On the use of pattern recognition technique to develop data hiding schemes : application to document security

Vinh Loc Cu

UNIVERSITÉ DE LA ROCHELLE

ÉCOLE DOCTORALE EUCLIDE

Laboratoire Informatique, Image et Interaction (L3i)

THÈSE présentée par :

CU Vinh Loc

soutenue le : **19 juillet 2019**

pour obtenir le grade de : **Docteur de l'Université de La Rochelle**

Discipline : **Informatique et Applications**

On the use of pattern recognition technique to develop data hiding schemes - Application to document security

---

JURY

Directeur	**Jean-Christophe BURIE**	Professeur, Université de La Rochelle, France
Co-encadrant	**Jean-Marc OGIER**	Professeur, Université de La Rochelle, France
Rapporteurs	**Faisal Shafait**	Professeur, National University of Sciences and Technology, Islamabad, Pakistan
	Utpal Garai	Professeur, Indian Statistical Institute, Kolkata, India
Examinateurs	**Nicole VINCENT**	Professeur, Université Paris Descartes, France
	William PUECH	Professeur, Université de Montpellier, France

Résumé

Au cours des dernières années, la croissance rapide des technologies de l'information et de l'usage du numérique a rendu les images de documents plus omniprésentes que jamais. Dans les faits, il existe une grande variété de documents administratifs et commerciaux ayant une valeur "juridique" tels que les certificats, les diplômes, les contrats, les factures, etc. Ces documents sont utilisés par les institutions, les banques, les assurances, les établissements d'enseignement, etc. Par souci de simplicité, ces documents sont souvent échangés par des canaux numériques (la messagerie électronique, transfert de fichiers). L'interception de ces documents et leur potentielle falsification est devenue une question inévitable, en particulier avec le développement de la cybercriminalité. Par conséquent, la fiabilité de ces documents numériques peut être remise en question avec un impact important sur la confiance et entrainer des conséquences pénales, économiques et sociales en cas de fraudes avérées. Pour protéger ces documents numériques contre toute ingérence non autorisée, le domaine de la lutte contre la fraude a évolué et attiré l'attention des chercheurs de la communauté de l'analyse et de la reconnaissance de documents. Une solution efficace pour lutter contre la fraude consiste à dissimuler des données en utilisant des techniques de reconnaissance de formes.

L'objectif de ce travail est de développer des approches fiables pour dissimuler des informations et être capable de vérifier si un document est authentique ou falsifié. Les problématiques abordées dans cette thèse concernent: (1) l'extraction de caractéristiques stables dans les documents, même en présence de distorsions; et (2) la capacité à détecter avec précision les informations cachées pour sécuriser les documents notamment lorsque les documents "protégés" sont soumis à des distorsions causées par des processus tels que impression / numérisation ou impression / photocopie / numérisation. La première problématique est abordée en tirant parti des techniques conventionnelles de reconnaissance des formes et

d'approches basées sur les apprentissages profond (deep learning). Plus précisément, nous utilisons des détecteurs de la littérature pour détecter les points caractéristiques au sein des documents et proposons un nouveau détecteur de points caractéristiques pour développer une méthode de stéganographie. Afin d'améliorer la stabilité des caractéristiques face aux distorsions réelles, nous proposons plusieurs approches de tatouage (watermarking) utilisant des régions stables du document au lieu des points caractéristiques. Ces approches combinent des techniques conventionnelles et les réseaux entièrement connectés (FCN). Les réseaux antagonistes génératifs (GAN) sont également utilisés pour produire un document de référence, et générer des caractères alternatifs utilisés pendant le processus de tatouage. Nous proposons ainsi deux approches pour dissimuler et détecter des informations. La première repose sur la modification de l'intensité des pixels, l'autre sur la forme des caractères.

Les évaluations montrent que nos approches sont capables de détecter correctement les informations cachées lorsque les documents "protégés" sont soumis à diverses distorsions. Une comparaison avec les méthodes de la littérature montre que nos approches offrent des performances compétitives en termes de robustesse pour sécuriser différents types de documents.

Abstract

The fast-growing information technologies and digital image technology over the past decades have made digital document images becoming more ubiquitous than ever. In reality, there have been variety of legal documents consisting of administrative and business documents such as certificate, diploma, contract, invoice, etc. These documents are in use in government agencies, banks, educational institutions and so on. Due to convenience of exchanging information, the genuine documents are often transferred from one place to another by using digital channels. The tampering of these documents during the transmission has become an unavoidable matter, especially in the field of cybercrime. Hence, the credibility and trustworthiness of the legal digital documents have been diminished, this often results in a serious aftermath with respect to criminal, economic and social issues. To secure the genuine digital documents against unauthorized interference, the field of document forensics has been evolved, and it has drawn much attention from researchers in the community of document analysis and recognition. One of the efficient solutions to address this matter is data hiding in conjunction with pattern recognition techniques.

The objective of this work is to develop a data hiding framework as trustworthy as possible that enables to verify if a document is genuine or phony. The challenging problems dealt with in this thesis are: (1) extraction of enough stable features from the documents even in the presence of various distortions; and (2) be able to detect precisely hidden information embedded for securing documents from watermarked documents undergone real distortions caused by print-and-scan, or print-photocopy-scan processes. For the former issue, we address it by taking advantage of conventional pattern recognition techniques and deep learning based approaches. Specifically, we utilize well-known detectors to detect feature points from the documents, and propose a new feature point detector for developing a steganography scheme. To enhance feature stability against the real distortions, we approach to develop watermarking systems based on stable regions instead of feature points, which are based

on the conventional techniques and fully convolutional networks (FCN). In addition, the generative adversarial networks (GAN) are also applied to produce a reference document, and character variations or fonts used for watermarking process. For the later issue, we have come up with two approaches to develop data hiding and detection algorithms: one is based on the changing of pixel intensities, and the other is relied on the shape of characters and symbols.

The assessments show that our approaches are able to properly detect the hidden information when the watermarked documents are subjected to various distortions. In comparison with state-of-the-art methods, our approaches give competitive performance in terms of robustness with applications to various types of document.

Acknowledgments

This research has been carried out at the Laboratoy of Information, Image and Interaction (L3i), University of La Rochelle (ULR). It has been funded by Vietnamese government scholarship (project 911), CampusFrance and CPER NUMERIC programme.

This work has been accomplished under the supervision of Professor Jean-Christophe Burie and Professor Jean-Marc Ogier. I would like to express my deepest appreciation to them for their highly professional guidance and limitless support. They have given me high confidence, which is definitely the greatest reward of my endeavors. I still remember the time when I just came to ULR, I was facing with lots of difficulties in my research. They indeed gave me great help and great words of encouragement in the first year of my doctoral studies. It is my honor and pleasure to work with them, and that is definitely priceless. I am very grateful to my supervisors. Besides, I highly appreciate the friendly professional environment created during 3.5 years of my doctoral period.

I would like to thank my friends and researchers at L3i, and in the other places that I have visited during the period of my studies. Valuable remarks provided by the respectful experts in this field like them would help improve the quality of work.

I also would like to express my special thanks to Professor Cheng-Lin Liu and reseachers at National Laboratory of Pattern Recognition (NLPR), Chinese Academy of Science, where I have had opportunities of meeting many outstanding people during nearly two months working there. It has been a pleasure to know them both personally and professionally.

Most importantly, I would like to thank my parents and my family for supporting me, encouraging me, and loving me all the time.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The fast-growing information and digital image technology over the past decades have made digital document images becoming more ubiquitous than ever. In reality, there have been variety of legal documents consisting of administrative and business documents such as certificate, diploma, contract, invoice, etc. These documents are in use in government agencies, banks, educational institutions and so on. The genuine documents are often exchanged by using digital channels. The tampering of these documents during the transmission has become an unavoidable matter, especially in the field of cybercrime. Thus, securing these documents against unauthorized intervention is a matter that draws a lot of attention. It also poses many challenges for researchers, especially in the community of document analysis and recognition.

This chapter presents our motivations and an overview of data hiding techniques relevant for our thesis. In the wake of the popularity of transmission of daily genuine documents over the digital channels, we present the necessity of using data hiding techniques to secure these legal documents against illegal intervention or falsification. We briefly introduce the concept of data hiding technique, its applications and the properties required for a data hiding system. We also point out challenges in designing a data hiding system for document images. In addition, our general framework and contributions are also detailed in this chapter.

1.1 Problem definition

Due to convenience of information exchange, the genuine documents are daily transmitted over the digital channels more than ever. The availability and effeciency of fast-growing advanced technologies make digital data becoming more popular for the end users. Thus, the daily legal documents as contracts, invoices, reports, bank guarantees, balance sheets and so on are mostly scanned and stored under the digital format. These documents are in use in several sectors like government agencies, bank, military, business, etc. The free-access digital information communication also leads to unprecedented opportunities to violate the

usage of these genuine documents. Specifically, these digital documents are easily modified by inauthentic users with the support of available image processing softwares, e.g. Adobe Photoshop, which can be used to delete or replace content in some areas of the documents without leaving any detectable trace. The alteration may be physical or intellectual in which physical alteration refers to crossing-out of items or references, addition of information to change the original content of a document, etc. whereas the other indicates the content of a document that does not accord with the reality such as false description of services, false content of reports, false signature on contracts, etc. The action of illegally modifying legitimate documents is known as tampering, the modified version generated from tamper is regarded as fraudulent document.

