}{\sum_{j=1}^{K}e^{a_j^i}}] + \lambda|\theta|, \tag{4.2}$$

where regularization weight λ balances the data loss and regularization terms.

We also attempt L_2 regularization and find that L_1 regularization yields better results. A possible explanation is that in this work, we consider a binary classification problem, which is not highly complex. From a human cognition point of view, solving such a problem may not require a large amount of brain activity and area to learn and

understand. Analogically, our problem may just require a relatively simple model, that is, a model with sparse parameters reflected by the selected L_1 regularization.

4.2 Experimental Results and Analysis

4.2.1 Dataset

Our experiments are mainly conducted on the Columbia Photographic Images and PRCG Dataset [Ng+04]. The experiments consider three sets of images from the Columbia dataset: (1) 800 PRCGs from 40 3D graphic websites (PRCG), (2) 800 NIs from some personal collections (Personal), and (3) 800 photographic images from Google Image Search (Google). We remove five images with incorrect labels from the Google set after discussing via email with the first author of the dataset and obtaining his approval. The final number of images in the Google set is 795. Previous studies have considered several common dataset settings, such as Personal+Google versus PRCG [Ng+05; CSX07], Personal versus PRCG [GC08], and authors' own datasets (mostly not publicly available) [LF05; Kha+08; LYS13; Pen+17], which were sometimes combined with the Columbia dataset. The NIs in the authors' own datasets are often acquired by a small number of digital cameras; this is similar to the configuration of Columbia's Personal set and thus appears to be less challenging. To the best of our knowledge, no previous method has been tested under the challenging setting of Google versus PRCG. This setting is difficult because NIs in Google and CG images in PRCG have heterogeneous origins [GC08]. We focus on this most challenging setting, *i.e.*, Google versus PRCG, which comprises images that we typically encounter in a real-world forensic scenario. We also test our method on two other settings: Personal versus PRCG and Personal+Google versus PRCG.

4.2.2 Experimental Settings

All experiments in this chapter use the deep learning framework Caffe [Jia13]. Before conducting all the experiments, we resize all images using bicubic interpolation so that the shorter edge of each resized image has 512 pixels. This operation can reduce the impact of scale and thus ensure the consistency of all image patches. For all three settings, namely, Google versus PRCG, Personal versus PRCG and Personal+Google versus PRCG, we use the ratio of 3:1 to randomly split each dataset into training and testing sets. To follow the local-to-global strategy and generate sufficient training data for our CNN model, we randomly crop 200 patches from each training image using MPS [Qua+16]. Similarly, the testing set is obtained by cropping 30 patches from each testing image. Every patch

Table 4.1: Impact of number of extracted patches (in row) on the network's performance for testing patches of 240 × 240 pixels on the Google vs. PRCG dataset.

Num.	Accuracy (%)	Standard deviation (%)
100	84.67	0.6203
200	85.15	0.2861
300	85.15	0.2545

is pre-processed by subtracting the per-pixel mean of all training patches. Stochastic gradient descent with a minibatch of 128 patches is used to train CNN models. The base learning rate is initialized to 1e-3 and is divided by 10 every 30K iterations. The training procedure stops after 90K iterations. The default value of regularization weight λ is 1e-4, except for patch sizes of 60×60 and 30×30, whose regularization weights are 5e-5 and 1e-5, respectively. As the patch size decreases, the number of parameters in the corresponding CNN model decreases (additional details of networks for different patch sizes are given in Section 4.2.4). Thus, using a small λ value for regularization is reasonable, and experimentally, this leads to a slightly improved performance.

As described above, we extract 200 patches from each training image. For the patch size of 240×240, we have relatively high overlapping between patches. This does not weaken the performance of our network. Table 4.1 shows the median and standard deviation of the results of 7 runs for different amounts of cropped patches (*i.e.*, 100, 200 and 300). Compared with 100 patches, the classification accuracy of 200 patches is increased by 0.48%, and the standard deviation is reduced by 0.3342%. This result means that in this case, doubling the training data can improve the performance and stability of our network. However, when we increase the number of cropped patches from 200 to 300, the network's performance remains nearly the same, but the computational cost increases.

We compare our proposed method with four state-of-the-art methods at the time of this study which are based on hand-crafted features, namely, Spam [PBF10], Geo [Ng+05], Mfra [Pen+17], and Vlie [ZWN12] (the fourth method is mainly for robustness evaluation against JPEG post-processing). These four methods follow the conventional two-stage pipeline of machine learning and use SVM as the classifier. Considering the long training time and high memory footprint of SVM, we randomly crop 10 patches from each training image to construct the corresponding training sets for the first three conventional methods [PBF10; Ng+05; Pen+17] and 15 patches for the Vlie method to compensate for the 100 images of each category that are used to compute the visual vocabulary, similar to the original paper [ZWN12]. The number of samples in these training sets ensures reasonable training time and memory consumption and is

Table 4.2: Impact of different numbers of extracted patches on the classification accuracy of testing patches of 240×240 pixels and full-sized testing images with voting for the best two state-of-the-art methods, namely, Geo [Ng+05] and Spam [PBF10]. Experiments are conducted on the Google vs. PRCG dataset.

Num.	Geo		Spam	
	patch	voting	patch	voting
10	80.65	87.91	76.13	84.63
20	80.76	88.16	76.33	84.89

also sufficient for obtaining stable and near-optimal results for the four methods. As shown in Table 4.2, doubling the training samples exerts a minor impact on classification performance, but the memory footprint considerably increases. For Spam with a high-dimensional feature vector, the memory consumption becomes prohibitive even on a computer equipped with 32GB of RAM. All these methods are evaluated on the same testing set as our proposed method. For SVM training, we use the popular and efficient LS-SVM implementation [Suy+02].

4.2.3 Fine-Tuning CaffeNet and Analysis of convFilter Layer

To solve the CG image forensic problem, we first explored the fine-tuning of CaffeNet, and corresponding experimental results are reported in Table 4.3, in which the accuracy is computed on all testing image patches of 240×240 pixels in the setting of Google versus PRCG. In addition to this patch-wise accuracy, we generally observe the same trend for other metrics, such as the accuracy after voting on full-sized images. We also train CaffeNet on the Google versus PRCG dataset from scratch for comparison. All of the results of fine-tuning are better than the result of the network trained from scratch (the column of "C-S" in Table 4.3), which is consistent with the observation in [Yos+14]. Through fine-tuning, we can obtain relatively good classification accuracies that are higher than those of traditional methods based on hand-crafted features. The detailed results of traditional methods can be found in the second last column of Table 4.8, with the highest attained accuracy being 80.65%, which is lower than any accuracy obtained by fine-tuning (*i.e.*, "C-1" to "C-7" in Table 4.3). A possible explanation is that having a large number of NIs from ImageNet (to our knowledge, no CG image exists in ImageNet) is beneficial for the network during its pre-training, and this helps the network understand the "intrinsic" properties of NIs, one of the two classes that we want to distinguish in our work.

Our proposed NcgNet copes better with this forensic problem. Here we first analyze

Table 4.3: Classification accuracy of fine-tuning different layers of CaffeNet on dataset of Google vs. PRCG. Accuracy is computed on all patches of 240×240 pixels in the testing set of Google vs. PRCG. "C" stands for CaffeNet. "C-S" means training CaffeNet form scratch on Google vs. PRCG. "C-N" means fine-tuning the first N layers of pre-trained CaffeNet [Jia+14] with the remaining layers retrained using random weight initialization, where $N = 1, 2, \cdots, 7$.

Network	C-S	C-1	C-2	C-3	C-4	C-5	C-6	C-7
Accuracy (%)	76.85	81.22	82.08	82.23	81.17	**82.71**	82.13	82.06

Table 4.4: Impacts of different configurations related to convFilter layer on the classification accuracy of testing patches of 240×240 pixels on the Google vs. PRCG dataset. 'cF' is the abbreviation of convFilter. We show the median of results of 7 runs, each with random initialization of CNN.

Network	with cF	without cF	with cF and ReLU	cF with constraint
Accuracy (%)	**85.15**	84.51	83.97	82.35

the performance of the convFilter layer of our network. Table 4.4 lists the classification accuracy of four different configurations related to this convFilter layer: (1) our proposed network shown in Figure 4.1 with two cascaded convolutional layers at the beginning of network; (2) removing the convFilter layer from the proposed network; (3) inserting an additional ReLU activation layer in our network after the convFilter layer; and (4) adding the high-pass filtering constraint from [BS18] to the convFilter layer in our network. Our configuration provides the highest accuracy, which demonstrates the utility of convFilter layer. The classification accuracy decreases when the convFilter layer is followed by ReLU activation, which may be an evidence that the relationship of "co-adaptation" between the convFilter layer and the successive convolutional group is weakened by ReLU activation to some extent. Adding a constraint to the convFilter layer, such as in Bayar and Stamm's work [BS18], also leads to performance degradation, which means that prior knowledge that is useful for image manipulation detection in [BS18] is not well suited for the task of NI and CG image classification.

4.2.4 Performance Evaluation

In this subsection, we compare the classification accuracy of the proposed method with that of state-of-the-art methods on patches of different sizes. As mentioned earlier, our network is slightly modified to accommodate different input sizes. The difference of networks used for different patch sizes is provided in Table 4.5. Furthermore, the max-pooling of the C1 of Net-3 has no stride (*i.e.*, the stride is equal to 1). The network

Table 4.5: Difference of networks used for different patch sizes. "$7 \times 7(2)$" means that the convolutional kernel size is 7×7 with a stride equal to 2, and all other strides are equal to 1.

	convFilter	C1	C2	C3	F4	F5
Net-1	7×7	$7 \times 7(2)$	5×5	3×3	4096	4096
Net-2	5×5	5×5	3×3	3×3	2048	2048
Net-3	3×3	3×3	3×3	3×3	2048	2048

adjustment is simple. For small patches, we only reduce the kernel size and remove the stride to guarantee the structural stability of our CNN with a fixed depth and ensure that the information flow can pass to the FC layers under an appropriately abstracted form. For Net-2 and Net-3, we cut the number of neurons of the FC layers in half to prevent over-fitting. The correspondence between networks and patch sizes is as follows: Net-1 for 240×240 and 180×180; Net-2 for 120×120 and 60×60; and Net-3 for 30×30. The corresponding input sizes of the networks for the five patch sizes are 233×233, 169×169, 107×107, 51×51, 27×27.

Figure 4.2 shows a comparison of the classification accuracy of our method and those of three hand-crafted-feature-based methods (Spam [PBF10], Geo [Ng+05] and Mfra [Pen+17]) under different patch sizes. For each patch size and method, the experiment is repeated seven times with different randomized initialization/parameterization to enhance the statistical significance of the results. We show the median of the results obtained by seven runs. Our method demonstrates the best performance for all patch sizes, followed by Geo and at last Mfra as the worst. As mentioned earlier, the classification performance of conventional methods based on hand-crafted features mainly depends on the discriminability of features. These features do not appear to be discriminative in this complex and challenging setting of Google versus PRCG. By contrast, our method automatically learns, as much as possible, useful and task-specific information from available data with the aid of the powerful learning capacity of CNN. Such automatic learning and unified "end-to-end" optimization for this classification task is a better choice than previous two-stage solutions. The accuracies of almost all methods decrease with decreasing patch size. This observation is understandable because smaller patches intuitively contain less information. Therefore, correctly classifying them is difficult for computational forensic algorithms and even human beings. The numerical results that correspond to the median accuracies shown in Figure 4.2 can be found in the group of columns labeled "Original" in Table 4.6. The performance improvement of our method compared with the second-best method, Geo [Ng+05], varies between 2.48% and 4.50% depending on the patch size.

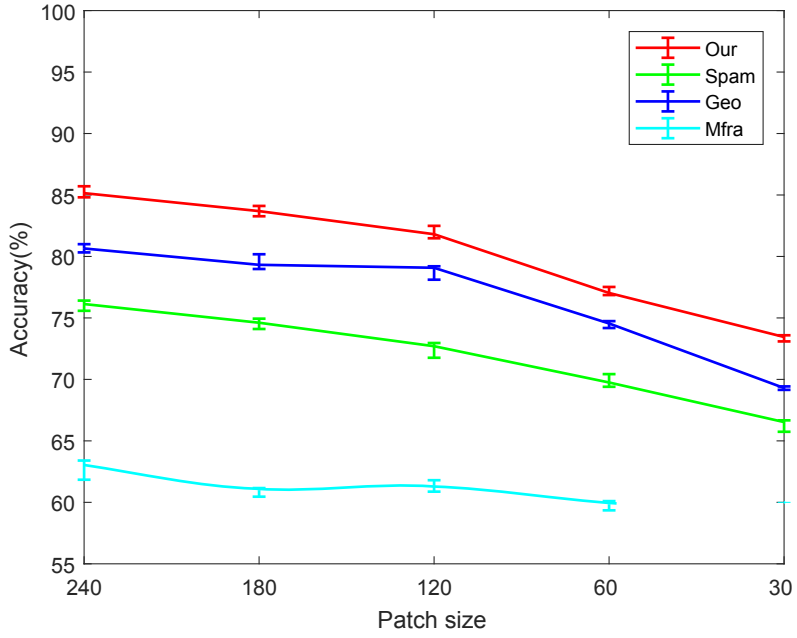


Figure 4.2: Comparison of our method with three state-of-the-art methods under different patch sizes. The solid lines show the median of 7 runs, and the error bars illustrate the maximum and minimum. The result of Mfra for 30×30 patches is not indicated because Mfra fails to extract features from such small patches.

Table 4.6: Classification accuracies (%) for different testing settings: Original, Scale300, Scale1000, JPEG90, and JPEG80. "-" means that in this case, the Mfra method cannot successfully extract features. For each testing setting, we show the accuracies of four methods (in group of four columns) on patches of five different sizes (in row). For every testing setting, the first column (our method) within the group of four columns always has the highest accuracy.

Patch	Original				Scale300				Scale1000				JPEG90				JPEG80			
	Our	Geo	Spam	Mfra	**Our**	Geo	Spam	Mfra	**Our**	Geo	Spam	Mfra	**Our**	Geo	Spam	Mfra	**Our**	Geo	Spam	Mfra
240 × 240	85.15	80.65	76.13	63.04	84.33	76.38	64.09	56.15	85.01	80.04	73.62	61.44	83.52	76.94	65.53	58.35	82.39	75.82	63.47	59.50
180 × 180	83.69	79.32	74.61	60.94	83.63	75.18	63.67	55.94	83.78	78.77	72.04	59.27	81.79	75.58	64.05	56.13	80.92	75.05	62.70	57.26
120 × 120	81.81	79.08	72.70	61.37	80.65	74.82	63.41	57.56	81.72	78.44	71.64	60.04	79.10	74.50	64.21	57.98	77.56	73.82	62.43	58.10
60 × 60	77.03	74.55	69.75	59.81	76.71	72.59	61.13	57.75	76.98	74.12	69.64	-	74.69	71.32	62.80	56.98	73.73	71.51	61.34	56.85
30 × 30	73.45	69.30	66.53	-	72.38	67.96	57.61	-	73.33	69.19	66.16	-	71.05	67.55	60.81	-	69.80	66.61	59.71	-

Next, we analyze the robustness of NcgNet. An effective image forensic algorithm should not only correctly deal with original data, which is Columbia's testing data in our experiments, but should also have a good level of robustness on post-processed data because post-processing is likely to occur either as a routine operation or an intentional attack. To evaluate robustness, we perform tests against two typical post-processing operations of rescaling and JPEG compression. For rescaling, we consider down-scaling

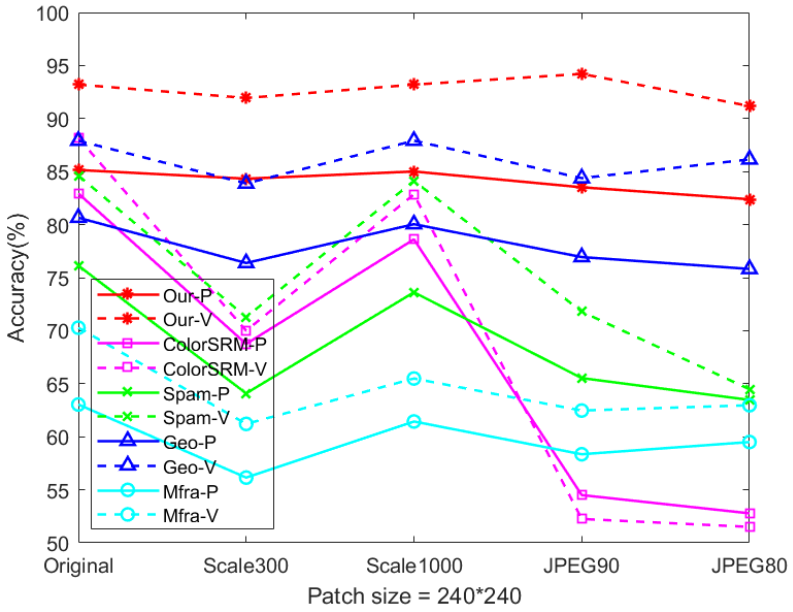


Figure 4.3: Classification accuracies of the five methods on five different testing sets. The patch size is 240×240. "-P" is the accuracy on patches, and "-V" is the accuracy after voting on full-sized images.

and up-scaling. Section 4.2.2 indicates that all images to be classified are resized in a pre-processing step before they are fed into CNN so that the shorter edge of the resized image has 512 pixels. Therefore, we test our trained network on testing sets, including images with a shorter edge rescaled to 300 pixels (simulation of post-processing) and then resized back to 512 pixels by the pre-processing of our method ("Scale300") and images with a shorter edge rescaled to 1000 pixels and then to 512 pixels ("Scale1000"). We compare the results with the baseline setting (denoted by "Original"). We use bicubic interpolation to rescale the image while preserving its aspect ratio, and we intentionally choose 300 and 1000 pixels for rescaling to avoid the potential side effect induced by the divisor and multiple of 512 (*e.g.*, 256 and 1024). As for JPEG compression, in the first place we consider two typical quality factors: 90 ("JPEG90") and 80 ("JPEG80").

For all methods, we select the trained model, which provides the median classification accuracy of seven runs in the "Original" setting, to perform this robustness test. All testing results are reported in Table 4.6. Mfra fails to extract features from 30×30 patches because the patch is too small. For all five testing settings and five patch sizes, the performance of our method is stable and always better than that of the three other methods. As an example, for Spam, the average performance drop of "Scale300" on all patch sizes is 9.962%, whereas the corresponding value of our method is only 0.686%. In

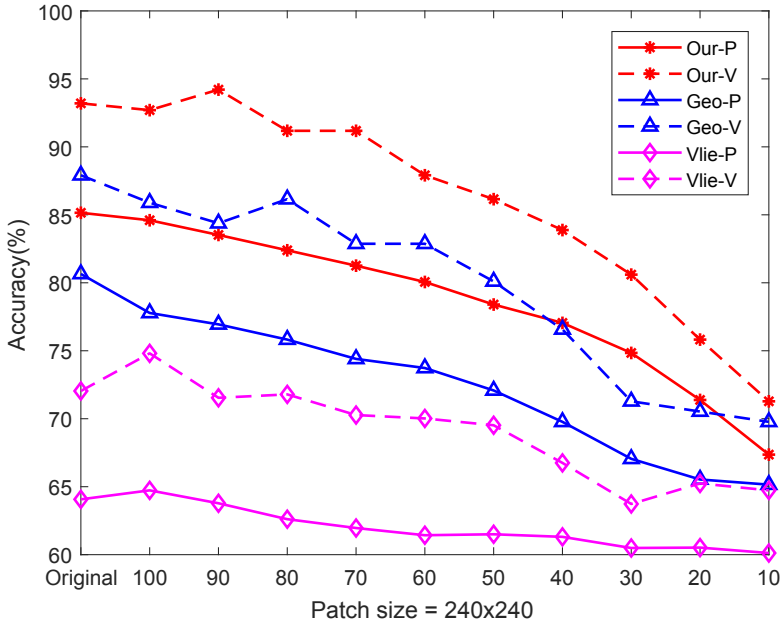


Figure 4.4: Classification accuracies of our method, Geo [Ng+05], and Vlie [ZWN12] for a large range of quality factors, *i.e.*, from 100 to 10 with a step of 10. The patch size is 240×240. "-P" is the accuracy on patches, and "-V" is the accuracy after voting on full-sized images.