	Use of breeder documents	Fraudulent travel and other identification documents	Fraudulent declaration of goods/Mislabelling	Fraudulent import/export certificates	Fraudulent certificates of provenance or origin	Fraudulent work permits	Fraudulent visas	Fraudulent company registration documents	Fraudulent vehicle registration documents	Fraudulent accounting statements	Fraudulent invoices
Criminal finances and money laundering	X	X		X	X			X		X	X
Drug production, trafficking and distribution		X	X	X	X			X	X		X
Environmental crime		X	X	X	X			X			X
Fraud	X	X		X				X		X	X
Intellectual property crime		X	X	X	X			X			X
Organised property crime		X	X	X	X	X	X	X	X		X
Migrant smuggling	X	X				X	X	X	X		
Trafficking in human beings	X	X				X	X	X	X		
Trafficking of firearms		X	X	X	X			X			X

Figure 1.1: *The use of different types of fraudulent documents for illegal activities (Council of the European Union (September 2017)).*

According to Interpol 2017 (COM/FS/2017-01/FHT-05), the fraudulent use of identity and travel documents presents a threat to the security of countries and their citizens, the economy and global commerce, and it facilitates a wide range of crimes. The inauthentic users often make fraudulent use of genuine documents in order to carry out their illegal activities. The fraudulent documents are classified into three categories: (*i*) the modified documents are typically based on a genuine document in which a part has been inserted or altered in order to give misleading information about the person who presents it. This kind of modified documents is known as forgery; (*ii*) the documents are produced with no authority,

they are not officially recognized. They can occur in various forms and may have the physical appearance of a genuine document. This kind of produced documents is regarded as pseudo document; (*iii*) the documents that are constituted from an unauthorized reproduction of a genuine document. These documents are not legitimately manufactured or recognized by an official authority, they are known as counterfeit.

According to Council of the European Union (September 2017), document fraud entails the production and the use of false documents as well as the misuse of genuine documents. The use of fraudulent documents in the EU has significantly increased the unauthorised users counterfeit or fabricate different types of paperwork and administrative documents such as transport certificates or company registration forms to facilitate their illegal activities. Figure 1.1 shows the use of different types of fraudulent documents in illegitimate activities.

In France, according to a recent survey from Pwc 2016 report "La fraude en entreprise", the 68-percent of interviewed companies have identified at least one case of fraud in 2016.

With the growing risk of using falsified documents in governmental agencies and business, securing the legal documents against unauthorired interventions is not just a matter that draws much attention from those who make the genuine documents. It also introduces many challenges which are dealing with by reseachers, especially in the community of document analysis and recognition. In fact, to secure or authenticate genuine documents, several approaches that can be used as follows.

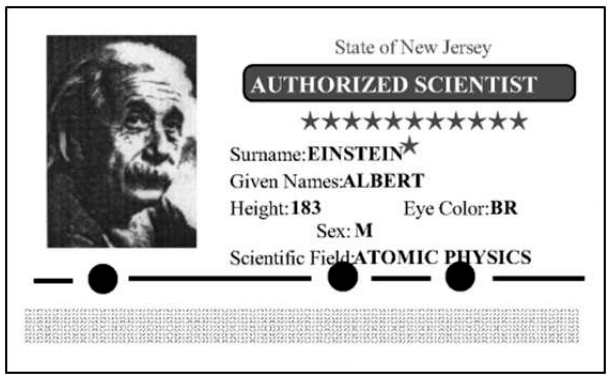


Figure 1.2: *An example of identification document: the holder image or photograph is at top left. Textual personal information is on document in human readable form. Photo signature (encrypted form) in machine readable form is at bottom of document.*

Photo signature[17;18] has been proposed to secure identification documents as passport, driver's licenses, welfare cards, national identification, credit cards, etc. This method utilizes the technique of pattern recognition, public key, digital signature cryptography for establishing and maintaining authentic documents. It is more effective than physical means such as microprinting, embedded holograms and optical laminates which are used to prevent tampering. The identification documents as in Figure 1.2 often consists of holder's photograph and textual personal information. The idea of this approach is to generate a photo signature from the holder photograph in which each holder image or photograph produces a

concise and unique descriptor. The photo signature is then affixed to the documents under an encrypted form for document verification.

Biometric-based authentication systems[19] are relevant to human characteristics that identify individuals based on fingerprint as illustrated in Figure 1.3(a), voice, face, hand geometry, signature, iris and retina. These methods are more effective than traditional authentication systems such as knowledge-based information or token-based information because biometrics can not be borrowed, stolen, or forgotten. They are more convenient because there is no need to refer to any extra information. One of the widely used applications of biometrics is fingerprint-based authentication system. Although this method gives high effectiveness, it has also posed some limitations as it is used for remote authentication[20]. By using spoofing approaches, the falsified fingerprint can be easily generated by malicious users.

Figure 1.3: *(a) is a fingerprint sample from the United States National Institute of Standards and Technology database. (b), (c) and (d) are the appearance of barcode, quick response code and document signature respectively.*

Digital signature associated with barcode-based system[21;22] has been proposed to verify the integrity of the document text content and the origin of a document. The idea of this approach is to use 1D or 2D barcode to carry integrity and authenticity information (presented as digital signature form) within the documents. This method allows to verify the document author when the document has not been falsified. The system also enables to detect whether or not the received document has been altered by malicious users. In general, the technique combining digital signature and barcode provides high effectiveness without any requirement of special equipment or expertise. Similar to barcode, quick response code as in Figure 1.3(c) has also been used to develop document authentication system[23]. To improve the performance of the system, the quick response code is sometimes combined with other methods as visual secret sharing[24].

Document signature[25;26] has been put forward for the purpose of document security. The idea of this approach is to extract document content and transform the extracted content to its proper format. A hash algorithm is then applied on it to generate a document signature. The document signature as in Figure 1.3(d) is then affixed to the document with the support of barcode or quick response code for document verification. Besides, the approach of passive image forensics has been implemented to detect tampered regions in the images, and the existing approaches[27–33] show that they provide high performance when they are applied on

natural images.

As we can see that various approaches have been proposed to protect legal documents against unauthorized interventions. Apart from their advantages, each method has also the drawbacks mentioned above as in the case of fingerprint-based authentication system. With regard to photo signature, barcode (1D, 2D), quick response code and document signature, they enables to visibly affix a secret information or digital signature to the documents. We refer to visibly affixed information on documents as visible codes. Although these approaches give high performance for the purpose of securing genuine documents, they also have introduced some concerns such as: (1) the visible codes are almost unattractive and meaningless to normal users; (2) the visible codes often require a certain blank space on documents for embedding in and sometimes make the genuine documents losing their aesthetics; (3) this kind of codes also draws much attention from malicious users who try to break these codes to illegitimately access to embedded information.

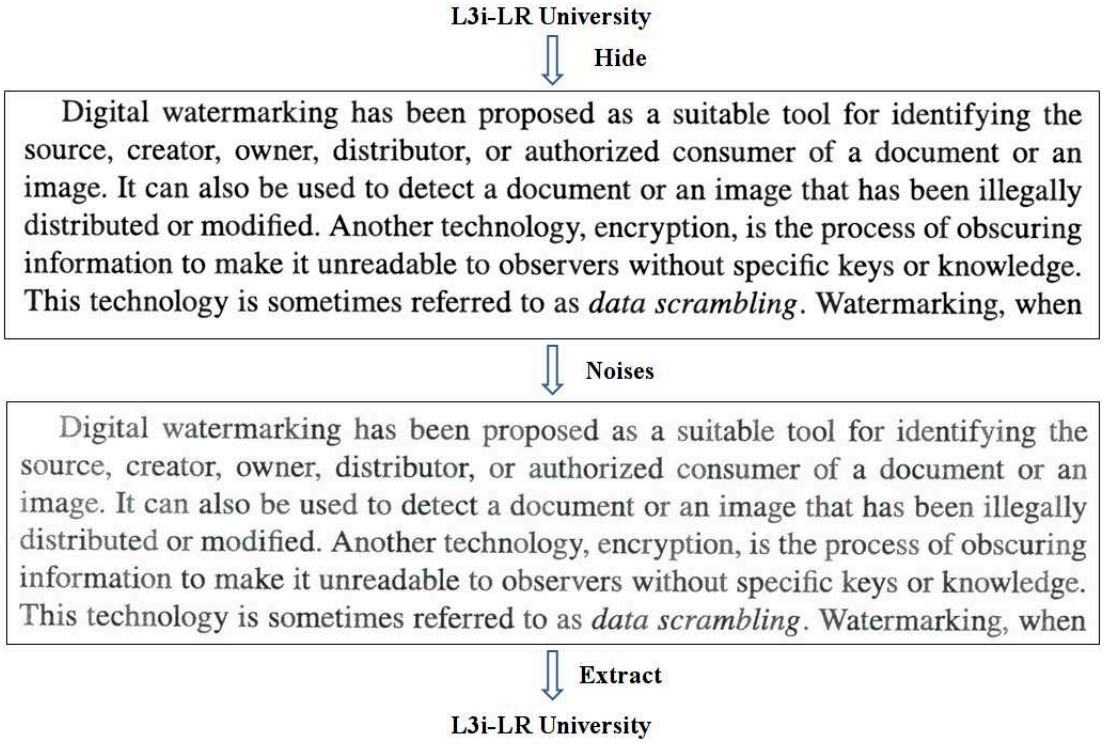


Figure 1.4: *The illustration of data hiding technique that enables to hide a secret information into a document and detect the hidden information from the watermarked document, even in the presence of distortions.*

To overcome the issues of presented approaches, the data hiding technique can be used as an effectively alternative solution. This technique enables to integrate invisibly a security feature within the genuine documents, and the security feature is later extracted by recipients to verify whether a received document is genuine or falsified, as illustrated in Figure 1.4. Moreover, recently, the data hiding technique is also applied for protecting pre-trained models of deep neural networks[34;35]. The purpose of this application is to reduce complexity and

save time for training models. In fact, the authors are willing to release their trained models to help reseachers and engineers develop deep learning-based system and do research with less effort. However, they want to remain the ownership authorization of their pre-trained models. Thus, prior to releasing the pre-trained models publicly, the authors hide necessary information into their pre-trained models for later verification. As a result, we are motivated by the advantage of data hiding technique, so we have adopted data hiding approach in the field of securing legal documents.

1.2 An overview of data hiding technique

Data hiding[36] is a technique that enables to hide an information (e.g. secret message or watermark) into a carrier (e.g. image, video, software, etc.) without causing much perceptual distortion of the original content. In the scope of this thesis, we refer to documents or images as carrier. Digital watermarking and steganography are two sub-branches of data hiding. The hidden information is either visible or invisible with regard to watermarking system whereas it is always invisible for steganography. Depending on the purpose of use like document authentication, document security or covert communication, the watermarking or steganography approach is going to be selected to meet the requirement of real application. With the invisible data hiding scheme, hiding information embedded in documents should not be perceived by normal observers. This means that the observers have difficulties to recognize the regions of the document where the secret information is hidden inside. For detection process, extracting the hidden information without any reference to the original document is regarded as a blind detection. A general data hiding system as shown in Figure 1.5 consists of three main components such as information hiding process, distortion channel and information detection process. During its life, the document, where secret information is hidden, is more or less subjected to distortions which can be intentional or unintentional. In general, there are fundamental properties that we need to take into consideration when designing a data hiding system as follows.

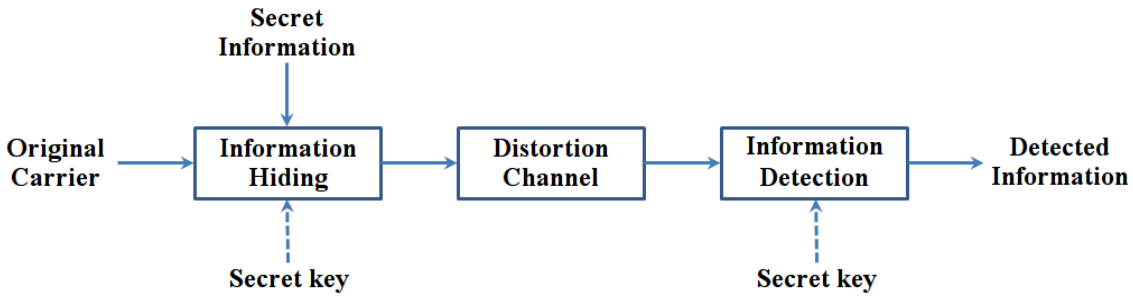


Figure 1.5: *Fundamental components of a data hiding system.*

Imperceptibility: Hiding a secret information into documents has to degrade minimally the quality of document content, and it should not be perceptible by the human visual system. This property refers to a similarity between the document images before and after hiding a secret information. In some data hiding applications, the hidden positions can be recognized

by experts in the field. However, it still remains unnoticed with respect to the eyes of normal end-users.

Capacity: This property indicates the amount of information that can be stored in a document image. Depending on the application, the amount of information would be more or less. For copy control application, for example, a little amount of information seems to be sufficient. Meanwhile, the intellectual property applications require more capacity to store necessary information such as author, copyright, etc. It is noticeable that high capacity could result in reduction of quality of images after hiding a secret information.

Robustness: It refers to the ability of a data hiding system against intentional or unintentional distortions. In other words, it is able to properly detect the hidden secret information in case the documents undergo distortions. These distortions can be caused by JPEG compression, geometric transformation, print-and-scan process, print-photocopy-scan process, print-photograph and so on. This property often conflicts with the imperceptibility and capacity. It means that increasing the robustness may result in reducing the imperceptibility and capacity.

Security: It is the inability for malicious users having permission to decode the hidden secret information. The process of hiding secret information should be difficult for unauthorized users to destroy the hidden information without the knowledge of a secret key.

Digital watermarking[37] has been proposed as an effective solution for determining creator, owner, source, authorised receivers of a document or image. Besides, this method has been also used to detect a document or image that is illegitimately modified or distributed. Unlike traditional watermarking for printed or visible watermark, the digital watermarking system is mostly designed to be imperceptible to human eye. For the purpose of real application, the watermarking system can be classified into three categories including fragile, semi-fragile and robust. Fragile and semi-fragile watermarks[38;39] are primarily designed for detecting the integrity and authenticity of images. The fragile watermark is always used to detect unauthorized modification for the purpose of image authentication. The robust watermarking scheme[40;41] is mainly designed to withstand distortions such as common image processing operations, geometric transformation, print-and-scan operation, etc., and it is mainly applied for copyright protection. In the context of digital watermarking, the original document is known as a host, the secret information is regarded as a watermark, and the document after hiding a watermark inside is known as a watermarked document.

The watermarking system can be implemented in either spatial domain or transform domain. For watermarking system in the spatial domain, the gray value of pixels in the original images is directly adjusted to obtain the purpose of hiding information. For example, the least significant bit (LSB) algorithm[42] is a popular spatial domain method. For transform domain[41;43], the watermarking scheme is implemented by transforming the images into discrete cosine transform (DCT), discrete Fourier transform (DFT) or discrete wavelet transform (DWT). The information is often hidden into the low and medium frequency areas because of less degradation. The transformed domain is then reversed to achieve the watermarked image. As a result, the watermarking system can be apply for variety of practical

applications such as copyright protection, data authentication, fingerprinting, copy control and device control.

Digital steganography[37] is an art and science of hiding information into images in order to conceal the information and prevent the detection of hidden data. Unlike watermarking applications, steganography is mostly designed for the purpose of covert communication in which this technique aims to obscure the hidden information in an image with the intent of not drawing suspicion that the information is being transferred. Thus, a good steganography should meet the requirements of imperceptibility, capacity and security. To meet the property of security, steganography is often combined with cryptography (symmetric or asymmetric). Basically, there are three types of steganography: (*i*) technical steganography uses scientific methods to hide a secret information, these methods are invisible ink, microdots, shaved head; (*ii*) linguistic steganography applies written natural language to conceal a secret message. Hiding information can be performed by using visual symbols, signs or special pre-defined pattern which is not recognized by normal observers; (*iii*) digital steganography is the art of hiding secret information into digital media which is increasingly used so far. The terminologies used in steganography are defined as follows. The original document is known as a cover whereas the document after hiding a secret information inside is regarded as a stego-document.

Depending on the practical applications, the performance of a data hiding system is reflected through its properties. As mentioned above, the imperceptibility is measured by the quality of watermarked images or stego-images. The higher the quality of watermarked images or stego-images, the better the imperceptibility will be. The methods to measure this property consist of peak signal to noise ratio (PSNR), structural similarity index (SSIM)[44], distance reciprocal distortion measure (DRDM)[45], edge line segment similarity[46] and flippability score[47]. Meanwhile, the evaluation of robustness is measured by bit error rate, or false positive rate, false negative rate and true positive rate. The smaller the value of false positive rate and false negative rate, the better the accuracy ratio of hidden data detection will obtain.

1.3 Objective and challenges

From the merits and demerits of existing authentication systems, we utilize pattern recognition techniques in conjunction with document image analysis to develop data hiding systems in order to secure documents against malicious activities as illustrated in Figure 1.6. There are two main objectives from the context of our study. From a perspective of practical applications, we are going to provide a framework that is reliable enough to meet the necessary requirements that a document might have to undergo during the process of document exchange. From a conceptual perspective, we are going to provide data hiding algorithms that are robust enough to be capable of hiding a secret information into and detecting the hidden information from various types of documents in the presence of possible distortions.

Several data hiding approaches have been proposed for document images and natural

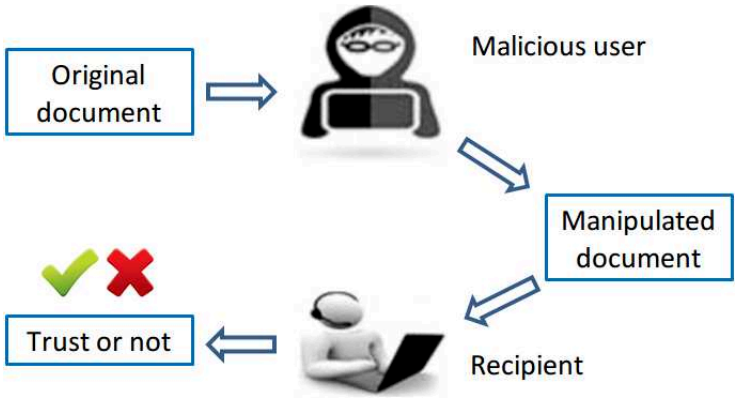


Figure 1.6: *The illustration of securing documents against inauthentic activities.*

images. For document images, most existing data hiding approaches have been proposed for text content[1;48–54], or specific languages such as Indian or Chinese. Recent works[55;56] have been designed for general content but they have been developed in the transform domain. The methods for text content seem not to work well for general content because these works provide no detail on how to identify and separate text and non-text elements. In addition, the approaches for natural images[57–61] can be applied for document images but they need to be adjusted to improve the robustness (e.g. extracted features as keypoints or edges are unstable regarding to distorted document images) or to eliminate empty regions (e.g. document images often contain a lot of empty areas). Moreover, deep learning is also exploited to develop data hiding systems, which are designed for natural images[62;63] or text document[53;54]. The types of deep neural network used in these works are convolutional neural network (CNN), or CNN-based autoencoder.