Figure 4.3, the solid lines show the classification accuracies of the five methods on five different testing sets for 240×240 patches. Our method has stronger robustness than the other methods. The two post-processing operations slightly change the correlation of pixels with their local neighborhoods, and conventional methods, especially the ColorSRM method [GFC14] and Spam method, might be sensitive to this subtle alteration of local statistical property. By contrast, our method is almost insensitive to rescaling and quite robust against JPEG compression. This robustness can be attributed to the diversity of the challenging dataset and the powerful learning capability of CNN.

In addition, we investigate the JPEG robustness of our method, Geo [Ng+05], and Vlie [ZWN12] for a large range of quality factors, that is, from 100 to 10 with a step of 10. The corresponding results are shown in Figure 4.4. Compared with Geo and Vlie, our method always demonstrates the best performance on patches and full-sized images under all considered factors. Although the results of Vlie remain relatively stable, the accuracies on patches and full-sized images are the lowest with or without (corresponding to the "Original" case in Figure 4.4) JPEG post-processing. When the quality factor is very low (*e.g.*, 20 and 10, although such factors are rarely used in real-world applications), the performance of our method drops more rapidly than that of Geo but remains the

Table 4.7: Comparison of classification accuracy (%) of NcgNet with that of StatsNet on four datasets: Raise vs. Level-Design, Google vs. PRCG, Personal vs. PRCG, and Personal+Google vs. PRCG. "StatsNet" refers to the case where the number of training patches is almost the same as (in fact slightly larger than) 40,000 used in the original paper [Rah+17], and "StatsNet2" refers to the case where more training samples are used with 200 patches cropped from each training image (the same as in 'Our' method).

Method	Raise vs. Level-Design		Google vs. PRCG		Personal vs. PRCG		Personal+Google vs. PRCG	
	patch	full-size	patch	full-size	patch	full-size	patch	full-size
Our	94.75	99.58	77.03	88.41	95.60	97.25	80.15	86.43
StatsNet	89.76	99.30	67.50	75.31	63.63	75.75	69.68	75.38
StatsNet2	89.68	99.30	68.27	76.57	65.56	80.00	68.51	69.51

best among the three methods.

In the following, we experimentally compare the classification performance of NcgNet with that of a parallel work (StatsNet [Rah+17]), not only on their dataset (Raise versus Level-Design comprising 1,800 CG images from the Level-Design Reference Database [Pia17] and 1,800 photographic images randomly selected from RAISE dataset [DN+15]) but also on all the three datasets described in Section 4.2.1. The results are reported in Table 4.7. For StatsNet, we use the authors' shared implementation [Rah17] and follow the default setting described in [Rah+17]. The patch size of StatsNet is 100×100, while we use patches of 60×60 pixels for our method. This is a disadvantageous setting for our method because smaller patches contain less information. However, although the patch size of our method is nearly a quarter of that of StatsNet, our network performs consistently better on all the four datasets (Table 4.7). Furthermore, instead of using the default number of training samples as described in [Rah+17], we increase the amount of training data of StatsNet to match that of our method (*i.e.*, cropping 200 patches from each training image), and this network variant is denoted by StatsNet2. The last two rows in Table 4.7 show that in general, no guaranteed performance improvement from StatsNet to StatsNet2 is observed, although a large amount of data is used for training. A possible reason is that StatsNet is a three-layer network with a limited number of parameters; thus, using more training data than necessary would not improve its patch classification accuracy considerably.

4.2.5 From Local to Global Decision

The local-to-global strategy, an important component of our framework, is highly flexible and produces local and global decisions. Such a strategy applies not only to our CNN-based method, which is related to data augmentation, but also to conventional methods

Table 4.8: Effect of local-to-global strategy on the classification accuracy (%) of the seven methods. "Our-DF" means that the activation output of FC5 is extracted as deep feature and an SVM classifier is trained. The second column corresponds to the case of training on the full-sized images, and the third and fourth columns correspond to the two cases of training on the local patches (240×240) then testing on either patches or full-sized images with voting. The first case does not apply to "Our" method and "Our-DF" method.

Method	Full-sized images	Local patches	
	full-size	patch	voting
Our	-	85.15	93.20
Our-DF	-	84.62	92.70
ColorSRM	86.63	82.93	88.16
Geo	86.14	80.65	87.91
Spam	81.86	76.13	84.63
Vlie	69.77	64.07	72.04
Mfra	65.49	63.04	70.28

based on hand-crafted features, as shown later in this subsection.

The accuracies obtained by our method after voting from patches of different sizes (ranging from 30×30 to 240×240 pixels) are as follows: 83.63%, 88.41%, 88.66%, 92.70%, and 93.20%. Accuracy after voting refers to the accuracy on full-sized images where the predicted label of each testing image is obtained via majority voting of the predictions of 29 cropped patches (we ignore the last one of the 30 randomly cropped testing patches to avoid tie votes). The voting result is improved when the patch size increases, but we find a very minor performance improvement for patches larger than 240×240, which lead to a more costly computation. Therefore, we select the 240×240 patch to produce the final voting result. In addition, the voting accuracy, that is, the classification accuracy on full-sized images, of our method is always higher than the corresponding values of existing methods, as can be seen from the last column of Table 4.8. In particular, a considerable performance improvement of 5.04% is observed for our method compared with the ColorSRM method, which is the best hand-crafted-feature-based method.

For a fair comparison with previous SVM-based methods, we extract CNN deep features (*i.e.*, the activation output of FC5) and train an SVM classifier with the same experimental setting of SVM-based methods (*i.e.*, cropping 10 patches from each training image). The results on local patches and full-sized images are reported in Table 4.8. Compared with the ColorSRM method (best among all hand-crafted-feature-based methods), our deep-feature-based method ("Our-DF" in Table 4.8) is improved by 1.69% and

4.54% on local patches and full-sized images, respectively. This improved performance indicates that our deep feature has better discriminative power than traditional hand-crafted features. In addition, our CNN-based method has slightly higher classification accuracies than our deep-feature-based method (compare the second and third rows of Table 4.8). This finding implies that merging feature extraction and classifier training into a unified "end-to-end" framework brings additional benefits and supports our motivation of developing a CNN-based method.

We then evaluate and verify the robustness of the voting accuracy against the post-processing operations, and the obtained results are shown by dashed lines in Figure 4.3. The comparison of the five dashed lines indicates that our method has a stable and consistently better performance than the three state-of-the-art methods.

Next, we validate this local-to-global strategy on hand-crafted-feature-based methods, and the results are shown in Table 4.8. Each method has three cases: train on full-sized images and test on full-sized images; train on local patches and test on local patches; and train on local patches and test on full-sized images using voting. These three cases correspond to the last three columns of Table 4.8, respectively. The first case does not apply to our methods. The accuracy of training on local patches and testing on full-sized images with voting is higher than that of directly training and testing on the full-sized images for the five conventional methods, and this can be observed by comparing the second and last columns of Table 4.8. A possible reason behind this improvement is that the local-to-global strategy increases the diversity of training samples to some extent. In this work, we use the simple majority rule to vote. This point can be further improved in our future work.

4.2.6 Further Analysis and Failed Examples

As reported above, our method demonstrates good performance in the highly challenging dataset of Google versus PRCG. We also conduct tests of the proposed method on other datasets, namely, Personal versus PRCG and Personal+Google versus PRCG and compare our method's performance with that of state-of-the-art methods. Figure 4.5 shows a comparison of patch classification accuracies under the two settings. In these two settings, our method still exhibits the best performance, especially for Personal versus PRCG (Figure 4.5(a)), where the classification accuracy remains stable for patches larger than 60×60 pixels. Table 4.9 presents the classification accuracies on full-sized testing images obtained after voting from 240×240 patches. We observe an accuracy improvement of 1.50% and 4.02% when we compare the result of our method to that of the second-best method (Spam and Geo, respectively) under the setting of Personal versus PRCG and Personal+Google versus PRCG, respectively. In addition, compared

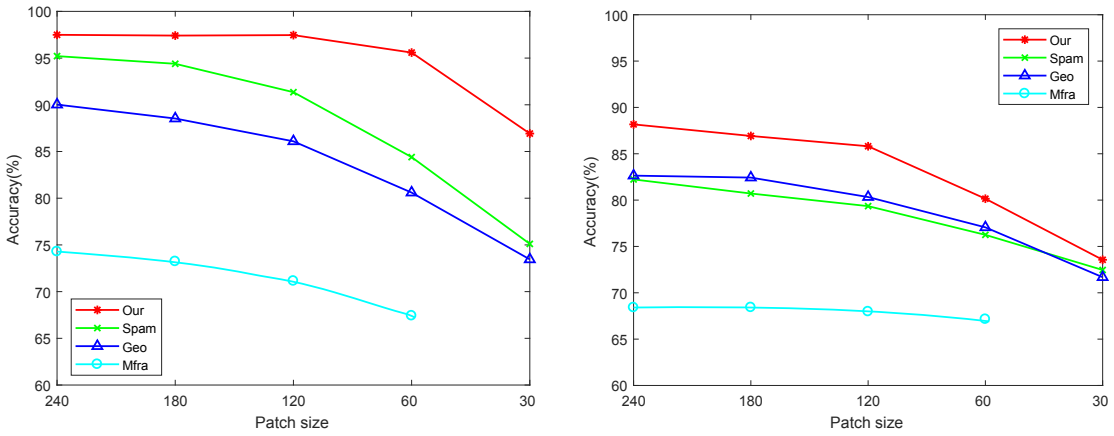


Figure 4.5: Comparison of patch classification accuracy on two other datasets: (a) Personal vs. PRCG and (b) Personal+Google vs. PRCG. The results of Mfra on 30×30 patches are not provided because of feature extraction failure on such small patches.

Table 4.9: Comparison of classification accuracy (%) on full-sized testing images, obtained after voting from 240×240 patches, on two other datasets.

Dataset	**Our**	Geo	Spam	Mfra
Personal vs. PRCG	98.50	95.50	97.00	82.75
Personal+Google vs. PRCG	93.13	89.11	88.61	72.19

with the setting of Google versus PRCG (the last column in Table 4.8), a noticeable performance improvement for the setting of Personal versus PRCG (the second row in Table 4.9) is observed for our method and existing methods. Our explanation is that the Personal set is simpler (*i.e.*, acquired by a small number of digital cameras) than the Google set. Thus, the classification is less difficult, and the result is improved for all methods.

We further analyze the results of our method in terms of two additional measures, namely, the error rate of CG patches misclassified as NI (denoted as CGmcNI) and its counterpart (denoted as NImcCG). The corresponding results are reported in Table 4.10. With decreasing patch size, these two measures increase for almost all testing settings, which is consistent with the previous findings (Section 4.2.4). The two error rates are often balanced. However, the NImcCG values of JPEG90 and JPEG80 are clearly higher than CGmcNI (last four columns of Table 4.10, particularly for small patches). A possible reason is that the details of natural patches are partially removed by JPEG compression. Thus, the NI patches, especially those of small sizes, become relatively "simple" and

Table 4.10: Statistics on misclassification rates (%) of NcgNet for different testing settings: Original, Scale300, Scale1000, JPEG90, and JPEG80. We consider the error rate of CG patches misclassified as NI (CGmcNI) and its counterpart (NImcCG). For each testing setting, we show these two error rates (in group of two columns) on patches of five different sizes (in row).

Patch	Original		Scale300		Scale1000		JPEG90		JPEG80	
	CGmcNI	NImcCG	CGmcNI	NImcCG	CGmcNI	NImcCG	CGmcNI	NImcCG	CGmcNI	NImcCG
240 × 240	15.67	14.03	16.65	14.67	15.70	14.26	16.06	16.90	19.13	16.06
180 × 180	15.95	16.67	16.45	16.29	16.20	16.26	14.63	21.84	17.87	20.32
120 × 120	19.20	17.17	22.93	15.70	20.28	16.24	17.55	24.30	21.55	23.33
60 × 60	20.98	25.00	23.77	22.80	22.00	24.06	19.52	31.18	21.15	31.47
30 × 30	25.70	27.41	28.12	27.11	27.02	26.31	24.15	33.82	24.97	35.52

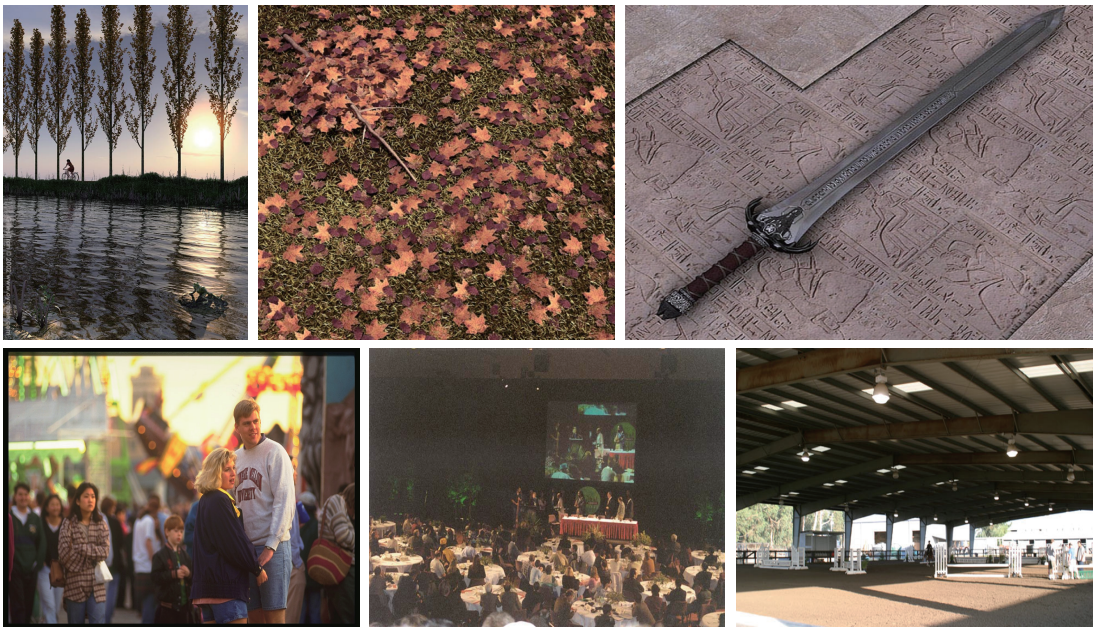


Figure 4.6: Failed examples: the top row corresponds to CG images misclassified as NIs, and the bottom row corresponds to NIs misclassified as CG images.

appear computer generated.

Figure 4.6 shows several failed examples of our method, including the case of CG images mistakenly classified as NIs (top row) and the case of NIs mistakenly classified as CG (bottom row). The light in the first image on the top row has good naturalness, and the color transition and texture of the two other images are rather plausible. Therefore, these CG images are misclassified as NIs by our network. On the contrary, the first two natural images on the bottom row have a certain degree of unnaturalness in light and

color, and the last image has a dramatic color transition (*e.g.*, the ceiling and shadow). These clues lead to the wrong classification.

4.3 Visualization and Understanding

Our CNN-based method exhibits good performance in terms of classification accuracy and robustness against typical post-processing operations. This characteristic is attributed to a well-designed and implemented CNN model. In this section, we wish to understand knowledge that is hidden in the data and cannot be reflected by the quantitative evaluation metrics used in Section 4.2. Specially, we analyze and understand what our CNN method has learned about the difference between NIs and CG images by using several advanced and appropriate visualization tools that are relevant to CNNs. This point was omitted in previous attempts of using CNNs for image steganalysis and forensics.

We study the filters that CNN has learned at its first layer. The first convolutional layer of CNN directly takes raw pixel data as an input and is thus more interpretable than other layers in the remaining part of the network [LJY17]. A common phenomenon exists in many CNNs well trained on natural images for computer vision tasks: the kernels they learn in the first layer are similar to Gabor filters and color blobs [Yos+14]. From the signal processing perspective, the convolution kernels of the first layer are linear filters. Therefore, a powerful analysis tool, that is, fast Fourier transform (FFT), can be used to analyze the properties of these kernels. Figure 4.7 shows the FFT of the kernels in the first layer of our CNN, our CNN with an additional high-pass filtering constraint from [BS18], and CaffeNet pre-trained on ImageNet. The filters are organized in groups of three (in columns), which correspond to the three color channels B, G and R. Many kernels in the first layer with the constraint of [BS18] [Figure 4.7(b)] have an apparent high-pass response, whereas the convFilter kernels of our method [Figure 4.7(a)] mainly capture the band-pass frequency information. However, the high-pass filtering constraint reduces the performance by approximately 3% (see results in Table 4.4). This performance drop is evidence that the band-pass information in a certain range of frequency may be more useful for identifying NIs from CG images than that at high frequencies. Furthermore, the first group of 96 filters of the first layer in CaffeNet shown in Figure 4.7(c) are highly consistent among the three color channels, but this consistency slightly decreases in the last group of 96 filters, as shown in Figure 4.7(d). The former collects orientated information, and the latter considers color to some extent. By contrast, almost all the filters in our method show no apparent consistency among the three channels, as illustrated in Figure 4.7(a), which implies that this identification task between NIs and CG images is more color-sensitive than conventional computer vision tasks.

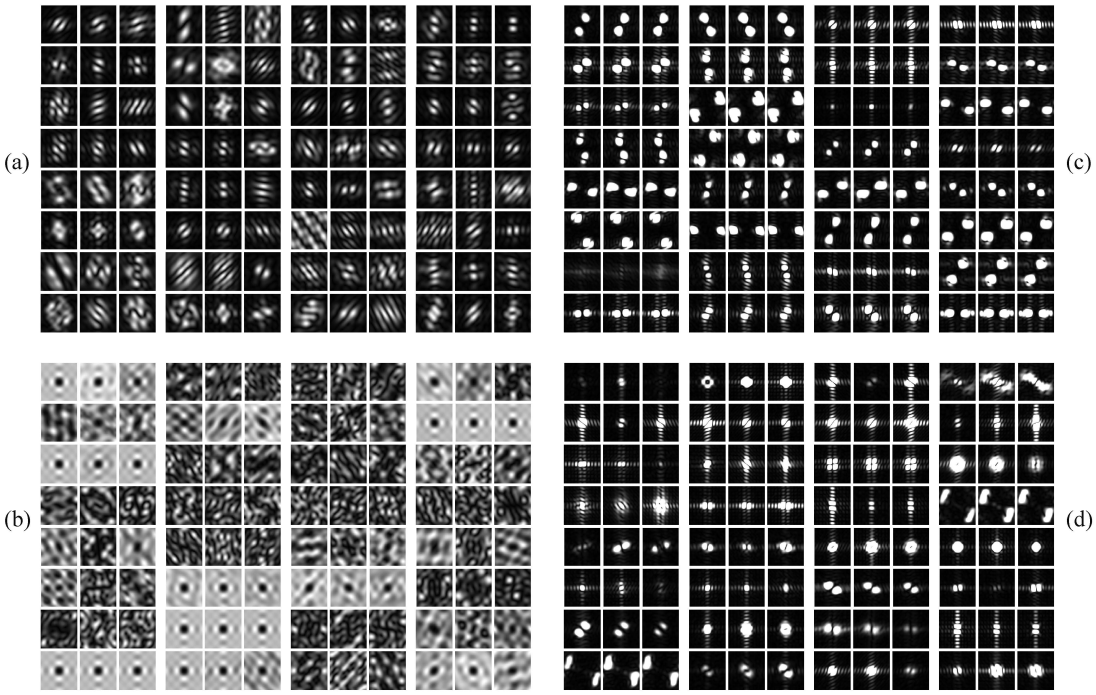


Figure 4.7: Visualization of the FFT of the first-layer filters in our NcgNet (a), NcgNet with the constraint from [BS18] (b), and pre-trained CaffeNet [Jia+14] (c,d). (c) corresponds to the first 96 kernels of the first layer in CaffeNet and (d) corresponds to the last 96 kernels. The filters are organized in groups of three (in columns) corresponding to the three color channels B, G and R. Brighter pixels mean higher values.