In the scope of our study, to make a data hiding as trustworthy as possible, we have to deal with three main challenging problems:

- The first challenge is to design a data hiding system which is compatible with diverse documents. Due to the nature of the administrative and business documents, the system should be applicable for typewritten and handwritten documents where the content of these documents could be a combination of picture, text, table and so on. Besides, these documents could be grayscale or binary documents.

- Second, there are always a conflict among imperceptibility, capacity and robustness. Thus, hiding authentic information into documents has to preserve the property of robustness as much as possible whereas it has to meet the other requirements. In fact, the features extracted from document images by using conventional pattern recognition technique are relatively unstable against distortions. The instability of extracted features will result in reducing the precision of detecting the hidden information.

- The third challenge is that the data hiding algorithms should be robust enough to hide the secret information into genuine documents and to detect the hidden information from the watermarked or stego documents, which are possibly subjected to distortions

like JPEG compression (due to size reduction or storing scanned versions), geometric transformation, print-and-scan process (scanning at various resolutions), and printing-photocopying-scanning process (multi rounds of photocopying prior to scanning at various resolutions).

The general framework of our data hiding system is shown in Figure 1.7.

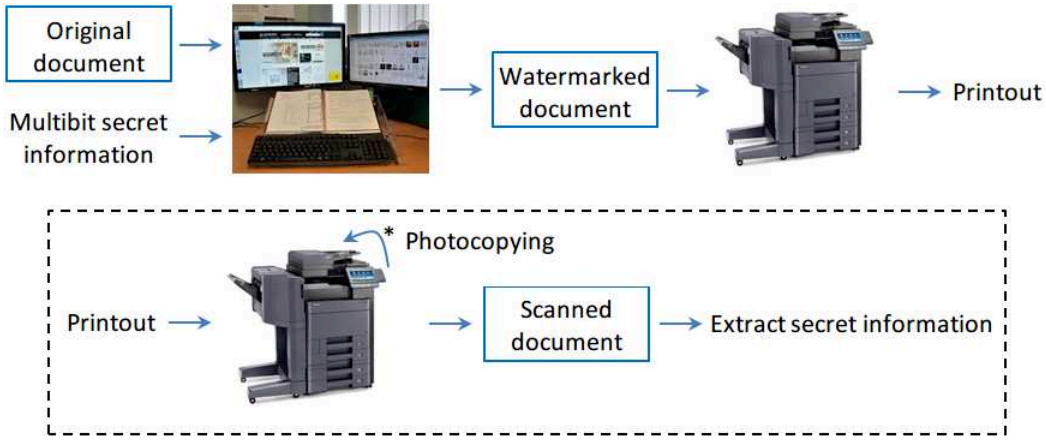


Figure 1.7: *The general framework of our data hiding system.*

1.4 Contributions

In this research, we have contributed a number of achievements to the field of securing legal document by taking advantage of pattern recognition techniques in combination with data hiding techniques. Specifically, we have developed the data hiding schemes based upon conventional approaches and deep learning approaches. On the one hand, we have proposed a number of solutions to improve the stability of features extracted from the document images, which are used to develop data hiding systems in the spatial domain. These solutions consist of feature point extraction, detection of stable regions, generation of an intermediate document from the input document, generation of character and symbol variants. On the other hand, we have come up with a number of new robust data hiding algorithms which are capable of properly detecting the hidden information from the watermarked documents which are undergone different transformations such as JPEG compression, geometric transformation, print-and-scan operations and print-photocopy-scan operations. The summary of our contributions is listed as follows.

1. Conventional approaches

   (a) Constructing a steganography scheme based on the feature points which are extracted by BRISK detector[64].

   (b) Proposing a new feature point detector by using a combination of non-subsampled contourlet transform (NSCT)[65] and distance transform.

(c) Developing a watermarking scheme based on stable region approach, which is performed by combining common image processing operations and NSCT.

2. Deep learning approaches

   (a) An approach for detecting watermarking regions has been developed by using fully convolutional networks (FCN)[66].

   (b) A method for generating an intermediate document from the input document has been proposed by utilizing generative adversarial networks (GAN)[67]. The generated document is then used as a reference for watermarking process.

   (c) Another GAN network has also been proposed to generate variations of document characters and symbols from their skeleton, which is known as font generation.

   (d) Taking advantage of FCN network to detect the character variants from the watermarked documents.

1.5 Thesis organization

The thesis is organised in six chapters whose current chapter presents an introduction to the thesis. The rest of the thesis is structured as follows.

Chapter 2 presents a detailed review of state-of-the-art approaches that are relevant to the data hiding system, evaluation of geometric correction method based on feature points, and assessment of stability of extracted features used to construct hiding regions. Regarding the data hiding schemes, we have reviewed watermarking schemes for document images with textual content, document images with hybrid content and natural images, watermarking scheme for protecting pre-trained model of deep neural network, forgery detection, and steganography and steganalysis schemes for natural images. The reviewed schemes will also cover typical techniques used to detect image features as well as data hiding algorithms in both spatial and transform domain.

Chapter 3 introduces feature points-based steganography scheme and stable regions-based watermarking scheme. In the former, we use speeded up robust features (SUFR) detector to extract feature point for constructing hiding regions, local binary pattern (LBP) and local ternary pattern (LTP) for determining potential hiding positions. Hiding data into document is based on odd and even feature of gray level values, and we also utilize error correction code to enhance the accuracy of hidden data detection. Besides, a new feature point detector for stability improvement is also presented in this chapter. In the later, we combine common image processing operations and non-supsampled contourlet transform to detect the stable hiding regions. The watermarking algorithm is developed depending on each group of pixel values assigned with weights in which we have eliminated pixel values that are vulnerable to distortions.

Chapter 4 presents the watermarking schemes designed by taking advantage of deep learning. Specifically, the FCN network for the purpose of semantic image segmentation is adjusted to solve the problem of detecting watermark hiding regions, and to detect positions of character variants. In addition, we introduce two watermarking algorithms which are based on changing pixel intensities for watermarking process, and they are able to apply for both typewritten and handwritten documents. Besides, we make use of GAN to develop two other watermarking schemes: the first scheme is applied to produce a good quality document from an input document, and the generated document is then used as a reference for developing watermarking algorithms; meanwhile, the other is based on the shape of document characters and symbols, and it is performed by generating new fonts of document or variations of document characters and symbols. Finally, a watermarking scheme for binary documents is also presented in this chapter. We introduce hiding patterns used for detecting edge features and corner features, and these features are utilized to carry information bits. Two watermarking algorithms for binary documents are then presented in which one is designed to hide information bit directly into each hiding pattern, and the other is based on the disparity between the number of edge features and the number of corner features in each group of objects.

In **Chapter 5**, we detail the performance of our approaches through various experiments. The performance is evaluated based on the following factors such as imperceptibility, capacity and robustness against common distortions like JPEG compression, geometric transformation, and real noises caused by print-and-scan process and print-photocopy-scan process. The experimental results show that our proposed approaches are able to detect the hidden information from: the watermarked douments scanned at low resolutions, even at the resolution of 200 dpi for some types of document; and the watermarked documents suffered two rounds of photocopying prior to scanning at various resolutions. Besides, we also present quantitative comparison to prove the improvement of our approaches, and comparison with other typical schemes.

Chapter 6 elaborates a summary of thesis contributions and possible improvements of data hiding scheme for securing documents.

Chapter 2

Review of data hiding researches in the literature

This chapter provides a comprehensive survey on data hiding research for document and natural images, and feature points-based methods for geometric correction and construction of hiding regions. For watermarking approaches, we review the existing techniques by classifying them into schemes dedicated to: text documents, hybrid documents, natural images and pre-trained models of deep neural networks. Meanwhile, steganography schemes are only designed for natural images. With each of data hiding schemes, we also introduce techniques used to detect image features as well as data hiding and detection algorithm, which are implemented in either spatial domain or frequency domain. The data hiding schemes for binary images are also reviewed in this chapter. In addition, we also present steganalysis schemes and forgery detection. Finally, we evaluate the performance of geometric correction processes and hiding region construction processes that are based on feature points, and which are usually designed for natural images in order to determine their relevance for document images.

2.1 Watermarking techniques

A digital watermarking system enables to hide a secret information with moderate capacity into images or documents in such a way the hidden information can be precisely extracted in the presence of various distortions caused by image processing operations, printing-photocopying-scanning process, operation of image capture. The most common important requirements for digital watermarking are imperceptibility, robustness and security.

2.1.1 Watermarking for text documents

Low *et al.*[68] have proposed a method based on line and word shifting. Regarding line shifting method, the marked line can be slightly shifted up or down from its normal position to carry one watermark bit. Meanwhile, word shifting method divides the words of each line into three groups of words with a large middle group. The middle group is slightly shifted left or right as hiding watermark bit whereas two neighboring blocks are regarded as control blocks, and remain stationary. This approach requires the original document when detecting watermark. A feature calibration method has been proposed by Amano and Misaki[69] in which each character is detected and grouped into text line, then the bounding box of text line is calculated. Each bounding box is divided into four partitions which are classified into four sets. The average width of the horizontal strokes of characters is computed as feature. The authors defined two operations like "make fat" and "make thin" as in Figure 2.1 to add or remove pixel values on these four sets for hiding watermark bits. This method is able to resist to noises caused by print-and-scan operations.