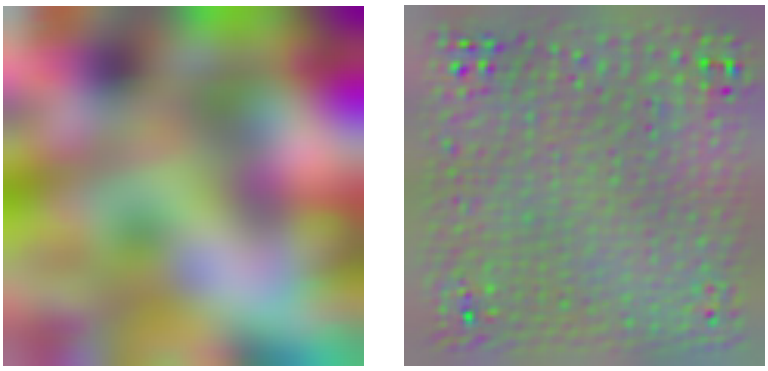
To summarize, we obtain the following observations concerning filters in the first layer. Image forensic tasks may need a new set of appropriate filters aside from those tailored for computer vision tasks, but not necessarily high-pass filters as suggested in [BS18]. Different forensic tasks may require different, adequate filters that can be learned with or without constraint. An appropriate constraint may improve performance as shown in [BS18], whereas an inappropriate constraint may decrease the performance as shown in our paper. In the latter case, "freely" learning these filters is a better solution. Further studies should examine the interesting research problem of the design and training of CNN and its layers for different forensic problems.

In the following, we continue our analysis of what our CNN has learned and what inspiration we can obtain from the well-trained model. Through two advanced visualization tools, namely, layer-wise relevance propagation (LRP) toolbox [Lap+16] and deep visualization toolbox [Yos+15], we analyze the trained model from the *data-centric* and *network-centric* point of view, respectively. The network-centric approach only requires the trained network for its analysis, and the data-centric approach additionally requires

Figure 4.8: Heatmaps of four sample image patches. The top row corresponds to NIs, and the bottom row corresponds to CG images. Each group consists of the input image patch (left) and its heatmap (right). The red color in heatmaps stands for a large value, blue for a small value, and black for an intermediate value.

passing sample data through that network.

LRP [Bac+15] is a technique for determining the degree of local contribution in an individual input to the neural network's output. In practice, we can obtain information about which pixels in the input image are relevant to the prediction outcome of the CNN by using the LRP toolbox [Lap+16]. The relevance scores assigned to the pixels can be visualized as an image with the same size as the input image, which is called a *heatmap*. Figure 4.8 shows four sample image patches and the corresponding heatmaps on our trained CNN model. Here, the CNN model does not use batch normalization due to the limitation of LRP toolbox. Figure 4.8(a) and (c) are NIs from Google while (b) and (d) are CG images from PRCG. We observe several very red pixels (meaning high contributions) in the heatmap of (b) corresponding to bright parts on the forehead, shoulder, and arm in the CG image, which implies that the prediction of CNN is relevant to the unnaturalness of light. The same CNN regards the light in (a) as rather natural, which contributes to the prediction (see red pixels corresponding to the left collar and slightly red pixels on the nose and chin). For (d), high relevance in the bottom of car is observed because the transition of the shadow is unnaturally sharp, but the color transition in (c) is smoother and natural and thus contributes to the prediction. Hence, our CNN model uses the naturalness degree of light and the smoothness of color transition as important clues for NI and CG image classification. This finding also provides insights into possible directions for computer graphics algorithms to further improve the photorealism of



(a) (b)

Figure 4.9: Visualization of preferred inputs in image space for two output units. The left corresponds to CG image, and the right corresponds to NI.

rendered images and synthesized images [BZ17].

With the help of the deep visualization toolbox [Yos+15], we compute the preferred inputs in the image space for two output units located immediately prior to the final softmax layer, which are shown in Figure 4.9. The preferred input refers to the input image that causes the corresponding unit to have high activation. Figure 4.9(a) is filled by multiple color blobs, which implies that CG images often contain large color primitives and look relatively "simple". By contrast, the recurrent appearance of "light points" shown in Figure 4.9(b) implies that natural images have more variability and look rather "complex". This condition might be one of the main differences between NIs and CG images. Our observation is completely in line with the hypothesis and observation of Dang-Nguyen et al. [DNBDN15], who assumed that synthetic facial animations present a less complex pattern, whereas natural ones have a much more complicated variability. Our observations of static NIs and CG images are similar to those of [DNBDN15] that examined the difference between natural and CG facial videos. This similarity may imply that CNNs can, to some extent, unconsciously follow a similar idea of the hard intelligent work of researchers when facing similar problems. In the future, we plan to design CNNs for the discrimination of natural and CG videos, and we expect to gain similar understanding and observation as those reported in [DNBDN15].

4.4 Summary

We proposed a generic framework based on the convolutional neural network to identify and understand the difference between natural and computer-generated images. The

performance of our network is better than that of conventional methods and a recent parallel CNN-based method not only on the highly challenging Google versus PRCG dataset, but also on relatively simple datasets. Our method also outperforms state-of-the-art methods in terms of performance in small image patches and robustness against typical post-processing operations. These factors are important for a forensic method to be useful in real-world applications.

We attempted to conduct an extensive study on using CNN for distinguishing between NIs and CG images. We considered the fine-tuning, structure, energy function design, flexibility, visualization, and understanding of CNN. To our knowledge, fine-tuning of pre-trained CNN from computer vision tasks and the visualization and understanding of what a CNN has learned are new in image forensics and might be useful and inspiring for other multimedia security tasks. Our source code is available at https://github.com/weizequan/NIvsCG.

Chapter 5

Identification of CG Images with Feature Diversity Enhancement and Learning from Harder Samples

Contents

On the basis of the research work in chapter 4, this chapter will continue to study the problem of CG image identification, and make improvements from the three aspects of dataset construction, network architecture and network training to obtain better classification accuracy and generalization capability.

For the CG image forensic problem, researchers have proposed hand-crafted-feature-based methods and CNN-based methods. Due to the powerful learning capacity of CNN, the CNN-based methods often achieve better forensic performance; however, the *blind detection* problem (or the so-called *generalization* problem) has been omitted in existing methods. This problem occurs when we train a CNN model using CG images from "known" computer graphics rendering techniques, and then test the model on images generated by "unknown" rendering techniques. Take Figure 5.1 as an example, the CG images in second, third, and fourth columns (rendered by Autodesk, Corona, and VRay,

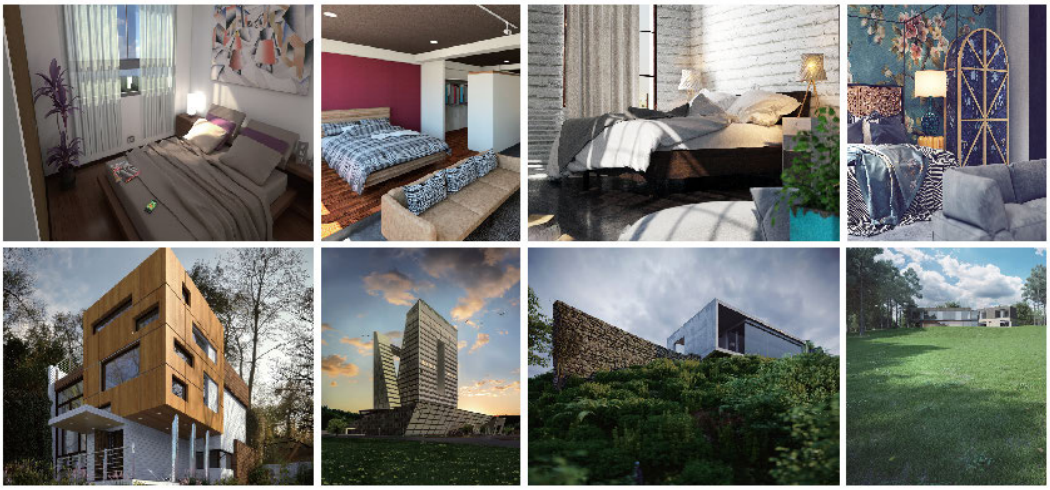


Figure 5.1: From left to right: four groups (columns) of computer-generated images which were rendered by Artlantis [Art], Autodesk [Aut], Corona [Cor], and VRay [Vra], respectively.

respectively) are misclassified as NI by a CNN model trained on NIs and CG images rendered by Artlantis (first column). The misclassification is probably due to the existence of subtle and different "intrinsic" traces left by each rendering technique, *e.g.*, in color use and light-material interaction. It is worth mentioning that this problem can be frequently encountered in practice when deploying detectors of CG images in real-world applications, as there can always exist CG images generated by new and/or customized rendering tools.

To improve the forensic performance, especially the generalization capability, we make efforts in two aspects of CNN: network architecture and network training. The core idea is to design and implement CNN with more *diversity* in feature learning and with the use of harder *negative samples* in the so-called enhanced training. The negative sample means the artificially constructed image by only using the original training dataset (potentially combined with information from CNN model), and its ground-truth label is same as that of CG image.

Specifically, we design a two-branch neural network which can capture more diverse features. Chapter 2 considered the generalization problem in the task of colorized image detection, and proposed negative-sample-based enhanced training to effectively improve the generalization performance of CNN. In Chapter 2, we used linear interpolation of paired natural and colorized images to construct negative samples. Although this requirement of paired images is not satisfied for the classification of NIs and CG images, we extend this interpolation method in a straightforward way to the unpaired setting and find that it can still improve the CNN's generalization for the CG image forensic

problem. This data-centric method only uses training images and is "blind" to the CNN model; therefore, motivated by a potential performance boost, we propose a new and more effective method (so-called model-centric method) by *coupling* the negative sample generation with the gradient information of CNN loss function.

Our contributions are summarized as follows:

- We for the first time in the literature raise and study the generalization issue of the CG image forensic problem. For the experimental study of this generalization problem, we collect four computer graphics datasets which were generated by four different rendering tools.
- We design a new network which has better generalization. The beginning part of the network has two branches with different initializations for the first layer.
- We propose a novel and effective model-centric method to generate negative samples. Given a trained model and a CG image, we iteratively modify this image via gradient-based distortion to make the distorted version close to the decision boundary of the CNN model. The gradient can be easily computed using back-propagation.

This piece of work is in press for publication in the international journal "Forensic Science International: Digital Investigation" [Qua+20] (https://doi.org/10.1016/j.fsidi.2020.301023).

5.1 Proposed Method

For the CG forensic problem, current CNN-based approaches can achieve high classification accuracy. However, the performance of these forensic detectors often drops when testing the trained model on CG images generated by "unknown" computer graphics rendering tools. To solve this generalization problem, in this work, we consider two aspects of CNN: network architecture and network training. Our network design is inspired by the work of Chapter 3 [Qua+19a] about the impact of CNN's first layer on forensic performance, where we proposed a simple criterion to combine the predictions of two independently trained networks for obtaining the final result. In this chapter, we design and implement a novel two-branch CNN model and apply different initialization strategy to the first layer of these two branches. This network can be trained in the end-to-end manner, and we expect to enrich the diversity of learned features through this ensemble-like design. For the network training, we adopt the enhanced training framework proposed in [Qua+19b]. An important component of enhanced training is

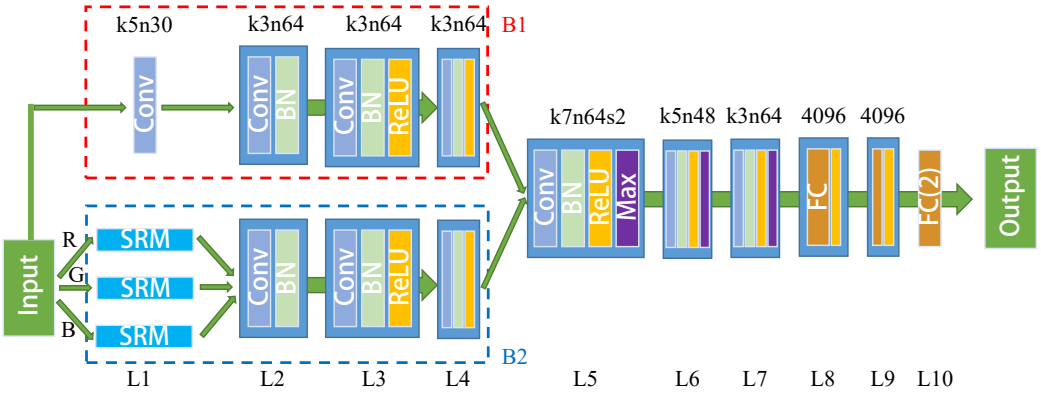


Figure 5.2: Architecture of our network named ENet (Ensemble Network). The network input is a 233×233 RGB image, and output is the class scores. For each convolutional layer, k is the kernel size, n is the number of feature maps, and s is the stride. "FC(2)" stands for a fully-connected layer with a 2-dimensional output of class scores. No padding exists in our network.

the so-called negative samples, which are generated in [Qua+19b] by linear interpolation of paired natural image and colorized image and which are deemed to be more difficult to classify. In this chapter, we propose two types of generation methods of negative samples (*same label as CG images*): (1) *data-centric* method, which constructs negative sample via linear interpolation of unpaired NI and CG image. (2) *model-centric* method, which generates negative sample by modifying the CG image based on the gradient of CNN.

5.1.1 Network Design

The standing point of network design is to enrich the diversity of feature learning. Inspired by the observation in [Qua+19a] and ensemble learning, we design a novel two-branch network to automatically and efficiently combine the kernels initialized with SRM filters [FK12] and Gaussian random distribution in the beginning of network, and it can be trained in the standard end-to-end way. This novel network is denoted by ENet (Ensemble Network), and the corresponding network architecture is shown in Figure 5.2. In practice, our network takes the NcgNet proposed in [Qua+18] as backbone: the beginning part (from L1 to L4 in Figure 5.2) has a new two-branch design; starting from L5, the network architecture is same as NcgNet. The input of ENet is an RGB image. After the first layer (so-called filter layer), we use three convolutional layers without pooling operation (L2-4) to analyze the filtered signal. The analysis results are concatenated (in the channel-wise manner) as the input of L5, and three consecutive convolutional layers (L5-7) and two fully-connected layers (L8-9) are applied to conduct high-level abstraction and reasoning. The last layer (L10) with softmax maps the high-level feature vector

to the 2-dimensional class scores (NI and CG). In total, ENet is a 10-layer network.

In our network, from L2 to L4, we do not use any pooling operation so as to retain useful discriminative information, which also helps for improving the diversity between the two branches. In the second branch (B2 in Figure 5.2), the SRM means that the convolutional kernels are fixed as the thirty 5×5 SRM residual filters borrowed from [FK12]. These three SRM blocks are applied to each color channel of input image, *i.e.*, R, G, and B, respectively. Then, all output channels are directly concatenated together to form a ninety-channel input of the second convolutional layer (L2). Except for the first layer (L1), each Conv is equipped with batch normalization (BN) layer. And we do not add activation layer after L2 to preserve useful information as much as possible. Following [Qua+18], all max-pooling layers have the same kernel size of 3×3 and a stride of 2.

5.1.2 Data-Centric Method

This method constructs negative sample by linear interpolation, and the corresponding formulation is:

$$I_{NS} = \alpha \cdot I_{NI} + (1 - \alpha) \cdot I_{CG}, \tag{5.1}$$

where I_{NS} is the negative sample, I_{NI} is the natural image, I_{CG} is the computer-generated image, and $\alpha \in \{0.1, 0.2, 0.3, 0.4, \cdots, 0.9, 0.99\}$ is the interpolation factor. For each factor, we randomly combine the NI and CG image of original training dataset. To clearly illustrate this process with examples, we select four CG images, and then randomly select an NI for each CG image to generate the negative sample via Eq. 5.1 with three different factors ($\alpha = 0.1, 0.5, 0.9$). The corresponding results are shown in Figure 5.3. When α increases [from Figure 5.3(c) to Figure 5.3(e)], the negative samples are progressively getting closer to the natural images [Figure 5.3(b)]. In addition, we allow the use of larger interpolation factor than the work in Chapter 2 [Qua+19b], *i.e.*, $\alpha > 0.4$, mainly due to the "blind" nature of this method, *i.e.*, it is blind to the decision boundary of trained CNN model.

5.1.3 Model-Centric Method

The model-centric method in this chapter is more related to adversarial samples in the field of machine learning. For the consistency of description, this section first briefly reviews adversarial samples. In addition, readers could refer to recent surveys [AM18; Yua+19] for a comprehensive coverage of this rapidly evolving topic.

Szegedy et al. [Sze+14] found an intriguing phenomenon: several high-performance

Figure 5.3: Examples of data-centric method. From top to bottom: (a) four CG images rendered by Corona [Cor]; (b) randomly selected NI for each CG image; (c), (d), and (e): the negative samples generated by Eq. 5.1 with CG image and NI in (a) and (b) of the same column, where the interpolation factor α is 0.1, 0.5, and 0.9, respectively. We calculate the PSNR (peak signal-to-noise ratio) for the negative samples [(c), (d), and (e)], with the CG image [(a)] as the reference. PSNR values (in dB) from left to right: (c) 28.12, 28.42, 28.09, and 27.32; (d) 14.24, 14.44, 14.31, and 13.12; (e) 9.14, 9.33, 9.20, and 8.02.

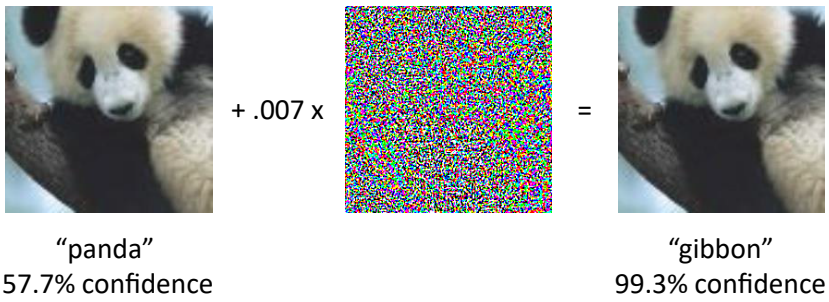


Figure 5.4: An example of adversarial sample. Network is the pre-trained GoogLeNet [Sze+15] on ImageNet [Den+09]. This image comes from [GSS15].

machine learning models, including advanced deep neural networks, are susceptible to *adversarial examples* (or adversarial attacks). When applying a small and imperceptible perturbation to a test sample, this perturbed version is very likely to be misclassified by trained deep models. Goodfellow et al. [GSS15] explained that linearity in high-dimensional spaces is the primary cause of neural networks' vulnerability to adversarial perturbation. Based on this linear view, they proposed the fast gradient sign method (FGSM) to generate adversarial examples, where the required gradient can be computed efficiently using backpropagation. Figure 5.4 illustrates an example of adversarial sample based on FGSM. By adding an imperceptibly small vector whose elements are of the same sign as that of the corresponding elements in the gradient of the cost function with respect to the input, GoogLeNet [Sze+15]'s classification of the image can be changed (from “panda” of 57.7% confidence to “gibbon” of 99.3% confidence). This gradient-based method is the backbone of many subsequent construction methods of adversarial examples. FGSM is essentially a one-step gradient-based method; therefore, a straightforward extension of this method is to apply it in a multiple-step fashion with smaller step size (in extreme cases, changing the value of each pixel only by 1 on each step) [KGB16]. Tramèr et al. [Tra+18] proposed to prepend FGSM by a small random step, which is based on the sign of a Gaussian distribution.

In computer vision and machine learning, adversarial examples have been used to improve the robustness of deep networks. There is less effort in the literature on using adversarial examples to improve network's generalization. To our knowledge, there is no such existing work in the image forensics community. In this chapter, we propose a refined and appropriate method to generate adversarial examples as negative samples (*i.e.*, simulated proxy of “unknown” CG images) for improving generalization. Specifically, our model-centric negative sample generation method is based on a new iterative version of FGSM, *i.e.*, we randomly select certain percent of pixels to be changed by 1 for each step. The essential motivation of our method is to strictly control the strength of attack (in other words and loosely speaking, the location of negative samples relative to the

decision boundary of CNN). Our method has different original intention when compared to the conventional adversarial attacks in the field of machine learning, where they prefer to maximally cross the decision boundary with as small as possible perturbation. Our method shares some similarities with the iterative strategy of Tondi [Ton18]; however, some differences exist: (1) [Ton18] uses adversarial examples to carry out attacks, while we use adversarial examples to improve the generalization of CNN-based forensic detectors which is to our knowledge new in the literature. (2) With some technical choices, our iterative version is more finely controlled in terms of the confidence level of negative samples, which is important for the enhanced training.

Model-centric method is related to the gradient-sign-based adversarial sample generation. In the following, we first recall the formulation of FGSM [GSS15] and its iterative variant [KGB16]. Then, we describe our iterative masked gradient sign method (IMGSM) for constructing negative sample.

Let $\mathbf{x}$ be the original (or clean) image, $\hat{\mathbf{x}}$ the perturbed version of $\mathbf{x}$ with expected target t, $\mathcal{M}$ a deep model and $J_{\mathcal{M}}(\mathbf{x}, t)$ the loss function (*e.g.*, cross-entropy loss) used to train the original model $\mathcal{M}$. The formulation of FGSM is

$$\hat{\mathbf{x}} = \mathbf{x} - \epsilon sign(\nabla_{\mathbf{x}} J_{\mathcal{M}}(\mathbf{x}, t)), \tag{5.2}$$

where hyper-parameter ϵ controls the magnitude of the perturbation.

An iterative variant of FGSM is

$$\hat{\mathbf{x}}^{k+1} = Clip_{\mathbf{x},\epsilon}\{\hat{\mathbf{x}}^k - \beta sign(\nabla_{\mathbf{x}} J_{\mathcal{M}}(\hat{\mathbf{x}}^k, t))\}, \tag{5.3}$$

where $Clip_{\mathbf{x},\epsilon}\{\mathbf{x}'\}$ is the operation which projects the image $\mathbf{x}'$ into the L_∞ ϵ-neighbourhood of the source image $\mathbf{x}$ [KGB16]. Usually, the value of β depends on the data type of image pixel (integer or float) and is set following the minimal distortion, *e.g.*, changing the integer pixel value only by 1 for each modification.

Our goal is to modify a CG image $\mathbf{x}$ and output a harder negative sample $\hat{\mathbf{x}}$ with predicted probability $\mathbf{p}$ as NI under original trained model. $\hat{\mathbf{x}}$ plays the role of simulated proxy of "unknown" CG image that may be encountered during testing. To exactly control the predicted probability of negative sample, we introduce the iterative masked gradient sign method (IMGSM). Compared with Eq. 5.3, our formulation of IMGSM has two differences: (1) we have no clip operation because we mainly consider the attack confidence of negative sample (*i.e.*, the probability $\mathbf{p}$) and do not need to strictly limit the magnitude of the perturbation. (2) we introduce the random-mask-based strategy to perturb the input image for each modification so that we can exactly control the attack confidence of negative sample, *e.g.*, falling into a certain interval. The formulation is

$$\hat{\mathbf{x}}^{k+1} = \hat{\mathbf{x}}^k - \beta \mathbf{m}_\lambda \odot sign(\nabla_{\mathbf{x}} J_{\mathcal{M}}(\hat{\mathbf{x}}^k, t)), \tag{5.4}$$

Algorithm 2 Iterative masked gradient sign method

Input: trained model $\mathcal{M}$, $\mathbf{x}$, attack confidence interval $[\mathbf{p}_{min}, \mathbf{p}_{max}]$, the maximal iterations $K = 200$, the initial mask probability $\lambda^0 = 0.96$.

Output: negative sample $\hat{\mathbf{x}}$.

Initialization: current mask probability $\lambda = \lambda^0$, $\mathbf{p}$ is set as the predicted confidence of $\mathbf{x}$ as NI.

1: **while** $\mathbf{p} < \mathbf{p}_{min}$ **and** $\lambda <= 0.998$ **do**
2: set $\hat{\mathbf{x}}^0 = \mathbf{x}$, $\hat{\mathbf{x}}_{cand} = \mathbf{x}$.
3: **for** $k = 0$ **to** $K - 1$ **do**
4: compute $\nabla_{\mathbf{x}} J_{\mathcal{M}}(\hat{\mathbf{x}}^k, t)$ using backpropagation.
5: compute $\hat{\mathbf{x}}^{k+1}$ via Eq. 5.4.
6: compute current attack confidence $\mathbf{p}$ of $\hat{\mathbf{x}}^{k+1}$.
7: **if** $\mathbf{p} > \mathbf{p}_{max}$ **then**
8: break.
9: **else**
10: set $\hat{\mathbf{x}}_{cand} = \hat{\mathbf{x}}^{k+1}$.
11: **end if**
12: **end for**
13: set $\hat{\mathbf{x}} = \hat{\mathbf{x}}_{cand}$, and compute the attack confidence $\mathbf{p}$ of $\hat{\mathbf{x}}$.
14: **if** $\lambda < 0.99$ **then**
15: $\lambda = \lambda + 0.01$.
16: **else**
17: $\lambda = \lambda + 0.002$.
18: **end if**
19: **end while**

where $\odot$ is the element-wise product operation and $\mathbf{m}_\lambda$ is the binary mask whose elements are randomly set as zeroes with probability λ. Note that $\mathbf{x}$, ∇, and $\mathbf{m}_\lambda$ have the same shape. In our experiment, the 8-bit integer pixel is scaled to float in $[-1, 1]$, therefore, we set $\beta = 2/255$ to guarantee the minimal distortion. In other words, after transforming back to the integer pixel value, $\pm\beta$ on float means adding or subtracting by 1.

Algorithm 2 illustrates the process of model-centric negative sample generation. To constrain the predicted confidence of generated negative sample belonging to a certain interval $[\mathbf{p}_{min}, \mathbf{p}_{max}]$, we introduce adaptive strategy to adjust the mask probability λ shown in line 14-18 of Algorithm 2. Starting from a minimal value of λ which experimentally leads to large distortion (thus large jump of $\mathbf{p}$) for each modification while iterating on k in line 3, the mask probability progressively increases when the predicted probability of generated negative sample cannot fall into the required interval. A mask with higher value of λ leads to milder increase of $\mathbf{p}$ in each iteration, with more chance

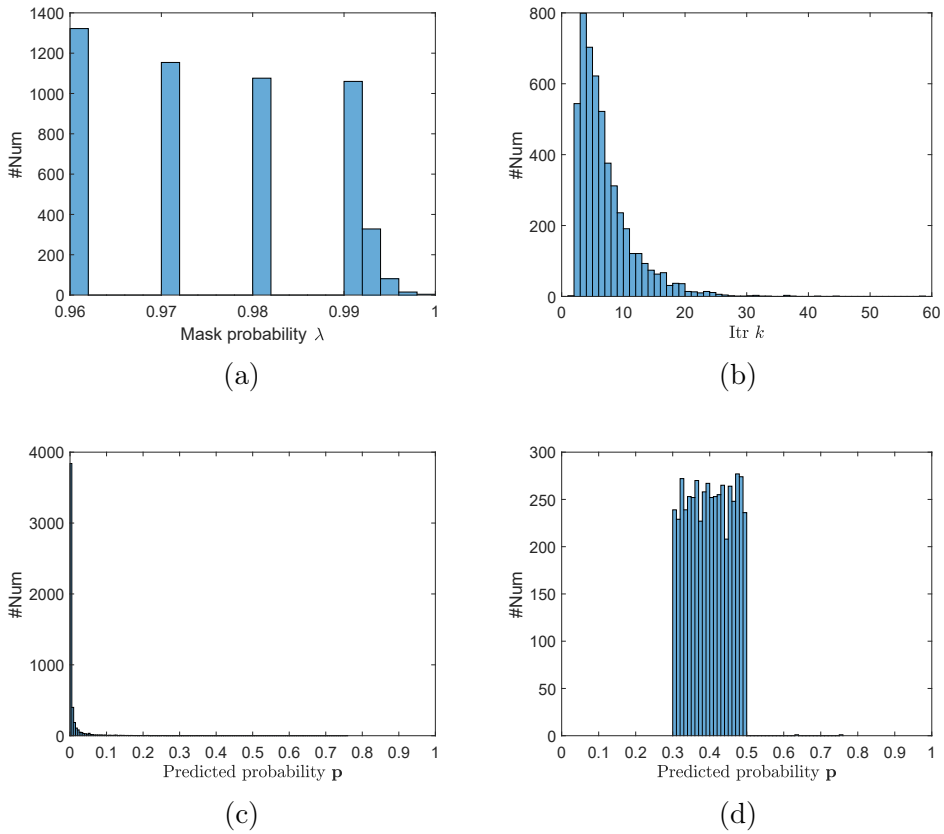


Figure 5.5: The statistics of model-centric negative sample generation: (a) mask probability λ, (b) iterations k, (c) the predicted probability $\mathbf{p}$ of original CG images, and (d) the predicted probability $\mathbf{p}$ of corresponding negative samples.

to meet the constraint of required interval for $\mathbf{p}$. In our experiment, we choose the confidence interval of generated negative sample as $[0.3, 0.5]$, which means the negative sample is close to original classification boundary and in the side of CG image.

To clearly illustrate the process of model-centeric negative sample generation with an example, we train the ENet on Corona dataset, and then generate the negative samples of CG images in training dataset (in total, 5040 images). Figure 5.5 shows the histogram of mask probability λ (a), iterations k (b), and predicted probability $\mathbf{p}$ of original CG images (c) and corresponding negative samples (d). Given a CG image, it is difficult to derive an algorithm which can theoretically guarantee that the predicted probability $\mathbf{p}$ of negative sample strictly falls into the expected interval (*i.e.*, $[0.3, 0.5]$ in our experiment). In practice, by using our adaptive mask strategy, we observe that experimentally almost all negative samples can successfully fall into this interval, like Figure 5.5(d), where only two samples are not in $[0.3, 0.5]$. This is because the two original CG images have $\mathbf{p}$

Figure 5.6: Four groups of CG sample (top row) and corresponding negative sample (bottom row) based on model-centric method. We calculate the PSNR for the negative sample, with the CG sample as the reference. PSNR values (in dB) from left to right: 56.32, 58.92, 56.63, and 56.25, respectively.

value larger than 0.5, *i.e.*, 0.76 and 0.64.

In addition, Figure 5.6 shows four groups of negative samples (bottom row) based on model-centric method and corresponding CG images (top row). From left to right: for CG sample, $\mathbf{p} = 2.40\text{e-}5, 4.92\text{e-}3, 1.95\text{e-}4, 2.01\text{e-}4$; for negative samples, $\mathbf{p} = 0.32, 0.38, 0.48, 0.35$. Comparing the two rows, we can find that the CG images and the corresponding negative samples are visually almost the same (see PSNR values in the caption of Figure 5.6). Furthermore, compared with the PSNR values reported in the caption of Figure 5.3, those of Figure 5.6 are obviously larger, which means that the perturbation introduced by model-centric method is much smaller than data-centric method. We also analyze the min/max perturbation for CG images in Figure 5.6, *i.e.*, the minimal/maximal perturbation value for pixels within an image. The min/max perturbations are from left to right: $-3/3$, $-3/2$, $-3/3$, and $-4/3$, respectively (pixel values are in the range of 0 to 255). These results demonstrate that when generating negative samples, gradient-based perturbation modifies very slightly the pixels of CG images in an almost imperceptible way.

5.1.4 Network Training

Basically consistent with Chapter 2, a complete network training includes two stages: normal training and enhanced training. The CNN model first conducts normal training

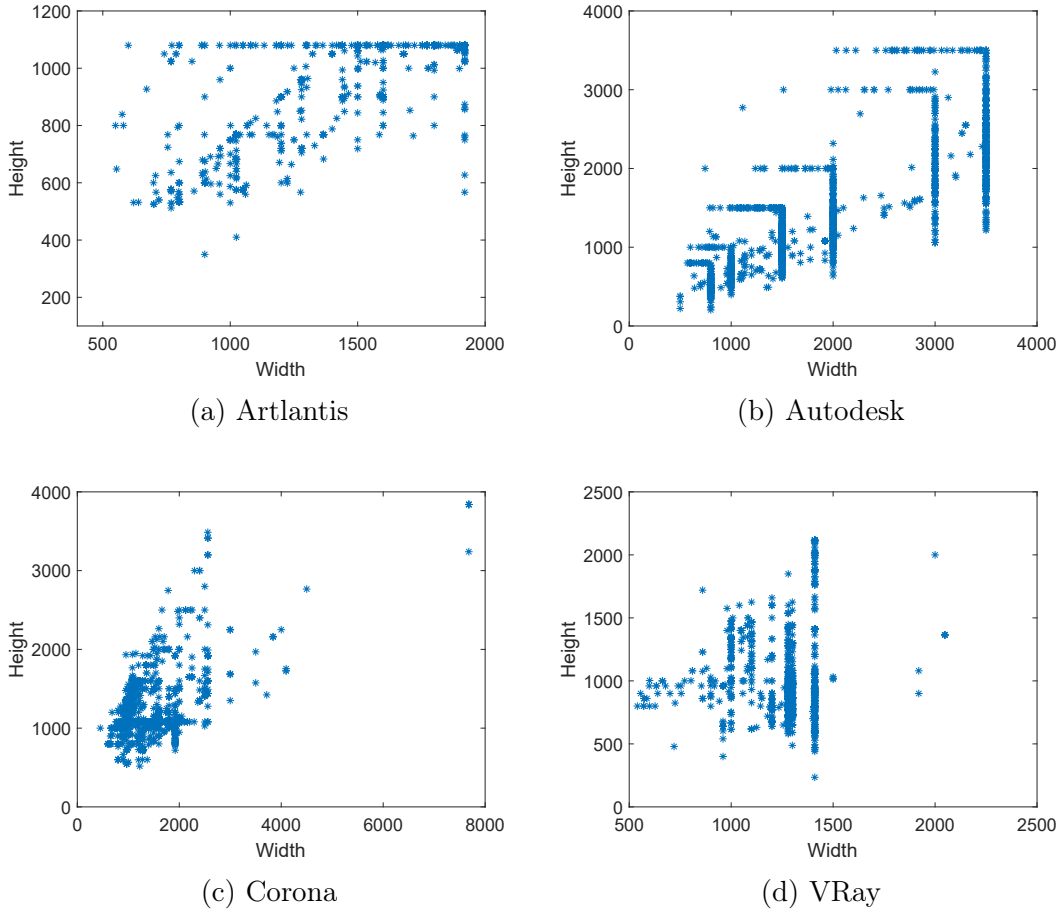
from scratch with the original training dataset $\mathcal{D}$, *i.e.*, NIs and CG images generated by "known" rendering tools. After the model converges, we continue to train the model with enhanced training based on negative sample insertion. This enhanced training is an iterative process, and the main pipeline is described as follows.

- We construct negative samples using Eq. 5.1 or Algorithm 2, and insert them into $\mathcal{D}$.
- We update the parameters of $\mathcal{M}$ using $\mathcal{D}$. Starting from the second half of training process, we compute the error rate r on the so-called *natural validation dataset* $\mathcal{V}$ and also mark the model as candidate model if its r is less than θ (a threshold that determines the accepted degree of final classification accuracy on NIs; in our experiment, we set $\theta = 4\%$).
- The above two steps interleave until reaching the stop condition. If the error rates on $\mathcal{V}$ starting from the second half of training process are all larger than θ, we stop the iteration process; otherwise, we stop when the number of iterations reaches maximal value Z: for data-centric method, $Z = 10$ (α can take 10 values, see in Section 5.1.2); for model-centric method, $Z = 20$.
- From all candidate models, we select the final model which has the maximal r.

5.2 Experimental Results of CG Image Identification

5.2.1 Dataset Collection

To study the generalization problem and validate our proposed method, we collect four CG datasets: Artlantis [Art], Autodesk [Aut], Corona [Cor], and VRay [Vra]. The CG images were downloaded from the websites of the four rendering software tools. The collected CG images have high level of photorealism and are very close to real-world scenes. Some examples are shown in Figure 5.1. The number of images of these four datasets are 1,620, 1,620, 1,593, and 1,579, respectively. Figure 5.7 and Figure 5.8 separately show the histogram of the image size and JPEG compression quality factor of the CG datasets. For each CG dataset, we randomly select 360 images as testing set, and the remaining images as training set (with the approximate ratio of 4:1). To guarantee the diversity of NIs, we combine two datasets of RAISE [DN+15] and VISION [Shu+17] in our experiments. RAISE is a collection of 8,156 raw images that were taken at very high resolution and we randomly select 4,700 images. In order to simulate the real-world setting, we randomly resize and compress these raw images. For each raw image, we first resize



(a) Artlantis (b) Autodesk

(c) Corona (d) VRay

Figure 5.7: The histogram of image size of CG datasets: (a) Artlantis; (b) Autodesk; (c) Corona; (d) VRay.

with bicubic interpolation and with the length of its shorter edge as an integer randomly sampled from the set of $\{500, 750, 1000, 1500, 2000, 2500, 3000\}$. Then, we compress the resized image with quality factor randomly sampled from the range of $[70, 100]$. VISION is composed of images captured by 35 mobile devices where each device includes 100 natural images (in total 3,500 images). In addition, these natural images were exchanged via the Facebook (high and low quality respectively) and WhatsApp social media platforms, and thus each image has four versions ("nat", "natFBH", "natFBL", and "natWA" in [Shu+17]). Considering the same content of these four versions, we randomly select one version for each image and obtain 3,500 images. In the end, we have 8,200 NIs from RAISE and VISION.

In the following, we provide the details of datasets used in our CG identification experiments. We randomly select 5,040 NIs and duplicate approximately 4 times of

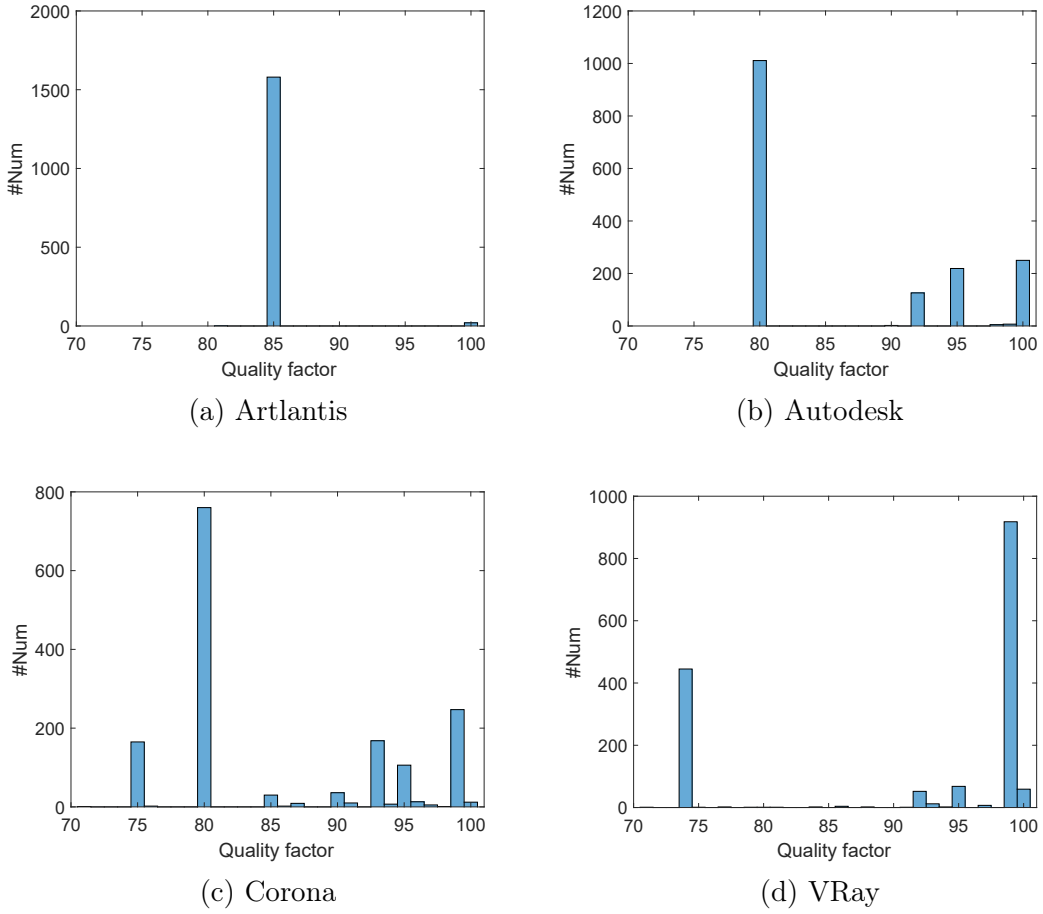


Figure 5.8: The histogram of JPEG compression quality factor of CG datasets: (a) Artlantis; (b) Autodesk; (c) Corona; (d) VRay.

each CG training set (5,040 CG images after duplication) to construct four final training datasets, corresponding respectively to the four rendering software tools (Artlantis, Autodesk, Corona, and VRay). From the remaining 3,160 NIs, we respectively select 360 NIs for each CG dataset and combine corresponding testing set (360 CG images) to construct four final testing datasets. The remaining 1,720 NIs constitute the so-called *natural validation dataset*, which is used for the final CNN model selection in the stage of enhanced training (the details are described in Section 5.1.4).