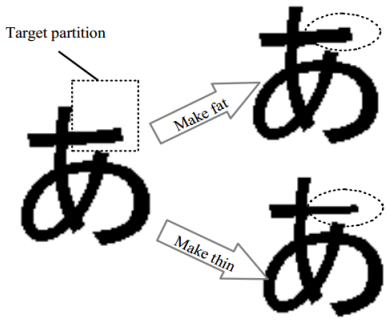


Figure 2.1: *The demonstration of "make fat" and "make thin" operations.*

Kim and Oh[1] have proposed an approach by using edge direction histograms. The edge direction histograms are calculated by making use of Sobel edge operator. The watermarking algorithm is based on the shape of the histogram. The edge direction is quantized into 16 levels. Figure 2.2 gives an example of edge direction and the encoding of 16 directions. The document is partitioned into blocks in which the first three blocks are used as a reference of edge direction histogram, the length of diagonal directions of remaining blocks are adjusted to carry watermark bits. This approach is robust against some distortions like rotation, blurring, sharping and etc.

Kim *et al.*[49] have put forward word classification and inter-word space statistics for text document. The number of words in each line are identified wherein the width of words is used as a feature for word classification. All words in the document are classified, and the segment is defined to be a group of consecutive words in a line. The watermark bits are hidden by slightly shifting words to left or right in each segment. This method gives good imperceptibility and robustness against the error of page segmentation. A weight-invariant partition-based scheme presented by Hu[50] divides document into partitions by using support vector machine (SVM). The authors claim that the weight of a partition is not likely to be significantly changed due to noises. With this approach, the watermark is then hidden into

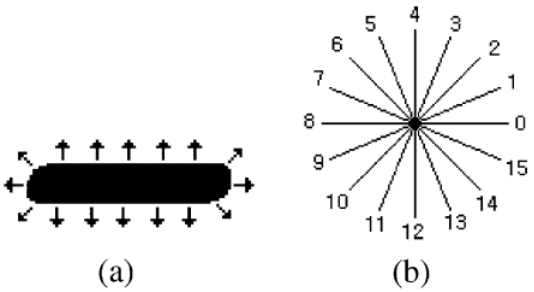


Figure 2.2: *Encoding of edge directions*[1] *extracted from: (a) edge directions, and (b) quantization.*

uniform partitions by adding or removing pixel values to a text line such that the line in the partition is suitable for hiding watermark bits. Another inter-word space method[51] hides secret information by partitioning words in a text row into two sets as in Figure 2.3, and each set contains the equal number of inter-word spaces. If there are an odd number of inter-word spaces within a text row, one of them will be removed. Hiding process is carried out by adjusting these spaces. The inter-word space methods give good imperceptibility and robustness to print-photocopy-scan (PCS) noises. In general, the approach based on word or line shifting can be applied for both grayscale and binary documents.

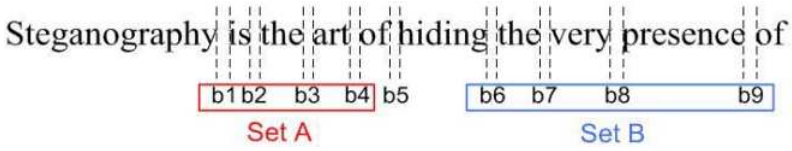


Figure 2.3: *An illustration of inter-word space method in which "set A" and "set B" contain the equal number of space elements.*

Palit and Garain[48] have proposed a method by hiding data into prototypes constructed from document. The authors use pattern matching techniques for clustering symbols into prototypes, and the image of prototypes is represented under binarized form. This method can withstand distortions like JPEG compression, uniform random noise and scaling. However, it is only designed for Indian text document. Another scheme based on continuous line[70] makes use all of word spaces and considers the document as one long line. The main idea of this method is to take word spaces, and to divide them into pairs of sets in which each set contains three consecutive word spaces as depicted in Figure 2.4. If the word spaces are not enough to form a pair of sets, these word spaces are then classified into a spare group. The horizontal and vertical profiles are utilized to partition document into lines, words and letters. To hide information, the difference among the total widths of the sets within each pair is adjusted. This method is able to resist to printing and scanning distortion, and it can be applied for binary and grayscale documents.

Varna *et al.*[52] hides information into the vertically left edge of the character stroke by adding or deleting two groups of pixels which are equidistant from the center as in Figure 2.5. The authors have combined message packetization and error correction code for improving

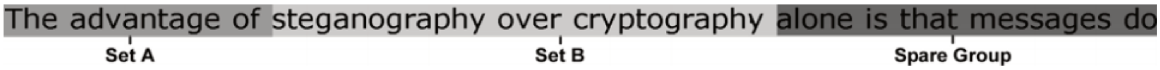


Figure 2.4: *An illustration of a pair of sets ("set A" and "set B") and spare group.*

the performance of their approach. This method is resistant to PCS noises. However, the distortion caused by hiding process is visually perceptible. Tan *et al.*[71] have proposed another work based on stroke direction. Watermark bits are hidden into individual characters by adjusting the directional characteristic of rotatable strokes. This approach gives high capacity, imperceptibility and resistance to PCS noises. However, it can only be applied for Chinese text.

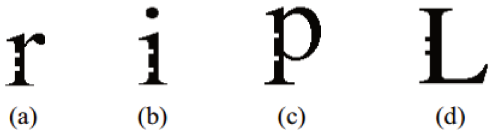


Figure 2.5: *The magnification of characters after hiding information.*

Hiding information into text portion of a document based on the shape of characters (character variations) has been proposed by SOOD company[1]. The first step in this method is to determine a specific character font used in document. With the specified font, the variations corresponding to an original graphic of characters are generated. Each variation is associated with a value which represents information that we need to hide into document. The hiding process is then conducted by replacing at least one original graphic with its variation in which the optical character recognition are employed to detect document characters. The variations of an original graphic are presented in Figure 2.6.

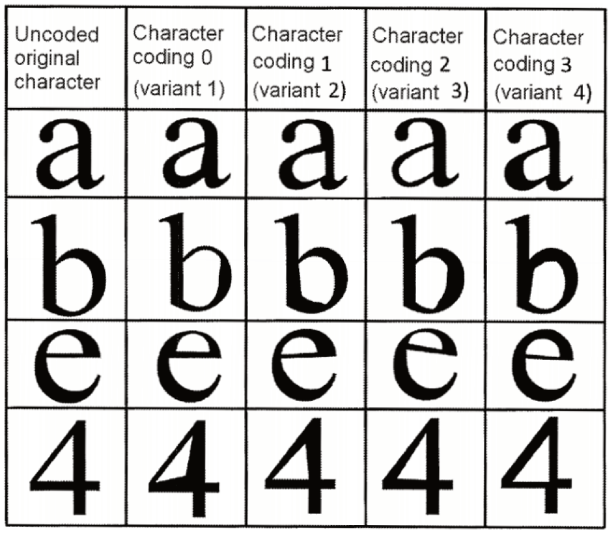


Figure 2.6: *An example of original characters and their corresponding variations.*

[1]http://sood.fr/en/patents/sood_tattoing.html

Similarly, another approach based on the shape of characters has been recently proposed by Chang *et al.*[53;54]. This method hides secret information into text by perturbing the glyphs of text characters. The variation of a character is produced by utilizing the font manifold presented in[72]. To do so, the authors have constructed a glyph codebook containing character variations. CNN is used for both codebook construction and glyph recognition in which the authors develop a lookup table which contains a set of variations for each character in three common fonts such as Times New Roman, Helvetica and Calibri. In addition, the authors have employed error correction code based on Chinese remainder theorem for rectifying recognition errors. This method is robust to real noises like print-and-scan as well as print-and-photograph. Figure 2.7 demonstrates the variations of a character.

Figure 2.7: *The illustration of 5 points (left) around the manifold of a character "a" in Times New Roman font (center) and generated glyphs (right).*

2.1.2 Watermarking for hybrid documents

At the time of conducting our research, we have only found few watermarking schemes designed for documents with mixed content, and they are implemented in transform domain. A combination of discrete cosine transform, singular value decomposition (SVD) and genetic algorithm has been proposed by Horng *et al.*[55]. The idea in combining DCT and SVD is to use luminance masking in accordance with the characteristics of the human visual system to improve noise sensitivity whereas genetic algorithm is used to find the scaling factor for watermarking optimization. The hiding process begins by calculating the masks of the original document and transforming them into coefficients. The information is then hidden into document by adjusting the singular values of transformed document, the singular values of document's mask coefficients and scaling factor. With this combination, the authors claims that their scheme provides high performance and security, and it is able to resist to common image processing operations. The method proposed by Chetan and Nirmala[56] hides data into document by using integer wavelets and block coding of binary watermark. To identify the regions for carrying watermark bits, the document is divided into empty and non-empty blocks depending on the presence of document content. Prior to hiding, the binary watermark is compressed by using binary block coding technique. Next, a level-2 of integer wavelet transformation is applied on the non-empty blocks of document. The low level subband of level-2 of the transformed document is subdivided into blocks with uniform

size, and the compressed watermark is then hidden into these blocks. This approach is capable of robustness against common image processing operations.

2.1.3 Watermarking for natural images

In contrast to watermarking for hybrid document, there are several schemes dedicated to natural images, and they are implemented in both spatial and transform domain.