5.2.2 Experimental Settings

In this chapter, we consider two recent state-of-the-art CNN models of YaoNet [Yao+18] and NcgNet [Qua+18]. In [Yao+18], the images are first converted to grayscale and

then fed into YaoNet to reduce the computational complexity. For convenience and fair comparisons, we directly use RGB images as the input of YaoNet and this variant achieves better performance compared with grayscale input. Following [Qua+18], all images in our experiments are resized using bicubic interpolation so that the shorter edge of each resized image has 512 pixels, and for each image, we rescale its pixel values to $[-1, 1]$. The input size of network is 233×233. Stochastic gradient descent (SGD) with a minibatch of 32 is used to train ENet. For SGD optimizer, the momentum is 0.9 and the weight decay is 1e-4. The initial learning rate is 1e-3. For the normal training (only using original training dataset) of ENet, we divide the learning rate by 10 every 100 epochs, and the training procedure stops after 300 epochs. For the normal training of YaoNet and NcgNet, we follow the same learning rate schedule as described respectively in [Yao+18] and [Qua+18]. In the stage of enhanced training of these three networks, we adopt the same strategy about learning rate: the learning rate is continued to be divided by 10 every 15 epochs (one insertion) and fixed after 4 iterations of negative sample insertion to avoid learning rate becoming too small. Following [Qua+18], we adopt the standard 10-crop testing [KSH12]: given a testing sample, the network extracts five patches of 233×233 pixels (the center and four corner patches), flips these five patches in the left-right direction (*i.e.*, horizontal reflection), and then averages the predictions of total 10 patches as the final result. We employ the *half total error rate* (HTER) to evaluate the detection performance. The HTER is defined as the average of misclassification rates (in %) of NIs and CG images, here same as the overall error rate on balanced testing datasets. In this work, all reported results are the average of 5 runs.

5.2.3 Validation of Proposed Network

We validate our network architecture design of ENet in terms of the conventional classification performance and the generalization capability. All the networks are trained on Autodesk and tested on Artlantis, Corona, and VRay. The corresponding results are reported in Table 5.1. Compared with YaoNet, NcgNet and NcgNet_SRM, ENet demonstrates better performance for both "known" and "unknown" rendering engines. Here, the NcgNet_SRM is the variant of NcgNet where the first layer of NcgNet is replaced with three SRM blocks like the first layer of B2 in Figure 5.2. In addition, although NcgNet_Comb and ENet have comparable results, the former needs to train two models separately, *i.e.*, which inevitably and approximately doubles the number of parameters and training time.

Next, we evaluate the performance of four variants of our proposed ENet, *i.e.*, ENet_d, ENet_SRM, ENet_d_half, and ENet_rand. We remove the B2 of ENet (the blue dotted rectangle in Figure 5.2) and double the number of feature maps of L3 and

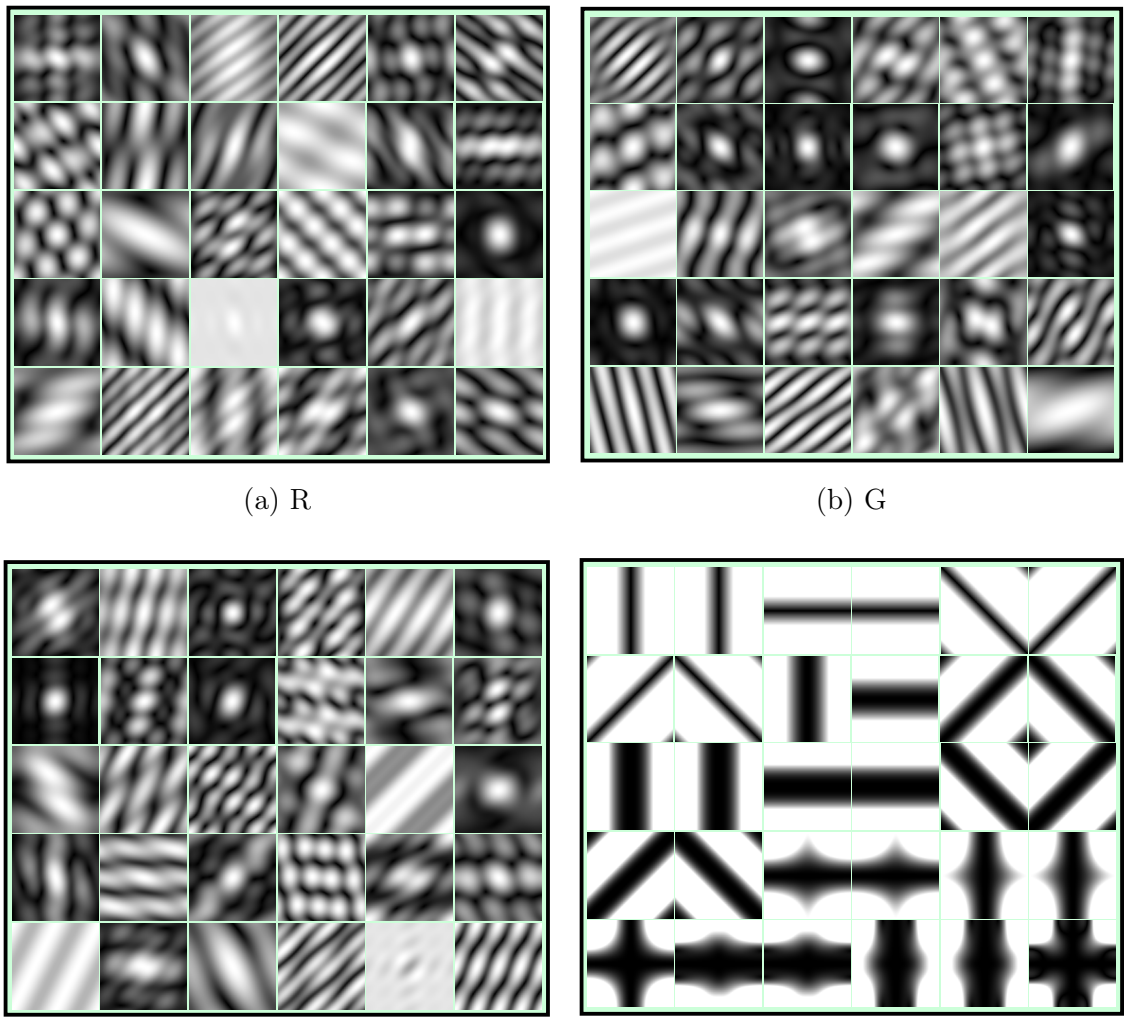
Table 5.1: The classification performance (HTER, in %, lower is better) of different network architectures. "NcgNet_Comb" is the combination result of predictions of "NcgNet" and "NcgNet_SRM" according to the combination criterion used by [Qua+19a]. For the sake of clarity, the results of generalization performance on "unknown" rendering engines are presented in italics.

Network	Autodesk	Artlantis	Corona	VRay
YaoNet	4.61	*28.14*	*15.00*	*21.17*
NcgNet	2.84	*16.61*	*12.78*	*16.58*
NcgNet_SRM	2.42	*17.97*	*8.97*	*16.09*
NcgNet_Comb	2.16	*13.14*	*7.75*	*12.61*
ENet	1.56	*10.39*	*7.67*	*13.39*

Table 5.2: The classification performance (HTER, in %, lower is better) of our proposed ENet and its four variants. All networks are trained on Autodesk. The results of generalization performance on "unknown" rendering engines are presented in italics.

Network	Autodesk	Artlantis	Corona	VRay
ENet_d	2.00	*15.06*	*10.25*	*16.11*
ENet_SRM	2.00	*11.19*	*7.30*	*13.44*
ENet_d_half	1.72	*12.53*	*8.44*	*13.44*
ENet_rand	2.28	*15.97*	*10.33*	*16.89*
ENet	1.56	*10.39*	*7.67*	*13.39*

L4 to obtain ENet_d. Similarly, we remove the B1 of ENet (the red dotted rectangle in Figure 5.2) and also double the number of feature maps of L3 and L4 to obtain ENet_SRM. For ENet_d_half, we combine the three SRM blocks with the first layer of ENet_d, *i.e.*, concatenating the outputs of these three SRM blocks with the output of first layer of ENet_d to form a 120-channel input of the L2 of ENet_d. In addition, we use the Gaussian random distribution to initialize the first layer of B2 of ENet and make the corresponding ninety kernels trainable, and this replaces the B2's original setting of using fixed SRM filters. The corresponding model is denoted by ENet_rand. Note that, the number of parameters of these four variants are slightly larger than that of ENet. Among these five networks, the ENet achieves better overall performance (see Table 5.2). We find that ENet outperforms ENet_rand (comparing the row of "ENet_rand" and "ENet"). This demonstrates that our ensemble-like design, with different intializations at the first layer of the two branches, can improve the performance of network, especially generalization. It can also be observed that ENet_SRM and ENet_d_half have satisfying performance and that the ENet can further decrease the conventional classification error rate meanwhile slightly improving the overall generalization. Our conjecture is that convolutional layers without pooling operation in the front part of the



(a) R (b) G

(c) B (d) SRM

Figure 5.9: Visualization of FFT of the first-layer filters of ENet. B1: (a)[R], (b)[G] and (c)[B]; B2: (d)[SRM].

two-branch structured ENet tend to preserve the diversity of the information flow, which may be helpful to improve the forensic performance.

To better analyze and understand ENet, we qualitatively analyze the first-layer filters of ENet from the signal processing perspective, *i.e.*, visualizing the FFT (fast Fourier transform) of the first-layer kernels of two branches (B1 and B2 in Figure 5.2), and the corresponding results are shown in Figure 5.9. Different from the apparent high-pass response in (d) for SRM, many kernels in the first layer of B1 capture the band-pass frequency information and several kernels even have relatively low-pass response. This implies that the ENet can automatically capture different frequency band information

with the help of our ensemble-like design. This is beneficial to enrich the extracted features and to improve the detection performance.

5.2.4 Effect of Enhanced Training

In this chapter, we introduced two types of negative sample generation: data-centric method and model-centric method. To verify the effectiveness of these two methods for generalization improvement, we conduct extensive experiments with three CNN models on four datasets. We report in Table 5.3, 5.4, 5.5, and 5.6 the performance before (the column of "NT") and after (the columns of "ET-I" and "ET-G") enhanced training of YaoNet, NcgNet, and ENet, when they are trained on Artlantis, Autodesk, Corona, and VRay, respectively. For each table, starting from the second column, each consecutive three columns form a group, in total four groups. The first group (*e.g.*, "Artlantis" in Table 5.3) is the conventional classification error rate, and the remaining three groups (*e.g.*, "Autodesk", "Corona", and "VRay" in Table 5.3) are the generalization performance.

From Table 5.3-5.6, we find that enhanced training with negative sample insertion based on data-centric and model-centric methods usually leads to slight increase of conventional classification error rate; however, the generalization of the three networks can be apparently and consistently improved by these two methods (except for one case in Table 5.5, when we trained YaoNet on Corona with data-centric method and tested on VRay, with a small increase of HTER by 1.39% from 9.14% to 10.53%, but model-centric method can decrease it to 7.19%). As an example of performance improvement, when we trained ENet on Artlantis with enhanced training based on model-centric method (comparing the columns of "NT" and "ET-G" of the last row in Table 5.3), the conventional classification accuracy decreases by 1.44%, whereas the generalization is improved by 5.75%(Autodesk), 7.30%(Corona), and 6.36%(VRay), respectively. Furthermore, in Table 5.3-5.6, the HTER value of ENet is always the lowest among three networks except for one case (3.08% in Table 5.4, *i.e.*, training on Autodesk and testing on Autodesk with data-centric method). This illustrates our proposed ENet has the superior performance.

We also compare the performance of data-centric and model-centric based enhanced training with that of "mixup" [Zha+18]. "mixup" is a learning principle to regularize the neural network and encourage the trained model to behave linearly in-between training examples. We train the ENet with "mixup" and set its hyperparameter $\alpha = 0.4$ as recommended in [Zha+18]. All the results are reported in Table 5.7. Comparing the columns of "MU" with "ET-I" and "ET-G", we find that data-centric and model-centric based enhanced training significantly outperforms "mixup". Furthermore, the generalization performance of "mixup" sometimes is worse than that of normal training (without "mixup"), *e.g.*, training on Autodesk and testing on Artlantis (10.39% vs. 13.30%). A

Table 5.3: Performance (HTER, in %, lower is better) of three networks when trained on Artlantis. "NT" stands for normal training; "ET-I" and "ET-G" stands for enhanced training with negative samples produced by unpaired linear interpolation and gradient-based distortion, respectively; The generalization performance results on "unknown" rendering engines are in italics (same for Table 5.4, 5.5, and 5.6).

Network	Artlantis			Autodesk			Corona			VRay		
	NT	ET-I	ET-G	NT	ET-I	ET-G	NT	ET-I	ET-G	NT	ET-I	ET-G
YaoNet	3.86	5.17	3.94	*20.31*	*12.61*	*12.64*	*18.17*	*14.14*	*13.72*	*15.11*	*13.42*	*12.47*
NcgNet	3.17	3.47	3.61	*12.69*	*8.47*	*8.17*	*16.00*	*11.72*	*11.22*	*12.58*	*10.78*	*9.58*
ENet	1.31	2.31	2.75	*10.06*	*5.09*	*4.31*	*14.58*	*8.15*	*7.28*	*11.86*	*7.22*	*5.50*

Table 5.4: Performance (HTER, in %, lower is better) of three networks when trained on Autodesk.

Network	Autodesk			Artlantis			Corona			VRay		
	NT	ET-I	ET-G	NT	ET-I	ET-G	NT	ET-I	ET-G	NT	ET-I	ET-G
YaoNet	4.61	3.86	2.72	*28.14*	*22.50*	*16.84*	*15.00*	*11.31*	*8.00*	*21.17*	*16.00*	*12.97*
NcgNet	2.84	2.87	2.67	*16.61*	*11.48*	*11.03*	*12.78*	*9.50*	*9.06*	*16.58*	*12.32*	*12.64*
ENet	1.56	3.08	2.39	*10.39*	*5.64*	*5.58*	*7.67*	*5.39*	*4.86*	*13.39*	*7.47*	*8.22*

Table 5.5: Performance (HTER, in %, lower is better) of three networks when trained on Corona.

Network	Corona			Artlantis			Autodesk			VRay		
	NT	ET-I	ET-G	NT	ET-I	ET-G	NT	ET-I	ET-G	NT	ET-I	ET-G
YaoNet	3.79	3.79	3.28	*19.53*	*17.17*	*11.33*	*10.87*	*10.50*	*8.42*	*9.14*	*10.53*	*7.19*
NcgNet	2.73	3.06	2.81	*21.74*	*15.05*	*13.69*	*9.56*	*7.25*	*7.67*	*8.08*	*6.11*	*5.75*
ENet	1.50	2.72	2.14	*16.08*	*7.61*	*6.39*	*7.92*	*6.83*	*7.03*	*7.78*	*4.36*	*4.39*

Table 5.6: Performance (HTER, in %, lower is better) of three networks when trained on VRay.

Network	VRay			Artlantis			Autodesk			Corona		
	NT	ET-I	ET-G	NT	ET-I	ET-G	NT	ET-I	ET-G	NT	ET-I	ET-G
YaoNet	4.28	4.22	3.64	*16.77*	*15.75*	*11.64*	*15.30*	*11.89*	*9.64*	*9.05*	*7.47*	*6.33*
NcgNet	3.20	3.53	2.84	*14.06*	*10.19*	*8.97*	*15.00*	*8.86*	*8.53*	*5.53*	*5.17*	*5.33*
ENet	1.25	2.22	1.72	*11.58*	*6.94*	*5.39*	*9.97*	*5.97*	*6.08*	*4.53*	*4.31*	*3.17*

possible reason is that "mixup" is essentially a form of data augmentation that implicitly affects the generalization of trained CNN model (in fact, this sometimes cannot guarantee the improvement of generalization as reported in Table 5.7 and mentioned above), whereas data-centric and model-centric based enhanced training can explicitly change the decision boundary and then improve the generalization of CNN-based detectors.

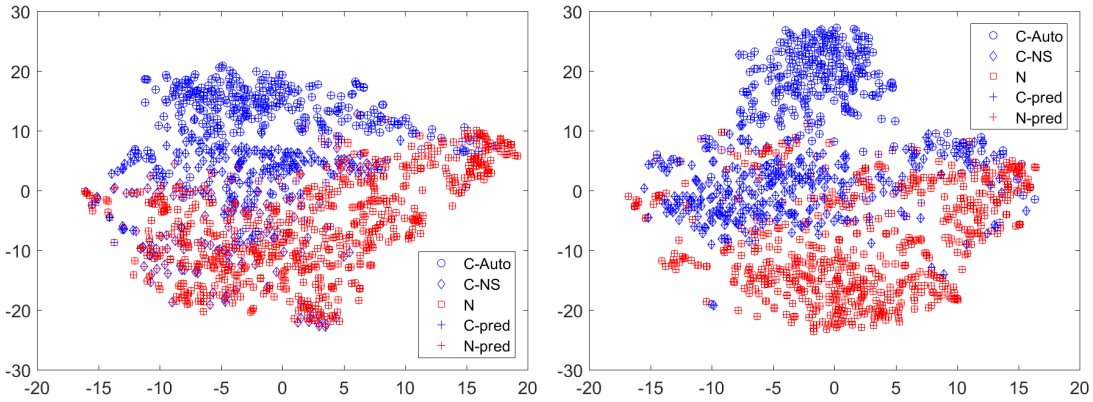
Table 5.7: Performance (HTER, in %, lower is better) of ENet. Each row stands for the training dataset. Starting from the second column, each consecutive four columns form a group, and each group stands for the testing dataset. "NT" stands for normal training; "ET-I" and "ET-G" stand for enhanced training with negative samples produced by unpaired linear interpolation (data-centric) and gradient-based distortion (model-centric), respectively; "MU" stands for "mixup" [Zha+18]. The generalization performance results are in italics.

Dataset	Artlantis				Autodesk				Corona				VRay			
	NT	ET-I	ET-G	MU	NT	ET-I	ET-G	MU	NT	ET-I	ET-G	MU	NT	ET-I	ET-G	MU
Artlantis	1.31	2.31	2.75	1.75	*10.06*	*5.09*	*4.31*	*11.95*	*14.58*	*8.15*	*7.28*	*16.03*	*11.86*	*7.22*	*5.50*	*11.86*
Autodesk	*10.39*	*5.64*	*5.58*	*13.30*	1.56	3.08	2.39	1.70	*7.67*	*5.39*	*4.86*	*9.44*	*13.39*	*7.47*	*8.22*	*14.44*
Corona	*16.08*	*7.61*	*6.39*	*14.67*	*7.92*	*6.83*	*7.03*	*6.97*	1.50	2.72	2.14	1.31	*7.78*	*4.36*	*4.39*	*6.06*
VRay	*11.58*	*6.94*	*5.39*	*9.22*	*9.97*	*5.97*	*6.08*	*9.72*	*4.53*	*4.31*	*3.17*	*4.33*	1.25	2.22	1.72	1.19

As shown above with experimental results, model-centric negative sample insertion usually achieves better performance in terms of conventional classification accuracy and generalization capability, especially for YaoNet, when compared with data-centric method. The reason is that model-centric method can more exactly control the location of negative samples relative to the decision boundary in the feature space of CNN, and thus more effectively improve the generalization with relatively small decrease of the classification accuracy on NIs. To clearly illustrate the location of negative samples generated by data-centric and model-centric methods in the feature space, we train YaoNet on NIs and CG images rendered by Autodesk. We visualize the deep features of negative samples in the last insertion of enhanced training with t-SNE [MH08], and results are shown in Figure 5.10. In Figure 5.10(a), many negative samples are mixed with point cloud of NIs and predicted as NI (blue diamonds with red +) because the linear interpolation is conducted in the image space and "blind" to CNN. On the contrary, all negative samples of model-centric method in Figure 5.10(b) are predicted as CG (blue diamonds with blue +) and almost located in the middle of point clouds of NIs and CG images.

5.2.5 Discussion

In our study, we first observe a new problem regarding the generalization performance of CG forensics, then propose a new method to cope with this challenging problem, and finally validate the proposed method with extensive experiments. New understanding we get from this study is mainly the following: when we roughly know that a class may have a relatively large distribution change during testing, we can use our method to generate proxy samples of the "unknown" distributions by only using available training data; the enhanced training with such samples is effective to improve generalization. This is valid



(a) (b)

Figure 5.10: The deep feature visualization of YaoNet with t-SNE [MH08]. "C" means computer-generated images and "N" means natural images. "C-Auto" means the CG images rendered by Autodesk and "C-NS" means the negative samples generated by data-centric method (a) and model-centric method (b). "Y-pred" means that the predicted label of CNN is Y. NIs and CG images are randomly selected from training dataset for visualization.

for the 12 (4*3) tested cases. Our work is a small step towards the ultimate goal of fully understanding CNN's generalization. We have also tried to gain new understanding with FFT of first-layer filters and t-SNE visualization, which may provide useful insights to colleagues.

5.3 Summary

In this chapter, we studied and proposed a solution for the challenging blind detection problem of CG image forensics. To facilitate this study, we collected four CG datasets with high level of photorealism. We designed and implemented a novel two-branch network with different initializations in their respective first layer to extract more diverse features, and this network has good generalization performance. In the meanwhile, we also introduced the data-centric and model-centric negative sample generation used for conducting enhanced training. This can further improve the generalization performance of CNN-based detectors. More information and materials, including the source code and datasets, are available at https://github.com/weizequan/CGDetection.

For this new and challenging CG forensic problem, our method does not offer a rigorous framework/formulation, which can be considered as a limitation. However, this

might also be an advantage: *e.g.*, we avoid implicit restrictions due to mathematical modeling, such as specific hypothesized parameterization of the distributions of "unknown" CG images. Further, our approach does not use any sample or prior knowledge of new distributions, which is usually necessary for a rigorous formulation (*e.g.*, in many domain adaptation algorithms). This makes our method simple, flexible and generic because our assumption is very weak and everything is done "off-line" at the training side. In the future, we plan to study the generalization improvement with a suitable rigorous formulation. Our proposed method and such a future method are not contradictory, and can even complement each other, *e.g.*, our "off-line" method applied first, before an "on-line" continual learning method. Last but not least, our study implies that in order to improve generalization it is beneficial to learn diverse features and to learn from harder artificial samples. We would like to test this idea in other research problems and in the meanwhile explore other approaches to understanding and enhancing the generalization of CNN-based forensic detectors.

Chapter 6

Conclusions

Contents

6.1 Summary of Contributions

With the rapid development of image editing software and computer graphics rendering technology, as well as recent generation algorithms based on deep learning tools (CNN, GAN, etc.), it has become easier to tamper with image content or generate high-quality images. While these generation technologies facilitate and enrich human daily life, they also weaken the reliability of digital images, and potentially endanger areas that are very sensitive to image data, such as public security, justice, medical care, and education, etc.

For the identification of computer-generated images, researchers have proposed a variety of solutions and made good progress, but there are still many problems to be solved. First of all, at the time when we started this thesis work, most of the existing forensic methods mainly used a two-stage framework of discriminative feature extraction and classifier training. These approaches usually have limited performance on complex data. Inspired by the successful application of advanced deep learning methods in many research fields, recently some researchers have tried to design new deep models and achieved good image forensics performance. However, some questions related to the trustworthiness and interpretation of CNN-based forensic methods are still worth studying. For instance, we may ask the following questions: What information does the CNN extract as discriminative features, and does it capture the "essential" difference between different types of images? In the testing phase, can the discriminative information extracted by CNN generalize well on the "unknown" data, and how to further improve its generalization capability? Finally, computer rendering tools, colorization techniques and other image generation technologies, now more or less using advanced deep learning methods, can generate images with higher and higher visual quality, which undoubtedly increases the requirements of good generalization (or blind detection) capability of forensic detectors. Therefore, evaluating and improving the generalization performance

of image forensic methods is of great research value and practical significance. For the problem of identifying computer-generated images, current research mainly focuses on improving the detection performance of forensic detectors (under an ideal experimental environment). Very few works consider and discuss the generalization of detectors for the "unknown" test data. Under this context, this thesis has studied the above issues in depth and obtained some useful results, which are summarized as follows:

1. A colorized image detection method based on negative sample insertion. Considering that the current forensic methods based on hand-crafted features or CNN have insufficient generalization capability on the "unknown" test data (*i.e.*, the challenging blind detection scenarios), this thesis proposes a CNN enhanced training method to improve the generalization performance of the network. This blind detection performance can be regarded as the generalization capability of forensic detectors. With the aid of the feature visualization, we first analyze the potential reasons for the performance degradation of CNN when used to identify the "unknown" test data, and then design a simple and effective enhancement training method. Specifically, the negative samples (with the same label as colorized images) are automatically constructed through linear interpolation of paired natural image and colorized image (both sharing a same grayscale luminance component), and then these negative samples are iteratively added to the original training data to be used in the so-called enhanced network training. Experimental results show that this enhanced training method can significantly improve the generalization performance of different CNNs.

2. An ensemble-like generalization method for colorized image detection. In the field of image forensics, somehow as expected, CNN-based methods usually achieve the state-of-the-art performance. However, some questions about the trustworthiness of such methods are worth studying and answering, for example, regarding the suitability of the discriminative features automatically extracted by the CNN and the generalization performance of these features on "unknown" test data. Taking colorized image detection as an example, this thesis carries out studies and analysis on the above issues through a series of experiments, and obtains some useful hints, concerning the preparation of experimental data and the use of some existing filters in the beginning of CNN. In the meanwhile, inspired by the experimental results, we propose a very simple method to obtain the final prediction by combining the decisions from CNNs with different settings at the network's first layer. Experiments show that this ensemble-like method can further improve the generalization performance of colorized image detection.

3. CNN-based identification of natural images and CG images. Having observed the limited forensic performance of methods based on hand-crafted features, we have conducted a comprehensive study of CNN-based solutions, which includes CNN fine-tuning, architecture design, loss function selection, visualization and understanding. This thesis

introduces a generic CNN-based framework for identifying computer-rendered images. We start with fine-tuning of a well-known pre-trained CNN from the computer vision community, and then design a new and improved CNN. Both of these CNN-based solutions are superior to the latest methods combining hand-crafted feature extraction and classifier training. Our proposed method shows the best performance on challenging public datasets (very close to the real-world scenarios), and has good robustness against several possible attacks (including resizing and JPEG compression). Last but not least, unlike the existing methods of applying CNN to other image forensic problems, this thesis presents the first attempt to use advanced visualization tools (such as FFT and layer-wise relevance propagation toolbox) to understand what our CNN has learned about the differences between natural images and computer-rendered images. These attempts could provide interesting observations and insights for this CG image detection problem as well as other image forensic tasks.

4. An improved method for identifying CG images based on feature diversity enhancement and adversarial samples. For this part of work, we make efforts to improve the generalization capability of CNN-based detectors from two aspects: network architecture and network training. To our knowledge, we propose and study for the first time in the literature the generalization performance of CG image detection. We first collect four high-quality datasets of computer-rendered images. For the network architecture, we design and implement a novel two-branch CNN. The first layer of the two branches of the network uses different initialization methods to enrich the diversity of deep features. For the network training, we propose a new method based on gradient perturbation to generate more difficult artificial negative samples, and then the enhanced training is carried out to further improve the generalization capability of the CNN-based detector. Our study implies that in order to improve generalization it is beneficial to learn diverse features and from harder artificial samples.

6.2 Perspectives

As summarized above, in this thesis we made some contributions to the research problem of identifying computer-generated images. More efforts shall be devoted to this interesting and important forensic problem. Regarding the future working directions, we have the following suggestions:

1. Increasing the diversity of learned features. Although the deep-learning-based method has achieved good forensic performance, the first layer of most networks uses a set of fixed high-pass filters to extract high-frequency signals. Despite of its effectiveness, this design may suppress some useful forensic traces, and high-pass filters may not

always be the optimal solution. Therefore, free learning or appropriate co-learning with high-pass filters (such as ensemble learning), is an interesting future working direction. In addition, it is promising to design a new deep network architecture to integrate multiple forensic traces, with the main purpose of increasing the diversity of feature learning. Possible attempts include the fusion of the features of the spatial domain with those of the transform domain, extracting useful information from automatically learned domains, and further enhancing the discrimination and generalization capability of features through automatic feature fusion and feature selection.

2. Robustness of the forensic detector. The continuous development of computer generation technology leads to fake images of higher and higher quality. At the same time, a robust forensic detector needs to be able to deal with a certain degree of malicious attacks. Therefore, the anti-forensics of computer-generated images is also worth studying. In addition, the deep network itself also has the risk of being attacked, such as adversarial samples. One possible research idea is to combine anti-forensics in the field of multimedia security and adversarial defenses of deep network to improve the robustness of CNN-based forensic detectors. Moreover, real-world scenarios often have demanding requirements for the robustness of forensic detectors, for example in the case of an adversarial and high-loss image compression. How to retain the forensic performance in such adversarial but realistic scenarios is an important research problem.

3. The detection and quality enhancement of GAN-generated images. Currently, GAN-generated images have become more and more realistic (such as DeepFake, Face2Face, etc), which inevitably bring security risks. Consequently, the detection of GAN-generated images has become a new forensic problem, which has attracted the attention of more and more researchers. In fact, the identification of computer-generated images and the enhancement of the perceived quality of generated images are two sides of the coin, but both are very interesting and related research topics. For example, the detection of GAN-generated images can be used as an analysis method to understand the generation process itself, and accordingly can provide inspiration and guidance for the enhancement of the realism of the generated images. Both research topics can be advanced through several rounds of competitions between them. This is similar to the interplay between steganalysis and steganography, or between cryptanalysis and cryptography.

Appendix A

Résumé en Français

Contents

A.1 Introduction

L'image numérique, de par son caractère direct et compréhensible, en fait un moyen de communication efficace et naturel. Historiquement, l'authenticité des données d'image est fiable. Par exemple, une photo imprimée dans un journal peut être largement acceptée comme preuve d'actualité ; ou, les enregistrements de vidéosurveillance sont proposés comme documents importants au tribunal. Aujourd'hui, grâce au faible coût et à la simplification des appareils d'acquisition, tels que les téléphones intelligents et les appareils photo numériques, presque tout le monde peut enregistrer, stocker et partager un grand nombre d'images/vidéos à tout moment et n'importe où. En attendant, de nombreux logiciels/outils d'édition d'images rendent également extrêmement simple la modification du contenu d'image ou la création de nouvelles images. En conséquence, la possibilité de falsifier le contenu visuel n'est plus limitée aux experts. La technologie numérique a commencé à affaiblir le degré de confiance dans le contenu visuel, et il est évident que « ce que vous voyez n'est plus digne de confiance ». La figure A.1 montre une fausse image très réaliste composée de 16 photos différentes[1]. Avec l'avancée et la complexité des outils de traitement, tous ces problèmes deviennent de plus en plus urgents, ce qui a poussé les recherches sur la criminalistique des images numériques. Le cœur et l'objectif de la criminalistique d'image sont de restaurer une certaine confiance dans les images numériques. D'une manière générale, les principaux objectifs de la criminalistique d'images sont d'analyser une image numérique donnée afin de détecter s'il s'agit d'une falsification, d'identifier son origine, de retracer son historique de traitement, ou de révéler des détails potentiels invisibles à l'œil nu [Fan15].

[1] Cette image provient de http://commons.wikimedia.org/wiki/User:Mmxx, auteur : mxx.

Figure A.1: Une fausse image très réaliste composée de 16 photos différentes. Logiciel utilisé : Adobe Photoshop® [Pho].

Cette thèse étudie principalement l'identification des images générées par ordinateur, y compris la classification des images naturelles (NI) et des images de synthèse (CG) (appelées *CG image forensics*), et la classification des images naturelles et des images colorisées (CI) (appelée détection de CI). Ici, les images naturelles font référence aux images capturées par un appareil photo numérique. Le premier est un problème de recherche important et relativement ancien dans le domaine de la criminalistique des images, et les chercheurs ont déjà mené un grand nombre de travaux [LF05; Ng+05] ; ce dernier est un problème de recherche émergent en criminalistique des images [Guo+18; YRC19]. La figure A.2 montre en (a) deux images CG de haute qualité[2] et en (b) deux images colorisées visuellement réalistes, qui sont obtenues en utilisant un algorithme de colorisation automatique avancé [ISSI16] ; la figure A.2 (c) montre les images naturelles originales correspondantes des images de (b), où (b) et (c) partagent les mêmes informations en niveaux de gris. En fait, il est difficile pour les observateurs humains de déterminer si les images en (a) et (b) ont été capturées par une caméra (c'est-à-dire des images naturelles). Ces questions de la criminalistique des images ont une importance de recherche significative dans les domaines de la sécurité publique, de la justice et du divertissement. Dans le même temps, l'apprentissage profond a récemment atteint un développement rapide grâce à la promotion de l'industrie et à l'attention considérable des chercheurs académiques.

A.2 Objectifs et Contributions

En gardant à l'esprit les problèmes et défis actuels dans le domaine de l'identification d'images générées par ordinateur, l'étude de cette thèse se concentre principalement

[2]Les images proviennent de https://area.autodesk.com/fakeorfoto/.

(a) Images d'infographie

(b) Images colorisées

(c) Images naturelles

Figure A.2: Exemples d'images CG, CI et NI : (a) Images d'infographie (CG) ; (b) Images colorisées (CI) ; (c) Images naturelles (NI). Les images de (b) et (c) partagent les mêmes informations en niveaux de gris.

sur les quatre aspects suivants : (1) Pour la capacité de généralisation de la détection d'images colorisées, nous utilisons la visualisation des caractéristiques pour comprendre les potentielles causes sous-jacentes, et nous introduisons une nouvelle procédure

d'entraînement renforcée basée sur des échantillons négatifs pour améliorer la capacité de généralisation ; (2) Pour la fiabilité des détecteurs criminalistiques basés sur les CNNs (*Convolutional Neural Networks*), nous prenons comme exemple la détection d'images colorisées pour étudier l'impact de la préparation des données et la première couche de CNN sur les performances criminalistiques ; (3) Pour le problème de la criminalistique d'image CG, nous étudions de manière exhaustive les solutions basées sur CNN, y compris la conception de réseau, les stratégies d'entraînement, la visualisation et la compréhension ; (4) Pour améliorer les performances de la détection des images CG, nous combinons les trois premiers travaux de recherche, concevons un nouveau réseau, collectons de nouveaux échantillons de données et améliorons la précision de détection et la capacité de généralisation. Plus précisément, les principaux contenus de recherche et contributions de cette thèse comprennent les quatre points suivants :

1. Détection d'image colorisée basée sur l'insertion d'échantillon négatif. Compte tenu de la capacité de généralisation limitée des détecteurs existants basés sur les caractéristiques fabriquées à la main ou basés sur CNN dans un scénario de détection aveugle difficile, cette thèse propose une méthode d'entraînement de CNN renforcé pour améliorer la capacité de généralisation du détecteur. Ici, la détection aveugle signifie que pendant la phase de test, les échantillons de test sont générés par des méthodes de colorisation « inconnues ». Il s'agit d'une situation fréquemment rencontrée, dans laquelle aucun échantillon des méthodes de colorisation « inconnues » rencontrées lors de la phase de test n'a été utilisé pendant la phase d'entraînement. Cette performance de détection aveugle est également appelée performance de généralisation des détecteurs criminalistiques. Nous analysons d'abord les raisons potentielles de la performance de généralisation limitée des réseaux de neurones au moyen de la visualisation des caractéristiques, puis nous concevons une méthode d'apprentissage renforcé basée sur l'insertion d'échantillons négatifs. Plus précisément, les échantillons négatifs sont automatiquement construits par interpolation linéaire des paires d'images naturelles et d'images colorisées, et ces échantillons ont la même étiquette que les images colorisées. Les échantillons négatifs construits sont ajoutés dans l'ensemble de données d'apprentissage d'origine de manière itérative, puis un entraînement renforcé est effectué, et enfin le modèle est choisi par une méthode simple basée sur des seuils. Cette approche est validée sur plusieurs jeux de données et différents CNNs, et les résultats montrent que l'entraînement renforcé proposé peut considérablement améliorer les performances de généralisation.

2. Une méthode avec une capacité de généralisation améliorée basée sur des études sur l'impact des données et du réseau CNN sur les performances des détecteurs criminalistiques. Récemment, les méthodes d'apprentissage profond ont obtenu de bonnes performances dans de nombreux domaines, et le domaine de la criminalistique d'image ne fait pas exception. De nombreux chercheurs ont introduit des méthodes basées sur CNN

pour résoudre des problèmes criminalistiques, et ces méthodes permettent généralement d'obtenir les meilleures performances criminalistiques. Cependant, des performances élevées peuvent masquer certains problèmes ou pièges potentiels. Par conséquent, cette thèse mène une étude sur la fiabilité des détecteurs criminalistiques basés sur CNN. Plus précisément, nous étudions et répondons à plusieurs questions étroitement liées à la fiabilité du détecteur, telles que l'adéquation des caractéristiques discriminantes extraites automatiquement par le modèle CNN et la capacité de généralisation à des données « inconnues » dans la phase de test. Prenant comme exemple la détection d'images colorisées, cette thèse étudie ces problèmes et obtient des conseils utiles. De plus, inspirés par l'idée de l'apprentissage d'ensemble, nous proposons une méthode simple et efficace pour obtenir les résultats finaux de la prédiction en combinant les résultats de décision des modèles CNN avec des réglages différents à la première couche du réseau. Les résultats expérimentaux montrent que cette méthode peut améliorer les performances de généralisation de la détection d'images colorisées.

3. Une étude approfondie sur l'identification des images naturelles et des images CG basée sur CNN. Motivée par l'observation des performances de classification limitées des méthodes traditionnelles basées sur des caractéristiques fabriquées à la main, en particulier lorsqu'il s'agit d'ensembles de données multi-sources plus complexes, cette thèse conçoit et met en œuvre un cadre d'identification générique, qui contient trois groupes de réseaux pour traiter les blocs d'images d'entrée de différentes tailles. Nous effectuons d'abord le réglage fin du modèle CNN pré-entraîné sur ImageNet, puis concevons un réseau CNN amélioré avec des couches convolutives en cascade pour ce problème criminalistique. Les résultats expérimentaux montrent que les deux solutions basées sur CNN sont supérieures aux méthodes de pointe basées sur l'extraction de caractéristiques fabriquées à la main et l'entraînement de classificateurs. Plus important encore, notre méthode montre une bonne capacité de classification sur un ensemble de données public difficile comprenant des images d'origines hétérogènes (très proches de l'application du monde réel), et démontre une forte robustesse contre plusieurs opérations de post-traitement, y compris le redimensionnement et la compression JPEG. Notre travail a été l'une des premières méthodes basées sur l'apprentissage profond pour détecter les images CG. De plus, contrairement aux méthodes existantes d'application de CNN aux problèmes de criminalistique d'image, nous utilisons des outils avancés d'analyse et de visualisation, y compris la transformation de Fourier rapide (FFT), la propagation de pertinence par couche (LRP) et les « entrées préférées » (PI), pour comprendre ce que notre CNN a appris sur les différences entre les images NI et CG.

4. Identification d'images CG basée sur l'amélioration de la diversité des caractéristiques et des exemples contradictoires. Les performances criminalistiques des détecteurs CNN existants peuvent être encore améliorées, en particulier la capacité de généralisation

sur des ensembles de données de test « inconnues » est limitée. Pour résoudre ce problème, nous faisons des efforts dans deux aspects de CNN : la conception de l'architecture du réseau et l'entraînement du réseau. Une autre contribution est que pour la première fois dans la littérature, nous proposons d'étudier la capacité de généralisation des méthodes criminalistiques d'images CG. Afin d'étudier ce problème difficile, nous collectons quatre jeux de données d'images CG de haute qualité. Pour l'architecture du réseau, nous concevons un CNN à deux branches. La première couche des deux branches du réseau utilise différentes méthodes d'initialisation, à savoir l'initialisation aléatoire gaussienne et un ensemble de filtres résiduels de passe-haut. Le but est d'enrichir la diversité des caractéristiques extraites. Pour l'entraînement du réseau, nous proposons une nouvelle méthode centrée sur le modèle CNN pour générer des exemples contradictoires comme des échantillons négatifs plus difficiles (en comparaison avec la méthode dite centrée sur les données, c'est-à-dire identique à la génération d'échantillons négatifs pour la détection d'images colorisées présentée plus haut, qui est basée sur l'interpolation d'une paire de NI et CI). Ensuite, un entraînement renforcé est effectué pour améliorer encore la capacité de généralisation de CNN qui utilise les échantillons négatifs générés. Les résultats expérimentaux sur plusieurs jeux de données montrent que notre méthode proposée peut obtenir une meilleure précision de classification et une meilleure performance de généralisation.