A salient feature points-based scheme[73] hides a version of inversely normalized watermark from a DCT transformation of original watermark into images. The watermarking process is conducted by transforming the input image into nomalized form using image moments. The feature points are extracted from the normalized image that will act as centers of watermark hiding regions where the watermark is hidden into. The proposed scheme are experimented on feature points extracted from four feature point detectors such as radial symmetry transform (RST), scale-invariant feature transform (SIFT), speeded up robust features (SURF) and features from accelerated segment test (FAST). This scheme is robust to geometric distortions and signal processing distortions. The diagram of watermark hiding process is depicted in Figure 2.8.

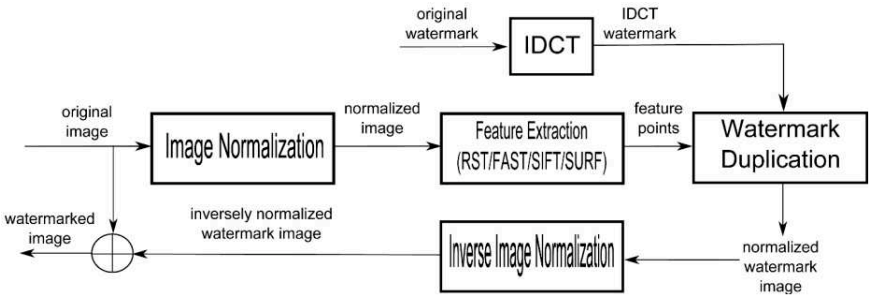


Figure 2.8: *Watermark hiding process: IDCT stands for inverse DCT.*

Discrete fourier transform (DFT) domain along with feature points[74] is utilized to hide information into images. To enhance security feature, the watermark is a pseudo random binary sequence generated by a secret key. It is then hidden into the magnitude of the middle frequencies of DFT which is transformed from the original image. The magnitude of middle frequencies of DFT is choosen because modifying coefficients in the magnitude of low frequencies leads to visible distortion whereas it is vulnerable to JPEG compression with respect to magnitude of high frequencies. Finally, the feature points of a watermarked image are extracted and stored in a file. The data stored in this file is used to match with feature points extracted from distorted image for the purpose of geometric correction. This method is robust to geometric and signal processing distortions. A scheme based on feature points and Zernike moments is proposed by Yuan and Pun[75]. SIFT detector is used to extract feature points, and circular regions centered at these feature points are constructed for watermark hiding and detection wherein the overlapping and small regions are eliminated. To avoid degradation of a watermarked image, each circular region is decomposed into a series of binary images as depicted in Figure 2.9. The Zernike moments are applied to these

binary images, and the watermark is then hidden by altering the magnitude of these local moments. Another method[76] is relied upon feature points extracted by SIFT detector and DFT transformation. The appropriate feature points are selected to construct disks centered at these feature points. For each disk, orientation alignment is performed to obtain rotation invariance. Next, the DFT is applied to these disks, and the secret information is then hidden into the middle frequency coefficients of each disk.

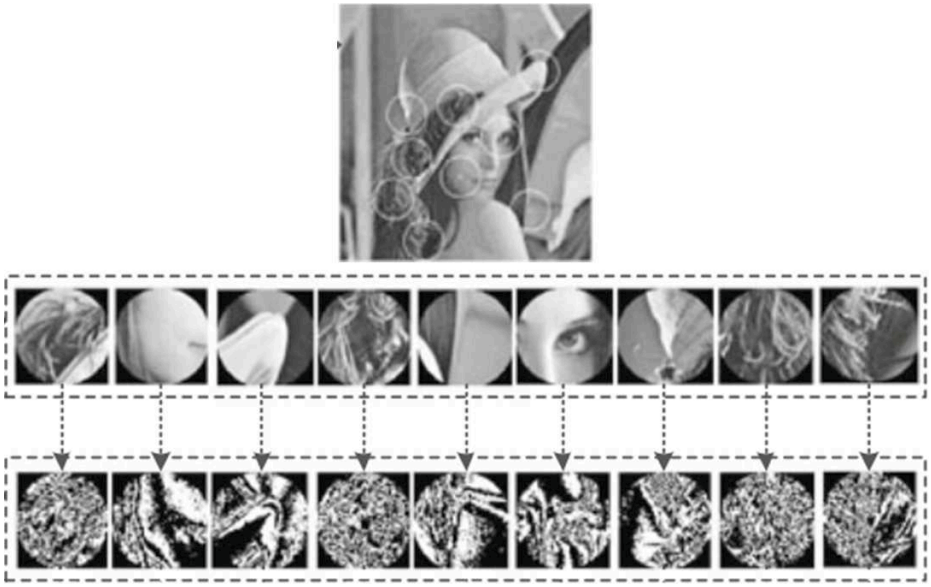


Figure 2.9: *An illustration of circular regions centered at extracted feature points and decomposition of binary images.*

Similarly, Zhang and Li propose a feature points and DCT-based scheme[77] that enables to hide data into circular regions. The authors use SURF detector to extract feature points, and the circular regions are constructed around these feature points wherein the overlapping and small regions are also eliminated. The watermark is then hidden into these circular regions by adjusting coefficients in the middle frequency of DCT. In general, the schemes based on feature points and circular regions in transform domain are robust to geometric and signal processing distortions. The scheme presented by Manuel *et al.*[78] is depending upon SURF feature matching and DFT domain. The watermarking process is conducted as follows. The watermark is produced as 1-D binary pseudo-random pattern generated by a secret key. The image is then transformed into frequency domain by using DFT, and the magnitude of middle frequency components are selected for data hiding and detection. Finally, the feature points are extracted from watermarked image, stored in a file, and used to match with the feature points extracted from distorted watermarked image for gemometric correction.

Dang *et al.*[79] have proposed a watermarking scheme based on neural network and memetic optimization for color images. The authors select the luminance component Y of $YCbCr$ colour image for hiding a watermark. The component Y is decomposed by Symlet-2 DWT in the four-level wavelet transform, and only appropriate subbands are used for hiding data. The selected subbands are then divided into non-overlapping blocks. The relationship between wavelet coefficients and its neighborhoods in selected blocks are calculated by general

regression neural networks. The multi-objective memetic algorithm is used to optimally select the value of standard deviation for this network. Another scheme in frequency domain has been presented in[58]. The two-dimensional DWT is applied on the original image to obtain the middle frequency subbands whereas the one-dimensional DCT is applied to the selected middle frequency subbands to extract the final coefficients for hiding information. To enhance the quality of watermarked images, the genetic algorithm is analyzed to determine suitable positions for watermarking process. Then, the coefficients of middle frequency subbands are modified for hiding data. This method is able to withstand print-and-scan distortions. Zolotavkin *et al.*[59] have come up with a technique of Distortion Compensation (DC) for two dimensional Quincunx Lattice Quantization in which the parameters for controlling DC are modified to improve the efficiency of their scheme. The main steps of hiding procedure are briefly described as follows: applying a transform on the original image for obtaining a sequence of coefficients; performing a modification of 2D Quincunx to obtain pairs of quantized coefficients; replacing original coefficients with quantized coefficients to hide secret information; reversing the tranformed image to obtain a watermarked image in the spatial domain.

The scheme proposed by Haribabu *et al.*[80] uses auto encoder-based convolutional neural networks for learning features from positive image and negative image. The positive and negative image are generated from the original image, and the two generated images are then fed into two different networks. These two networks produce positive learned image and negative learned images. Depending on the watermark bit, the watermarked image is generated by adjusting pixel values which are picked up from either the positive learned image or the negative learned image. Munib *et al.*[60] have put forward another scheme based on triangular regions and Zernike moments. With the keypoints obtained by using Harris detector, Delaunay tessellation is then applied to partition image into distinct triangular segments. The magnitude of Zernike moments of each triangle is then used to hide a secret information. Maedeh *et al.*[61] have proposed a method based on DWT and DCT in which the image features like edge, saliency and intensity are used to compute hiding factors with the support of a fuzzy system. The authors use appropriate sub-bands of a 2D wavelet transform in two levels. Hiding data is carried out by modifying coefficients of frequency sub-bands, and the modification is based on the value of hiding factor.

2.1.4 Watermarking for trained neural networks

Recently, digital watermarking is also exploited to maintain the ownership authorization of deep neural networks. We have found few works relevant to this research direction as follows. Uchida *et al.*[34] have proposed a framework that enables to hide a watermark into a host network. There are three possibilities to hide a watermark into a network including training, fine-tuning and distilling process, and there are two distortions that can affect the accuracy ratio of watermark detection like fine-tune process and model compression (parameter pruning). The main idea is to hide watermark into one of the convolutional layers as depicted in Figure 2.10. To avoid impairing the performance of the host network, the authors utilize parameter regularizer instead of modifying parameters of a trained network. This method

can be applicable to other networks such as standard multilayer perceptron, recurrent neural network and long short-term memory. The hidden watermark can be detected in case of fine-tune process and model compression. Similarly, another scheme for neural networks has been proposed by Nagai *et al.*[35]. This method enables to hide information into trained networks through the process of training, fine-tuning and distilling network. For the training and fine-tuning process, the copyright holders of the network is expected to hide watermark into their network. For the distilling process, the non-copyright holder is entrusted to hide information into the network on behalf of a copyright holder. This method is capable of detecting hidden watermark from the trained model in case it is subjected to fine-tuning, model compression and watermark overwriting.

Group	Output size	Building block	Parameters
conv 1	32×32	$[3 \times 3, 16]$	N/A
conv 2	32×32	$\begin{bmatrix} 3 \times 3, 16 \times k \\ 3 \times 3, 16 \times k \end{bmatrix} \times N$	$144 \times k$
conv 3	16×16	$\begin{bmatrix} 3 \times 3, 32 \times k \\ 3 \times 3, 32 \times k \end{bmatrix} \times N$	$288 \times k$
conv 4	8×8	$\begin{bmatrix} 3 \times 3, 64 \times k \\ 3 \times 3, 64 \times k \end{bmatrix} \times N$	$576 \times k$
	1×1	avg-pool, fc, soft-max	N/A

Figure 2.10: *The illustration of network structure wherein the watermark is hidden into convolutional layers enclosed by blue rectangles.*

Another CNN-based method[63] hides secret bits by dividing the original image and watermark into non-overlapping blocks. The CNN's weight parameters are then modified during the training process for watermarking image. With this approach, each block is able to carry one watermark bit. Recently, CNN-based scheme[81] enables to hide a watermark image into appropriate sub-bands of DCT transform of image by modifying its corresponding coefficients. The hidden information is detected by making use of CNN network.

2.2 Steganography technique

As mentioned above, digital steganography is utilized for the purpose of covert communication. From our survey, we have found several existing works that are designed only for natural images. Different from digital watermarking, the most common important requirements for digital steganography are imperceptibility, capacity and security.

The most common algorithm used in designing steganography scheme is least significant bit (LSB) substitution[42]. With this method, each pixel value of cover image is converted into binary form, and hiding secret data into image is conducted by replacing a number of bits which is farthest to the right and holds the least value in a multi-bit binary number as depicted in Figure 2.11. Changing the value of these bits does not much affect the quality

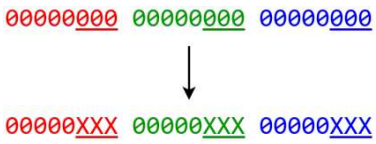


Figure 2.11: *Least significant bit (LSB) substitution.*

of stego-image. Chang *et al.*[82] have proposed a scheme based on Hamming code. The idea of this scheme is to hide a group of seven secret bits into a group of seven pixel values of the cover image. By doing so, the parity check matrix of the (7, 4) Hamming code is applied to classify 128 different bit strings of length seven into 16 groups whereas each group contains eight different bit strings of length seven. Hiding data is performed by replacing the least significant bit of a pixel with an appropriate bit string in the group.

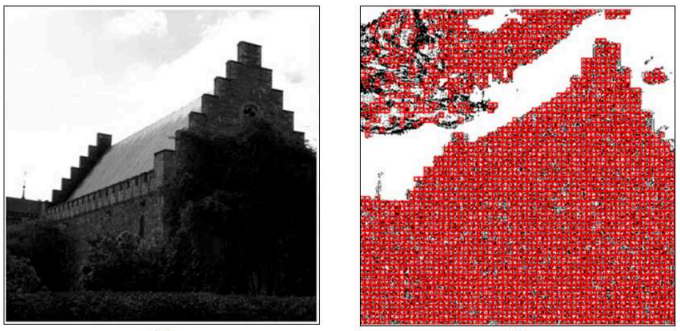


Figure 2.12: *An example of (a) cover image and (b) selected blocks at appropriate bit-plane.*

The method presented by Wang *et al.*[29] enables to hide four bits into a pixel value of the medical image by changing its four least significant bits. To enhance the security feature, a logistic mapping is used to generate pseudo random sequence and use it to scramble image before hiding secret information. Besides, to avoid losing important information in some regions of the medical images, these regions are ruled out before the hiding process begins. In[83], Nguyen *et al.* have proposed a scheme in which the secret information is hidden into multi bit-planes of an image. The cover image is first divided into non-overlapping blocks, and the complexity of the bit-planes of a block are then measured to identify which bit-plane of a block is used to carry secret bits. A block is only selected for carrying secret information when its bit-plane complexity satisfies a predefined condition as in Figure 2.12. Liu *et al.*[84] have introduced a scheme based on an extended turtle shell matrix in which a pair of non-overlapping pixel values (x_i, x_{i+1}) is used to carry one secret bit by referring to the constructed turtle shell matrix. The appearance of turtle shell matrix is depicted in Figure 2.13.

In[13], Soleymani *et al.* have proposed a method that enables to hide message scanned from a document in which the halftoning algorithm is used to convert the scanned document into binary image. To prevent hiding data into pixel values located in the smooth regions of cover image, the five most significant bit are used to compute standard deviation for filtering the concealable pixels, which are used to hide data, and preserving the quality of the stego-

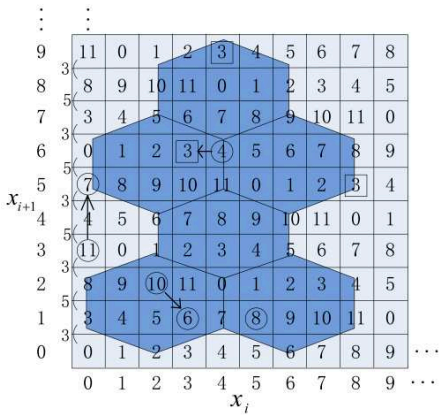


Figure 2.13: *An illustration of turtle shell matrix with the cycle of difference values (3, 5).*

image. The three least significant bits are then used to carry the message bits. Wang *et al.*[85] have introduced a scheme based on re-adjusted generalized exploiting modification direction (GEMD). The orginal GEMD algorithm is able to obtain a number of pixels for each group of pixels and maintain the hiding capacity of more than one bit per pixel. In this method, the authors adjust this algorithm to extend the hiding capacity up to two bits per pixel. The scheme presented in[86] is based on chaotic map in the DCT domain in which the authors select the alternating current (AC)-coefficients of the cover image for hiding secret information. To begin with, the cover image is partitioned into non-overlapping blocks, these blocks are then transformed into the frequency domain by applying DCT transformation. The AC coefficients are obtained by searching each transformed block in zigzag manner from the low significant DCT-coefficient to the most significant DCT-coefficient as depicted in Figure 2.14. Besides, the chaotic function is also utilized to enhance the security feature of the hiding algorithm.

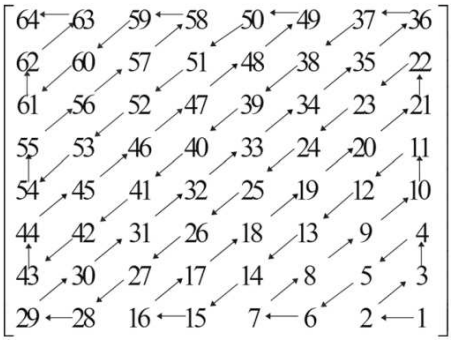


Figure 2.14: *Zigzag scanning in a block of size* 8×8.

Recently, deep neural networks-based schemes enable to hide a secret image into a cover image. Specifically, the scheme presented in[87] consists of three components as in Figure 2.15: (i) a preparation network is responsible for the preparation of the secret image. If the secret image is smaller than the cover image, this network increases the size of secret image to the size of the cover image. This network also transforms color-based pixels to more useful features; (ii) a hiding network takes RGB channels of the cover image and transformed

channels of secret image, and generates stego-image; and (*iii*) a reveal network is used to detect the secret image from the stego-image. Shi *et al.*[88] have come up with a scheme including one generative network and two discriminative networks. The former network is used to evaluate the visual quality of the generated images for steganography whereas the later networks are used to assess their suitableness for hiding information. Similarly, Volkhonskiy *et al.*[89] have proposed another scheme also based on three networks like one generative network and two discriminative networks. The generative network is utilized to produces realistic images from noise whereas the first discriminative network is used to classify whether an image is synthetic or real, and the second discriminative network is used to determine whether or not an image contains a concealed secret message. Another CNN-based scheme[90] has been proposed in which the authors make use of autoencoder neural network which is trained to generate an output image with the same size of the input image. Hiding or extracting secret information requires an estimation of parameters of the network, and this is obtained by using standard loss function (L_1) for weight optimization and variance losses.

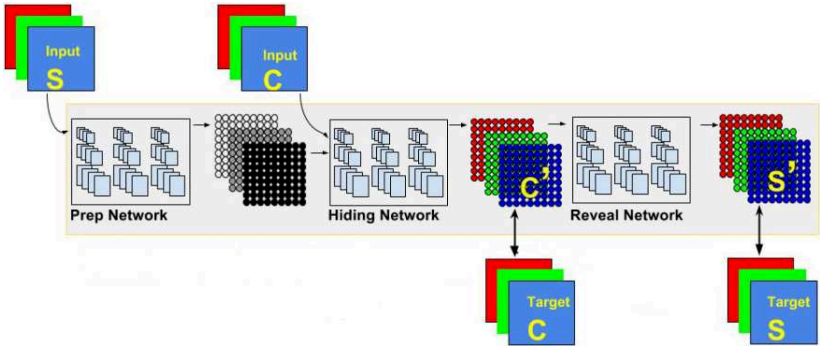


Figure 2.15: *Three networks for hiding and detecting secret image: S and C are the secret and cover image.*

2.3 Data hiding for binary images

Lu *et al.*[91] have proposed a data hiding scheme in the frequency domain by making use of direct current (DC) components of DCT. The approach is conducted by blurring a binary image to obtain a gray-level image. The blurred image is then divided into non-overlapping blocks with size of 8×8. Data is only hidden into non-uniform blocks whereas the uniform blocks (containing all white or black pixels) are eliminated because of imperceptibility. The DCT transformation is applied for the selected blocks, and then the coefficients of DC component are changed for carrying data. After hiding data, the gray-level image is converted into the binary form by using a dynamic threshold. The experimental results show that this method can resist to cropping and noises. Another scheme[92] enables to hide data by enforcing the odd-even feature of non-uniform blocks of size 8×8. The even number of black pixels in the block corresponds to bit 0 whereas the odd number corresponds to bit 1. Besides, the authors have employed a 2D shifting to provide security for their watermarking scheme.