En résumé, cette thèse considère principalement quatre tâches de recherche. Les deux premières tâches se concentrent sur la détection d'images colorisées. Le premier résout le problème de capacité de généralisation, et le second étudie l'impact des données et de l'architecture du réseau sur les performances criminalistiques (en particulier les performances de généralisation). Les deux dernières tâches se concentrent sur l'identification des images CG. Pour ce problème criminalistique, nous proposons d'abord un cadre générique basé sur CNN et effectuons une analyse et une compréhension du réseau en utilisant des outils avancés de visualisation. Sur la base de toutes nos études précédentes, nous améliorons ensuite la conception de l'architecture du réseau et la génération d'échantillons négatifs pour atteindre une capacité de généralisation améliorée de l'identification d'images CG.

A.3 Perspectives

Comme résumé ci-dessus, dans cette thèse, nous avons apporté quelques contributions au problème de recherche de l'identification des images générées par ordinateur, plus précisément les images d'infographie (CG) et les images colorisées (CI). Des efforts supplémentaires seront consacrés à ce problème criminalistique intéressant et important. Concernant les futures orientations de travail, nous avons les suggestions suivantes :

1. Augmenter la diversité des caractéristiques apprises. Bien que la méthode basée sur l'apprentissage profond ait obtenu de bonnes performances criminalistiques, la première couche de la plupart des réseaux utilise un ensemble de filtres passe-haut fixes pour extraire les signaux de haute fréquence. Malgré son efficacité, cette conception peut supprimer certaines traces criminalistiques utiles, et les filtres passe-haut peuvent ne pas toujours être la solution optimale. Par conséquent, l'apprentissage libre ou le co-apprentissage approprié avec des filtres passe-haut (tels que l'apprentissage d'ensemble), est une direction de travail future intéressante. En outre, il est prometteur de concevoir une nouvelle architecture de réseau profond pour intégrer plusieurs traces criminalistiques, dans le but principal d'augmenter la diversité de l'apprentissage des caractéristiques. Les tentatives possibles incluent la fusion des caractéristiques du domaine spatial avec celles du domaine de transformation, l'extraction d'informations utiles à partir de domaines appris automatiquement, et l'amélioration de la capacité de discrimination et de généralisation des caractéristiques par une fusion automatique et la sélection de caractéristiques.

2. Robustesse du détecteur criminalistique. Le développement continu de la technologie de génération d'images conduit à de fausses images de meilleure qualité. Dans le même temps, un détecteur criminalistique robuste doit être capable de gérer un certain degré d'attaques malveillantes. Par conséquent, l'anti-criminalistique des images générées par ordinateur mérite également d'être étudiée. De plus, le réseau profond lui-même risque également d'être attaqué, comme des échantillons contradictoires. Une idée de recherche possible est de combiner l'anti-criminalistique dans le domaine de la sécurité multimédia et les défenses antagonistes des réseaux profonds pour améliorer la robustesse des détecteurs criminalistiques basés sur CNN. De plus, les scénarios du monde réel ont souvent des exigences élevées en matière de robustesse des détecteurs criminalistiques, par exemple dans le cas d'une image contradictoire compressée à perte élevée. Comment conserver la performance criminalistique dans de tels scénarios contradictoires mais réalistes est un problème de recherche important.

3. La détection et l'amélioration de la qualité des images générées par GAN (*Generative Adversarial Network*). Actuellement, les images générées par GAN sont devenues de plus en plus réalistes (comme DeepFake, Face2Face, etc.), ce qui entraîne inévitablement des risques de sécurité. Par conséquent, la détection des images générées par le GAN est devenue un nouveau problème criminalistique, qui a attiré l'attention de plus en plus de chercheurs. En fait, l'identification des images générées par ordinateur et l'amélioration de la qualité perçue des images générées sont les deux faces d'une pièce de monnaie, mais les deux sont des sujets de recherche très intéressants et connexes. Par exemple, la détection d'images générées par GAN peut être utilisée comme une méthode d'analyse pour comprendre le processus de génération lui-même, et en conséquence peut fournir

une inspiration et des conseils pour l'amélioration du réalisme des images générées. Les deux sujets de recherche peuvent être avancés à travers plusieurs séries de concours entre eux. Ceci est similaire à l'interaction entre la stéganalyse et la stéganographie, ou entre la cryptanalyse et la cryptographie.

Bibliography

[AM18] N. Akhtar and A. Mian. "Threat of adversarial attacks on deep learning in computer vision: A survey". In: *IEEE Access* 6 (2018), pp. 14410–14430 (Cited on page 77).

[Art] *Artlantis gallery*. https://atl.artlantis.com/en/gallery/. (visited on 2020-4-1) (Cited on pages 74, 84).

[Aut] *Autodesk A360 rendering gallery*. https://gallery.autodesk.com/a360rendering/. (visited on 2020-4-1) (Cited on pages 74, 84).

[Bac+15] S. Bach et al. "On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation". In: *PLoS ONE* 10.7 (2015), e0130140:1–46 (Cited on page 69).

[Bap+19] J. H. Bappy et al. "Hybrid LSTM and Encoder–Decoder Architecture for Detection of Image Forgeries". In: *IEEE Transactions on Image Processing* 28.7 (2019), pp. 3286–3300 (Cited on page 3).

[BCV13] Y. Bengio, A. Courville, and P. Vincent. "Representation Learning: A Review and New Perspectives". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (2013), pp. 1798–1828 (Cited on page 7).

[BF04] P. Blythe and J. Fridrich. "Secure Digital Camera". In: *Proceedings of the Digital Forensic Research Workshop*. 2004, 11–13 (Cited on page 2).

[Bon+17] L. Bondi et al. "First steps toward camera model identification with convolutional neural networks". In: *IEEE Signal Processing Letters* 24.3 (2017), pp. 259–263 (Cited on pages 3, 37).

[BS18] B. Bayar and M. C. Stamm. "Constrained Convolutional Neural Networks: A New Approach Towards General Purpose Image Manipulation Detection". In: *IEEE Transactions on Information Forensics and Security* 13.11 (2018), pp. 2691–2706 (Cited on pages 17, 48, 52, 57, 67, 68).

[BT+19] D. Bhalang Tarianga et al. "Classification of Computer Generated and Natural Images based on Efficient Deep Convolutional Recurrent Attention Model". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop*. 2019, pp. 146–152 (Cited on pages 15, 16).

[BZ17] C. Barnes and F.-L. Zhang. "A survey of the state-of-the-art in patch-based synthesis". In: *Computational Visual Media* 3.1 (2017), pp. 3–20 (Cited on page 70).

[Cao+14] G. Cao et al. "Contrast Enhancement-Based Forensics in Digital Images". In: *IEEE Transactions on Information Forensics and Security* 9.3 (2014), pp. 515–525 (Cited on page 3).

[Cao+15] X. Cao et al. "Image composite authentication using a single shadow observation". In: *Science China Information Sciences* 58 (2015), pp. 1–13 (Cited on page 3).

[Che+09] D. Chen et al. "Identifying computer generated and digital camera images using fractional lower order moments". In: *Proceedings of the IEEE Conference on Industrial Electronics and Applications*. 2009, pp. 230–235 (Cited on pages 12–14).

[Che+12] X. Chen et al. "Manifold Preserving Edit Propagation". In: *ACM Transactions on Graphics* 31.6 (2012), 132:1–132:7 (Cited on page 7).

[Che+15] J. Chen et al. "Median filtering forensics based on convolutional neural networks". In: *IEEE Signal Processing Letters* 22.11 (2015), pp. 1849–1853 (Cited on pages 3, 17, 48, 52).

[Chr+12] V. Christlein et al. "An Evaluation of Popular Copy-Move Forgery Detection Approaches". In: *IEEE Transactions on Information Forensics and Security* 7.6 (2012), pp. 1841–1854 (Cited on page 3).

[CNH13] C. Chen, J. Ni, and J. Huang. "Blind Detection of Median Filtering in Digital Images: A Difference Domain Based Approach". In: *IEEE Transactions on Image Processing* 22.12 (2013), pp. 4699–4710 (Cited on page 3).

[Cog18] R. Cogranne. "Determining JPEG image standard quality factor from the quantization tables". In: *CoRR* abs/1802.00992 (2018), pp. 1–6 (Cited on page 39).

[Con11] V. Conotter. "Active and passive multimedia forensics". In: *PhD thesis. University of Trento* (2011) (Cited on page 2).

[Con+14] V. Conotter et al. "Physiologically-based detection of computer generated faces in video". In: *Proceedings of the IEEE International Conference on Image Processing*. 2014, pp. 248–252 (Cited on page 15).

[Cor] *Corona renderer gallery*. `https://corona-renderer.com/gallery`. (visited on 2020-4-1) (Cited on pages 74, 78, 84).

[CSX07] W. Chen, Y. Q. Shi, and G. Xuan. "Identifying Computer Graphics using HSV Color Model and Statistical Moments of Characteristic Functions". In: *Proceedings of the IEEE International Conference on Multimedia and Expo*. 2007, pp. 1123–1126 (Cited on pages 12–14, 54).

[CYS15] Z. Cheng, Q. Yang, and B. Sheng. "Deep Colorization". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 415–423 (Cited on page 9).

[Den+09] J. Deng et al. "ImageNet: A large-scale hierarchical image database". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255 (Cited on pages 9, 33, 34, 39, 50, 79).

[DN+15] D.-T. Dang-Nguyen et al. "RAISE: A Raw Images Dataset for Digital Image Forensics". In: *Proceedings of the ACM Multimedia Systems Conference*. 2015, pp. 219–224 (Cited on pages 62, 84).

[DNBDN12a] D.-T. Dang-Nguyen, G. Boato, and F. G. B. De Natale. "Discrimination between computer generated and natural human faces based on asymmetry information". In: *Proceedings of the European Signal Processing Conference*. 2012, pp. 1234–1238 (Cited on page 15).

[DNBDN12b] D.-T. Dang-Nguyen, G. Boato, and F. G. B. De Natale. "Identify computer generated characters by analysing facial expressions variation". In: *Proceedings of the IEEE International Workshop on Information Forensics and Security*. 2012, pp. 252–257 (Cited on page 15).

[DNBDN15] D.-T. Dang-Nguyen, G. Boato, and F. G. B. De Natale. "3D-Model-Based Video Analysis for Computer Generated Faces Identification". In: *IEEE Transactions on Information Forensics and Security* 10.8 (Aug. 2015), pp. 1752–1763 (Cited on pages 15, 70).

[Fan+12a] S. Fan et al. "Classifying computer generated graphics and natural images based on image contour information". In: *Journal of Information and Computational Science* 9.10 (2012), pp. 2877–2895 (Cited on pages 12–14).

[Fan+12b] S. Fan et al. "Real or Fake?: Human Judgments About Photographs and Computer-generated Images of Faces". In: *Proceedings of the SIGGRAPH Asia Technical Briefs*. 2012, 17:1–17:4 (Cited on page 12).

[Fan15] W. Fan. "Towards Digital Image Anti-Forensics via Image Restoration". In: *PhD thesis. University of Grenoble Alpes* (2015) (Cited on pages 2, 99).

[Far09] H. Farid. "A survey of image forgery detection". In: *IEEE Signal Processing Magazine* 26.2 (2009), pp. 16–25 (Cited on page 2).

[FB07] H. Farid and M. J. Bravo. "Photorealistic rendering: How realistic is it?" In: *Journal of Vision* 7.9 (2007), p. 766 (Cited on page 11).

[FB12] H. Farid and M. J. Bravo. "Perceptual Discrimination of Computer Generated and Photographic Faces". In: *Digital Investigation* 8.3 (Feb. 2012), pp. 226–235 (Cited on page 11).

[FCD12] X. Feng, I. J. Cox, and G. Doerr. "Normalized Energy Density-Based Forensic Detection of Resampled Images". In: *IEEE Transactions on Multimedia* 14.3 (2012), pp. 536–545 (Cited on page 3).

[FK12] J. Fridrich and J. Kodovsky. "Rich models for steganalysis of digital images". In: *IEEE Transactions on Information Forensics and Security* 7.3 (2012), pp. 868–882 (Cited on pages 24, 25, 38, 40, 52, 76, 77).

[FL03] H. Farid and S. Lyu. "Higher-order Wavelet Statistics and their Application to Digital Forensics". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop*. 2003 (Cited on pages 3, 14).

[FQ03] Z. Fan and R. L. de Queiroz. "Identification of bitmap compression history: JPEG detection and quantizer estimation". In: *IEEE Transactions on Image Processing* 12.2 (2003), pp. 230–235 (Cited on pages 3, 39).

[Fri93] G. L. Friedman. "The trustworthy digital camera: restoring credibility to the photographic image". In: *IEEE Transactions on Consumer Electronics* 39.4 (1993), pp. 905–910 (Cited on page 2).

[GC08] A. C. Gallagher and T. Chen. "Image authentication by detecting traces of demosaicing". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2008, pp. 1–8 (Cited on pages 12, 54).

[GFC14] M. Goljan, J. Fridrich, and R. Cogranne. "Rich model for Steganalysis of color images". In: *Proceedings of the IEEE International Workshop on Information Forensics and Security*. 2014, pp. 185–190 (Cited on page 61).

[Goo+14] I. Goodfellow et al. "Generative Adversarial Nets". In: *Proceedings of the Advances in Neural Information Processing Systems*. 2014, pp. 2672–2680 (Cited on page 3).

[GSS15] I. Goodfellow, J. Shlens, and C. Szegedy. "Explaining and Harnessing Adversarial Examples". In: *Proceedings of the International Conference on Learning Representations*. 2015 (Cited on pages 79, 80).

[Guo+18] Y. Guo et al. "Fake colorized image detection". In: *IEEE Transactions on Information Forensics and Security* 13.8 (2018), pp. 1932–1944 (Cited on pages 3, 9, 10, 26, 32, 33, 38–41, 100).

[Gup+12] R. K. Gupta et al. "Image Colorization Using Similar Images". In: *Proceedings of the ACM International Conference on Multimedia*. 2012, pp. 369–378 (Cited on pages 7, 8).

[Hah+00] R. H. R. Hahnloser et al. "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit." In: *Nature* 405 (Dec. 2000), pp. 947–951 (Cited on pages 25, 50).

[HBF16] O. Holmes, M. S. Banks, and H. Farid. "Assessing and Improving the Identification of Computer-Generated Portraits". In: *ACM Transactions on Applied Perception* 13.2 (Feb. 2016), 7:1–7:12 (Cited on page 12).

[He+16] K. He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 770–778 (Cited on page 51).

[He+18a] M. He et al. "Deep exemplar-based colorization". In: *ACM Transactions on Graphics* 37.4 (2018), 47:1–47:16 (Cited on pages 7, 8).

[He+18b] P. He et al. "Computer Graphics Identification Combining Convolutional and Recurrent Neural Networks". In: *IEEE Signal Processing Letters* 25.9 (2018), pp. 1369–1373 (Cited on pages 15, 16).

[HHS10] F. Huang, J. Huang, and Y. Q. Shi. "Detecting Double JPEG Compression With the Same Quantization Matrix". In: *IEEE Transactions on Information Forensics and Security* 5.4 (2010), pp. 848–856 (Cited on page 3).

[HST11] K. He, J. Sun, and X. Tang. "Single Image Haze Removal Using Dark Channel Prior". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.12 (2011), pp. 2341–2353 (Cited on page 10).

[ICOL05] R. Irony, D. Cohen-Or, and D. Lischinski. "Colorization by Example". In: *Proceedings of the Eurographics Conference on Rendering Techniques*. 2005, pp. 201–210 (Cited on pages 7, 8).

[IS15] S. Ioffe and C. Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *Proceedings of the International Conference on Machine Learning*. 2015, pp. 448–456 (Cited on page 52).

[ISSI16] S. Iizuka, E. Simo-Serra, and H. Ishikawa. "Let there be color!: Joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification". In: *ACM Transactions on Graphics* 35.4 (2016), pp. 1–11 (Cited on pages 3, 7, 9, 10, 26, 31, 32, 39, 100).

[IVR03] T. I. Ianeva, A. P. de Vries, and H. Rohrig. "Detecting cartoons: A case study in automatic video-genre classification". In: *Proceedings of the IEEE International Conference on Multimedia and Expo*. Vol. 1. 2003, pp. 449–452 (Cited on page 13).

[Jia13] Y. Jia. *Caffe: An open source convolutional architecture for fast feature embedding.* (visited on 2018-01-15). 2013 (Cited on page 54).

[Jia+14] Y. Jia et al. "Caffe: Convolutional Architecture for Fast Feature Embedding". In: *Proceedings of the ACM International Conference on Multimedia.* 2014, pp. 675–678 (Cited on pages 50, 57, 68).

[KF10] M. Kirchner and J. Fridrich. "On detection of median filtering in digital images". In: *Proceedings of the Media Forensics and Security II.* Vol. 7541. 2010, pp. 371 –382 (Cited on page 3).

[KGB16] A. Kurakin, I. J. Goodfellow, and S. Bengio. "Adversarial examples in the physical world". In: *CoRR* abs/1607.02533 (2016) (Cited on pages 79, 80).

[Kha+08] N. Khanna et al. "Forensic techniques for classifying scanner, computer generated and digital camera images". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing.* 2008, pp. 1653–1656 (Cited on page 54).

[KSH12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Proceedings of the Advances in Neural Information Processing Systems.* 2012, pp. 1097–1105 (Cited on pages 49, 51, 87).

[Lap+16] S. Lapuschkin et al. "The LRP Toolbox for Artificial Neural Networks". In: *Journal of Machine Learning Research* 17.114 (2016), pp. 1–5 (Cited on pages 68, 69).

[LF05] S. Lyu and H. Farid. "How realistic is photorealistic?" In: *IEEE Transactions on Signal Processing* 53.2 (Feb. 2005), pp. 845–850 (Cited on pages 3, 11–14, 54, 100).

[LFG06] J. Lukas, J. Fridrich, and M. Goljan. "Digital Camera Identification from Sensor Pattern Noise". In: *IEEE Transactions on Information Forensics and Security* 1.2 (2006), 205–214 (Cited on page 3).

[Li+15] J. Li et al. "Segmentation-Based Image Copy-Move Forgery Detection Scheme". In: *IEEE Transactions on Information Forensics and Security* 10.3 (2015), pp. 507–518 (Cited on page 3).

[Li+18] H. Li et al. "Can Forensic Detectors Identify GAN Generated Images?" In: *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference.* 2018, pp. 722–727 (Cited on page 3).

[Li+19] Y. Li et al. "Using Neural Networks for Fake Colorized Image Detection". In: *Proceedings of Advances in Digital Forensics XV*. 2019, pp. 201–215 (Cited on pages 9, 10).

[Liu+11] Q. Liu et al. "Identifying Image Composites Through Shadow Matte Consistency". In: *IEEE Transactions on Information Forensics and Security* 6.3 (2011), pp. 1111–1122 (Cited on page 3).

[LJY17] F.-F. Li, J. Johnson, and S. Yeung. *Visualizing What ConvNets Learn*. (Course notes of Stanford University, visited on 2019-10-25). 2017 (Cited on page 67).

[LLW04] A. Levin, D. Lischinski, and Y. Weiss. "Colorization Using Optimization". In: *ACM Transactions on Graphics* 23.3 (2004), pp. 689–694 (Cited on pages 7, 8).

[LMS16] G. Larsson, M. Maire, and G. Shakhnarovich. "Learning representations for automatic colorization". In: *Proceedings of the European Conference on Computer Vision*. 2016, pp. 577–593 (Cited on pages 7, 9, 10, 26, 31–33, 36, 39).

[Lua+07] Q. Luan et al. "Natural Image Colorization". In: *Proceedings of the Eurographics Conference on Rendering Techniques*. 2007, pp. 309–320 (Cited on page 7).

[LUO+07] W. LUO et al. "A survey of passive technology for digital image forensics". In: *Frontiers of Computer Science in China* 2.1 (2007), 166–179 (Cited on page 2).

[LYS13] Z. Li, J. Ye, and Y. Q. Shi. "Distinguishing computer graphics from photographic images using local binary patterns". In: *Proceedings of the International Workshop on Digital-Forensics and Watermarking*. 2013, pp. 228–241 (Cited on pages 12, 13, 54).

[LZ19] Y. Li and J. Zhou. "Fast and Effective Image Copy-Move Forgery Detection via Hierarchical Feature Point Matching". In: *IEEE Transactions on Information Forensics and Security* 14.5 (2019), pp. 1307–1322 (Cited on page 3).

[MBF17] B. Mader, M. S. Banks, and H. Farid. "Identifying Computer-Generated Portraits: The Importance of Training and Incentives". In: *Perception* 46.9 (Sept. 2017), pp. 1062–1076 (Cited on page 12).

[MH08] L. van der Maaten and G. E. Hinton. "Visualizing High-Dimensional Data Using t-SNE". In: *Journal of Machine Learning Research* 9 (2008), pp. 2579–2605 (Cited on pages 26, 27, 35, 92, 93).

[MS20] O. Mayer and M. C. Stamm. "Forensic Similarity for Digital Images". In: *IEEE Transactions on Information Forensics and Security* 15 (2020), pp. 1331–1346 (Cited on page 3).

[NC13] T.-T. Ng and S.-F. Chang. "Discrimination of Computer Synthesized or Recaptured Images from Real Images". In: *Digital Image Forensics: There is More to a Picture than Meets the Eye*. 2013, pp. 275–309 (Cited on page 12).

[Ng+04] T.-T Ng et al. *Columbia Photographic Images and Photorealistic Computer Graphics Dataset*. Tech. rep. 205-2004-5. ADVENT, Columbia University, 2004 (Cited on pages 12, 48, 54).

[Ng+05] T.-T. Ng et al. "Physics-motivated features for distinguishing photographic images and computer graphics". In: *Proceedings of the ACM International Conference on Multimedia*. 2005, pp. 239–248 (Cited on pages 3, 12, 13, 54–56, 58, 61, 100).

[NH10] V. Nair and G. E. Hinton. "Rectified Linear Units Improve Restricted Boltzmann Machines". In: *Proceedings of the International Conference on Machine Learning*. 2010, pp. 807–814 (Cited on pages 25, 50).

[Niu+19] Y. Niu et al. "An enhanced approach for detecting double JPEG compression with the same quantization matrix". In: *Signal Processing: Image Communication* 76 (2019), pp. 89 –96 (Cited on page 3).

[NYE19] H. H. Nguyen, J. Yamagishi, and I. Echizen. "Capsule-forensics: Using Capsule Networks to Detect Forged Images and Videos". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. 2019, pp. 2307–2311 (Cited on pages 15, 16).

[OA11] L. Özparlak and I. Avcibas. "Differentiating Between Images Using Wavelet-Based Transforms: A Comparative Study". In: *IEEE Transactions on Information Forensics and Security* 6.4 (2011), pp. 1418–1431 (Cited on pages 12–14).

[Pan+13] J. Pang et al. "Image colorization using sparse representation". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. 2013, pp. 1578–1582 (Cited on page 7).

[PBF10] T. Pevny, P. Bas, and J. Fridrich. "Steganalysis by Subtractive Pixel Adjacency Matrix". In: *IEEE Transactions on Information Forensics and Security* 5.2 (2010), pp. 215–224 (Cited on pages 52, 55, 56, 58).

[PCH09] F. PAN, J. CHEN, and J. HUANG. "Discriminating between photorealistic computer graphics and natural images using fractal geometry". In: *Science in China Series F: Information Sciences* 52.2 (2009), pp. 329–337 (Cited on pages 12, 13).

[Pen+17] F. Peng et al. "Discrimination of natural images and computer generated graphics based on multi-fractal and regression analysis". In: *AEU - International Journal of Electronics and Communications* 71 (Jan. 2017), pp. 72–81 (Cited on pages 12–14, 54, 55, 58).

[PF05] A. C. Popescu and H. Farid. "Exposing digital forgeries by detecting traces of resampling". In: *IEEE Transactions on Signal Processing* 53.2 (2005), pp. 758–767 (Cited on pages 3, 13).

[PF08] T. Pevny and J. Fridrich. "Detection of Double-Compression in JPEG Images for Applications in Steganography". In: *IEEE Transactions on Information Forensics and Security* 3.2 (2008), pp. 247–258 (Cited on page 3).

[Pho] *The best photo editing software for spectacular photos and graphics.* `https://www.adobe.com/products/photoshopfamily.html`. (visited on 2020-4-1) (Cited on pages 2, 100).

[Pia17] M. Piaskiewicz. *Level-disign reference database.* (visited on 2019-11-1). 2017 (Cited on page 62).

[Piv13] A. Piva. "An overview on image forensics". In: *ISRN Signal Processing* 2013 (2013), pp. 1–22 (Cited on page 2).

[PLL14] F. Peng, J. Li, and M. Long. "Identification of natural images and computer-generated graphics based on statistical and textural features". In: *Journal of Forensic Sciences* 60.2 (2014), pp. 435–443 (Cited on pages 12, 13).

[Pyt] *PyTorch.* `https://pytorch.org/`. (visited on 2020-4-1) (Cited on page 41).

[PZ14] F. Peng and D.-l. Zhou. "Discriminating natural images and computer generated graphics based on the impact of CFA interpolation on the correlation of PRNU". In: *Digital Investigation* 11.2 (2014), pp. 111–119 (Cited on pages 12, 13).

[Qia+15] Y. Qian et al. "Deep learning for steganalysis via convolutional neural networks". In: *Proceedings of the IS&T/SPIE Electronic Imaging.* Vol. 9409. 2015, 94090J1–94090J10 (Cited on page 52).

[Qua+16] W. Quan et al. "Maximal Poisson-disk Sampling via Sampling Radius Optimization". In: *Proceedings of the SIGGRAPH Asia Posters*. 2016, 22:1–22:2 (Cited on pages 50, 54).

[Qua+18] W. Quan et al. "Distinguishing between natural and computer-generated images using convolutional neural networks". In: *IEEE Transactions on Information Forensics and Security* 13.11 (2018), pp. 2772–2787 (Cited on pages 37, 48, 76, 77, 86, 87).

[Qua+19a] W. Quan et al. "Impact of Data Preparation and CNN's First Layer on Performance of Image Forensics: A Case Study of Detecting Colorized Images". In: *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence - Volume 24800*. 2019, pp. 127–131 (Cited on pages 38, 75, 76, 88).

[Qua+19b] W. Quan et al. "Improving the Generalization of Colorized Image Detection with Enhanced Training of CNN". In: *Proceedings of the International Symposium on Image and Signal Processing and Analysis*. 2019, pp. 246–252 (Cited on pages 24, 75–77).

[Qua+20] W. Quan et al. "Learn with diversity and from harder samples: Improving the generalization of CNN-based detection of computer-generated images". In: *Forensic Science International: Digital Investigation* 35 (2020), p. 301023 (Cited on page 75).

[Rah17] N. Rahmouni. *CGvsPhoto*. https://github.com/NicoRahm/CGvsPhoto/. (visited on 2020-10-18). 2017 (Cited on page 62).

[Rah+17] N. Rahmouni et al. "Distinguishing Computer Graphics from Natural Images Using Convolution Neural Networks". In: *Proceedings of the IEEE International Workshop on Information Forensics and Security*. 2017, pp. 1–6 (Cited on pages 15, 62).

[RL14] S.-J. Ryu and H.-K. Lee. "Estimation of Linear Transformation by Analyzing the Periodicity of Interpolation". In: *Pattern Recognition Letters* 36 (2014), 89–99 (Cited on page 3).

[RN16] Y. Rao and J. Ni. "A deep learning approach to detection of splicing and copy-move forgeries in images". In: *Proceedings of the IEEE International Workshop on Information Forensics and Security*. 2016, pp. 1–6 (Cited on page 3).

[Sal03] P. Sallee. *Matlab JPEG Toolbox*. https://github.com/MKLab-ITI/image-forensics/tree/master/matlab_toolbox/Util/jpegtbx_1.4. (visited on 2020-10-18). 2003 (Cited on page 39).

[SFH17] S. Sabour, N. Frosst, and G. E. Hinton. "Dynamic Routing Between Capsules". In: *Proceedings of the Advances in Neural Information Processing Systems*. 2017, pp. 3856–3866 (Cited on page 16).

[Shu+16] B. Shuai et al. "DAG-Recurrent Neural Networks for Scene Labeling". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 3620–3629 (Cited on page 16).

[Shu+17] D. Shullani et al. "VISION: a video and image dataset for source identification". In: *EURASIP Journal on Information Security* 2017.1 (2017), p. 15 (Cited on pages 84, 85).

[Sri+14] N. Srivastava et al. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15.1 (Jan. 2014), pp. 1929–1958 (Cited on page 52).

[Suy+02] J. A. K. Suykens et al. *Least Squares Support Vector Machines*. Singapore: World Scientific, 2002 (Cited on page 56).

[SZ14] K. Simonyan and A. Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *Proceedings of the International Conference on Learning Representations*. 2014, pp. 1–14 (Cited on pages 16, 49).

[Sze+14] C. Szegedy et al. "Intriguing properties of neural networks". In: *Proceedings of the International Conference on Learning Representations*. 2014 (Cited on page 77).

[Sze+15] C. Szegedy et al. "Going deeper with convolutions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 1–9 (Cited on page 79).

[SZY09] G. Sankar, V. Zhao, and Y. H. Yang. "Feature based classification of computer graphics and real images". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. 2009, pp. 1513–1516 (Cited on pages 12, 13).

[TCC16] A. Tuama, F. Comby, and M. Chaumont. "Camera model identification with the use of deep convolutional neural networks". In: *Proceedings of the IEEE International Workshop on Information Forensics and Security*. 2016, pp. 1–6 (Cited on page 3).

[Ton18] B. Tondi. "Pixel-domain adversarial examples against CNN-based manipulation detectors". In: *Electronics Letters* 54.21 (2018), pp. 1220–1222 (Cited on page 80).

[Tra+18] F. Tramèr et al. "Ensemble Adversarial Training: Attacks and Defenses". In: *Proceedings of the International Conference on Learning Representations*. 2018 (Cited on page 79).

[Vra] *Chaosgroup gallery*. https://www.chaosgroup.com/gallery. (visited on 2020-4-1) (Cited on pages 74, 84).

[WAM02] T. Welsh, M. Ashikhmin, and K. Mueller. "Transferring Color to Greyscale Images". In: *ACM Transactions on Graphics* 21.3 (2002), pp. 277–280 (Cited on pages 7, 8).

[Wan+14] X. Wang et al. "A statistical feature based approach to distinguish PRCG from photographs". In: *Computer Vision and Image Understanding* 128 (2014), pp. 84 –93 (Cited on pages 12, 13).

[Wan+17] J. Wang et al. "Forensics Feature Analysis in Quaternion Wavelet Domain for Distinguishing Photographic Images and Computer Graphics". In: *Multimedia Tools and Applications* 76.22 (2017), 23721–23737 (Cited on pages 12–14).

[Wan17] K. Wang. "A simple and effective image-statistics-based approach to detecting recaptured images from LCD screens". In: *Digital Investigation* 23 (2017), pp. 75 –87 (Cited on page 3).

[WQL18] L. Wen, H. Qi, and S. Lyu. "Contrast Enhancement Estimation for Digital Image Forensics". In: *ACM Transactions on Multimedia Computing, Communications, and Applications* 14.2 (2018) (Cited on page 3).

[XS12] G. Xu and Y. Q. Shi. "Camera Model Identification Using Local Binary Patterns". In: *Proceedings of the IEEE International Conference on Multimedia and Expo*. 2012, pp. 392–397 (Cited on page 3).

[Xu+09] K. Xu et al. "Efficient Affinity-based Edit Propagation Using K-D Tree". In: *ACM Transactions on Graphics* 28.5 (2009), 118:1–118:6 (Cited on page 7).

[Yan+14] J. Yang et al. "An Effective Method for Detecting Double JPEG Compression With the Same Quantization Matrix". In: *IEEE Transactions on Information Forensics and Security* 9.11 (2014), pp. 1933–1942 (Cited on page 3).

[Yan+17] Y. Yan et al. "Image Deblurring via Extreme Channels Prior". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 6978–6986 (Cited on page 10).

[Yan+19] P. Yang et al. "Source camera identification based on content-adaptive fusion residual networks". In: *Pattern Recognition Letters* 119 (2019), pp. 195 –204 (Cited on page 3).

[Yao+17] H. Yao et al. "Detecting Image Splicing Based on Noise Level Inconsistency". In: *Multimedia Tools and Applications* 76.10 (2017), 12457–12479 (Cited on page 3).

[Yao+18] Y. Yao et al. "Distinguishing Computer-Generated Graphics from Natural Images Based on Sensor Pattern Noise and Deep Learning". In: *Sensors* 18.4 (2018) (Cited on pages 15, 16, 86, 87).

[YDF19] N. Yu, L. Davis, and M. Fritz. "Attributing Fake Images to GANs: Learning and Analyzing GAN Fingerprints". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 7555–7565 (Cited on page 3).

[YNY17] J. Ye, J. Ni, and Y. Yi. "Deep Learning Hierarchical Representations for Image Steganalysis". In: *IEEE Transactions on Information Forensics and Security* 12.11 (2017), pp. 2545–2557 (Cited on page 17).

[Yos+14] J. Yosinski et al. "How Transferable Are Features in Deep Neural Networks?" In: *Proceedings of the Advances in Neural Information Processing Systems*. 2014, pp. 3320–3328 (Cited on pages 50, 56, 67).

[Yos+15] J. Yosinski et al. "Understanding Neural Networks through Deep Visualization". In: *Proceedings of the ICML Deep Learning Workshop*. 2015, pp. 1–12 (Cited on pages 68, 70).

[YRC19] Y. Yan, W. Ren, and X. Cao. "Recolored image detection via a deep discriminative model". In: *IEEE Transactions on Information Forensics and Security* 14.1 (2019), pp. 5–17 (Cited on pages 3, 9, 10, 37, 100).

[Yua+19] X. Yuan et al. "Adversarial examples: Attacks and defenses for deep learning". In: *IEEE Transactions on Neural Networks and Learning Systems* 30.9 (2019), pp. 2805–2824 (Cited on page 77).

[Zen+19] J. Zeng et al. "WISERNet: Wider separate-then-reunion network for steganalysis of color images". In: *IEEE Transactions on Information Forensics and Security* 14.10 (2019), pp. 2735–2748 (Cited on pages 10, 24, 38).

[Zha+09] W. Zhang et al. "Detecting photographic composites using shadows". In: *Proceedings of the IEEE International Conference on Multimedia and Expo*. 2009, pp. 1042–1045 (Cited on page 3).

[Zha+10] W. Zhang et al. "Detecting and Extracting the Photo Composites Using Planar Homography and Graph Cut". In: *IEEE Transactions on Information Forensics and Security* 5.3 (2010), pp. 544–555 (Cited on page 3).

[Zha+18] H. Zhang et al. "Mixup: Beyond Empirical Risk Minimization". In: *Proceedings of the International Conference on Learning Representations*. 2018, pp. 1–13 (Cited on pages 34, 35, 90, 92).

[Zho+18] P. Zhou et al. "Learning rich features for image manipulation detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1053–1061 (Cited on pages 3, 37).

[Zhu+18] L. Zhuo et al. "Fake colorized image detection with channel-wise convolution based deep-learning framework". In: *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*. 2018, pp. 733–736 (Cited on pages 9, 10, 24–26, 32–34, 38–42).

[ZIE16] R. Zhang, P. Isola, and A. A. Efros. "Colorful image colorization". In: *Proceedings of the European Conference on Computer Vision*. 2016, pp. 649–666 (Cited on pages 7, 9, 10, 26, 31, 32, 39).

[ZKC19] X. Zhang, S. Karaman, and S.-F. Chang. "Detecting and Simulating Artifacts in GAN Fake Images". In: *Proceedings of the IEEE International Workshop on Information Forensics and Security*. 2019, pp. 1–6 (Cited on page 3).

[ZQY19] N. Zhu, M. Qin, and Y. Yin. "Recaptured image detection based on convolutional neural networks with local binary patterns coding". In: *Proceedings of the International Workshop on Pattern Recognition*. Vol. 11198. 2019, pp. 11–16 (Cited on page 3).

[ZWN12] R. Zhang, R.-D. Wang, and T.-T. Ng. "Distinguishing photographic images and photorealistic computer graphics using visual vocabulary on local image edges". In: *Proceedings of the International Workshop on Digital-Forensics and Watermarking*. 2012, pp. 292–305 (Cited on pages 12, 13, 55, 61).

Author's Publications

International Journals

- **Weize Quan**, Kai Wang, Dong-Ming Yan, and Xiaopeng Zhang, "Distinguishing between natural and computer-generated images using convolutional neural networks", *IEEE Transactions on Information Forensics and Security*, vol.13, no.11, pp.2772-2787, 2018.
- **Weize Quan**, Kai Wang, Dong-Ming Yan, Xiaopeng Zhang, and Denis Pellerin, "Learn with diversity and from harder samples: Improving the generalization of CNN-based detection of computer-generated images", *Forensic Science International: Digital Investigation*, vol.35, pp.301023, 2020.
- Ruisong Zhang, **Weize Quan**, Lubin Fan, Liming Hu, and Dong-Ming Yan, "Distinguishing computer-generated images from natural images using channel and pixel correlation", *Journal of Computer Science and Technology*, vol.35, no.3, pp.592-602, 2020.

International Conferences

- **Weize Quan**, Kai Wang, Dong-Ming Yan, Denis Pellerin, and Xiaopeng Zhang, "Improving the generalization of colorized image detection with enhanced training of CNN", In: *Proceedings of the International Symposium on Image and Signal Processing and Analysis*, Dubrovnik, Croatia, pp.246-252, 2019.
- **Weize Quan**, Kai Wang, Dong-Ming Yan, Denis Pellerin, and Xiaopeng Zhang, "Impact of data preparation and CNN's first layer on performance of image forensics: a case study of detecting colorized images", In: *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence - Companion Volume*, Thessaloniki, Greece, pp.127-131, 2019.

Title: Detection of Computer-Generated Images via Deep Learning

Abstract: With the advances of image editing and generation software tools, it has become easier to tamper with the content of images or create new images, even for novices. These generated images, such as computer graphics (CG) image and colorized image (CI), have high-quality visual realism, and potentially throw huge threats to many important scenarios. For instance, the judicial departments need to verify that pictures are not produced by computer graphics rendering technology, colorized images can cause recognition/monitoring systems to produce incorrect decisions, and so on. Therefore, the detection of computer-generated images has attracted widespread attention in the multimedia security research community. In this thesis, we study the identification of different computer-generated images including CG image and CI, namely, identifying whether an image is acquired by a camera or generated by a computer program. The main objective is to design an efficient detector, which has high classification accuracy and good generalization capability. Specifically, we consider dataset construction, network architecture, training methodology, visualization and understanding, for the considered forensic problems. The main contributions are: (1) a colorized image detection method based on negative sample insertion, (2) a generalization method for colorized image detection, (3) a method for the identification of natural image (NI) and CG image based on CNN (Convolutional Neural Network), and (4) a CG image identification method based on the enhancement of feature diversity and adversarial samples.

Keywords: Image Forensics, Deep Learning, Computer-Generated Image, Colorized Image, Generalization, Trustworthiness

Titre : Détection d'images générées par ordinateur basée sur l'apprentissage profond

Résumé : Avec les progrès des outils logiciels d'édition et de génération d'images, il est devenu plus facile de falsifier le contenu des images ou de créer de nouvelles images, même pour les novices. Ces images générées, telles que l'image d'infographie (CG) et l'image colorisée (CI), ont un réalisme visuel de haute qualité et peuvent potentiellement menacer de nombreuses applications importantes. Par exemple, les services judiciaires doivent vérifier que les images ne sont pas produites par la technologie de rendu infographique, les images colorisées peuvent amener les systèmes de reconnaissance / surveillance à produire des décisions incorrectes, etc. Par conséquent, la détection d'images générées par ordinateur a attiré une large attention dans la communauté de recherche en sécurité multimédia. Dans cette thèse, nous étudions l'identification de différentes images générées par ordinateur dont les images CG et CI. Nous nous intéressons à identifier si une image est acquise par une caméra ou générée par un programme informatique. L'objectif principal est de concevoir un détecteur efficace, qui a une précision de classification élevée et une bonne capacité de généralisation. Plus précisément, nous considérons la construction de jeux de données, l'architecture du réseau, la méthodologie d'entraînement, la visualisation et la compréhension, pour les problèmes criminalistiques considérés. Les principales contributions sont : (1) une méthode de détection d'image colorisée basée sur l'insertion d'échantillon négatif, (2) une méthode d'amélioration de généralisation pour la détection d'image colorisée, (3) une méthode d'identification d'image naturelle (NI) et d'image CG basée sur CNN (réseau de neurones convolutifs), et (4) une méthode d'identification d'image CG basée sur l'amélioration de la diversité des caractéristiques et des échantillons contradictoires.

Mots clés : Criminalistique des images, Apprentissage profond, Image générée par ordinateur, Image colorisée, Généralisation, Fiabilité

GIPSA-lab, 11 rue des Mathématiques, Grenoble Campus BP 46,
F-38402 Saint Martin d'Hères CEDEX, France