Yang *et al.*[93] have proposed a scheme in which the image features are extracted by scanning through image by using a block B of predefined size, and a decomposition of non-interlaced blocks from B as depicted in Figure 2.16 is applied. A moving window is used to scan through each of non-interlaced blocks to identify the flippability of each pixel. Shuffling is also performed on the original image to increase the security of the scheme.

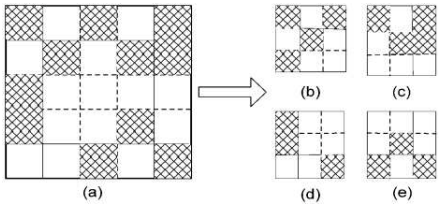


Figure 2.16: *An illustration of a block B (a) and a decomposition of its non-interlaced blocks (b), (c), (d) and (e).*

The method based on dots has been presented in[94] wherein the dots are inserted into and randomly distributed over the whole document for hiding data. The authors have proposed "dots-image" for data hiding and detection, this image contains tiny dots which are inserted into the watermarked document as in Figure 2.17(b). The "dots-image" is also divided into four quadrants, and each quadrant containing four data bearing pixels marked as $0, 1, 2, 3$ in Figure 2.17(a) is used to hide one data bit. To enhance the accuracy ratio of data detection, the authors divide document into four quadrants, and the same data is hidden into each quadrant. This method is robust to print-and-scan, print-photocopy-scan distortions and affine transformation. However, the distortion caused by the hiding process is visually perceptible.

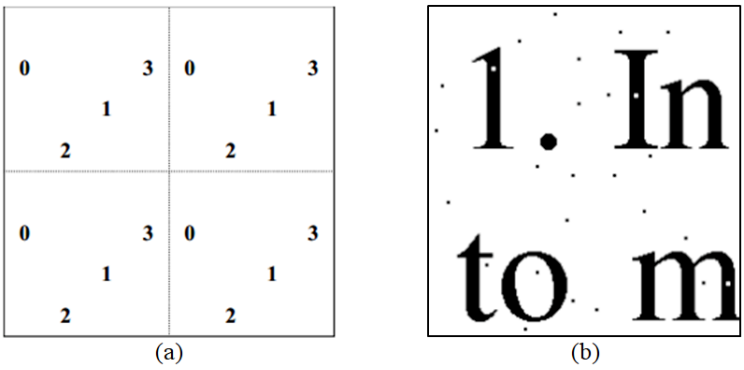


Figure 2.17: *An illustration of: (a) a "dots-image" with four quadrants where the dots at positions of $0, 1, 2, 3$ are flipped to carry one data bit; and (b) a watermarked document.*

Yang and Kot[14] have put forward an interesting method in which the embedding process is conducted by flipping a center pixel value of appropriate 3×3 blocks and preserving the connectivity of its corresponding neighbors. To identify the potential blocks depicted in Figure 2.18 for hiding data, the authors have defined rules such as: the number of uniform white transition and the number of uniform black transitions along vertical and horizontal directions; the number of the interior right angle transitions; and the number of transitions

from the center pixel to the sharp corners. The odd-even feature of the number of white and black pixels is considered for these blocks. Besides, the security feature of the scheme is also mentioned in this method.

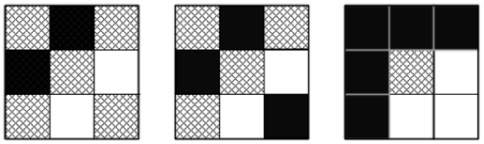


Figure 2.18: *A sample of potential blocks used for hiding data.*

A multi-level signature-based scheme[95] is effectively used to detect malicious tampering. The signature employed in this scheme is generated by utilizing the interlaced morphological binary wavelet transform in which the coefficients (LL) obtained from computing the odd-odd transform are used to hide information, and the coefficients (LL, HL, LH) obtained from computing the even-even transform are employed to generate signature. Lee *et al.*[96] have proposed another method to select flippable pixels with less distortion, and this method is based on edge line segment similarity (ELSS) measure. The ELSS is estimated to represent the degree of visual distortion. The authors use a block of size 3×3 for carrying information bit, and only block with appropriate ELSS value is selected to hide data. Instead of hiding data bits sequentially, the order of data bits is randomly suffled prior to being hidden into the document for security enhancement, and the recovery of extracted data requires a symmetric key. Edge adaptive grid-based scheme[15;97] enables to select pixel locations for carrying information better than methods based on the block. The authors have proposed a content adaptive process illustrated in Figure 2.19, which is used to trace new contour segments and look for new pixel locations for carrying data.

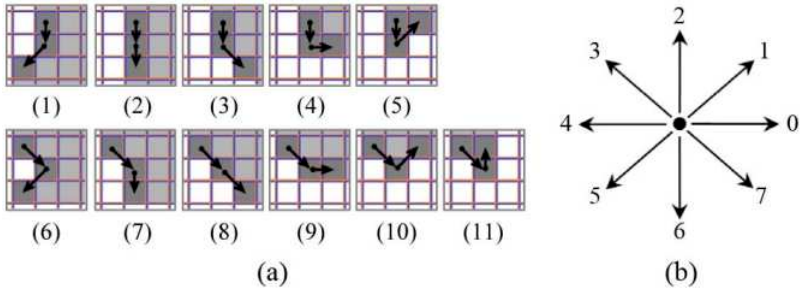


Figure 2.19: *(a) Several types of contour segment, and (b) change code used for transition determination.*

Wang *et al.*[98] have come up with two kinds of block patterns (including 14 variances) of size 2×2 and two types of matching pair methods for their scheme. The block patterns are used to improve hiding capacity whereas the matching pair methods including external and internal adjustment are employed to reduce changes in the image, which are occurred during the information hiding process. To enhance the security feature, the authors permute the set of predefined block patterns by using a secret key which is only authorized for legal users. Li *et al.*[99] have proposed a watermarking system in which the authors use scaling interpolation prior to flipping pixels for hiding secret data. Flipping pixel values is based on

the score of connectivity and the smoothness of a 3×3 pattern, and keeps a high structural similarity. This method obtains good image quality and capacity. Jung and Yoo[100] have proposed a scheme based upon key pairs depicted in Figure 2.20. These keys play a critical role in deciding potential hiding positions and determining whether or not it is possible to carry secret information bits. This method obtains good visual quality of the images after hiding secret information. Another watermarking scheme relied upon fractal codes has been proposed by Daraee and Mozaffari[101]. The secret information is hidden into the fractal code of selected range segments. The decoding process of fractal code is applied to produce the watermarked image. This approach is capable to be robust against common distortions.

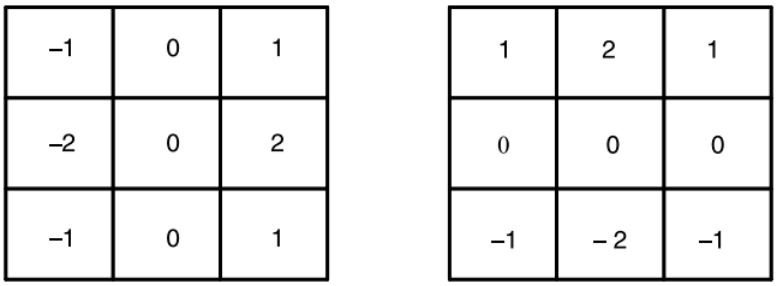


Figure 2.20: *An example of key pairs generated from a block of size* 3×3.

Recently, an additive model and sampling-based watermarking scheme has been proposed by Hou *et al.*[102] wherein the input image is splitted into multiple thumbnails by using a perimeter expansion and a sampling operation. The binary image perimeter expansion is applied on the original image prior to conducting the sampling operation because of avoiding devoid of external white margin surrounding the thumbnail images. The secret data is hidden into the appropriate thumbnails by adjusting the number of black pixels. The watermarked image is obtained by inversely sampling these thumbnails. This method is likely robust against print-and-scan distortion. Nguyen *et al.*[16] have put forward a data hiding method based on block classification in which the image is partitioned into non-overlapping blocks of size 3×3, and the complexity is calculated for each block. The value of complexity is used to determine if a block is a complex region or a smooth region. The complex regions are appropriately selected for carrying secret data bits. For each complex block, the center pixel is kept unchanged whereas eight neighboring pixels are used to hide a number of secret bits. To improve the security feature, the center pixel of each block can be selected by using a secret key. This approach is able to achieve high capacity. However, a threshold for determining the appropriate blocks needs to be estimated for a specific image, and the robustness against distortions is not mentioned.

2.4 Steganalysis

This technique aims at revealing the presence of a secret information which is hidden within the digital media. Liu *et al.*[103] have proposed a scheme based on feature mining and pattern classification for detection of LSB matching steganography in grayscale images. The authors

have proposed five types of feature such as: shape parameter of generalized Gaussian distribution (GGD) in the wavelet domain to measure the image complexity, entropy of the histogram of the nearest neighbors, the high-order statistics of the histogram of the nearest neighbors, probabilities of the equal neighbors, and correlations features. Besides, there are several learning classifiers such as naive Bayes classifier, support vector machines (SVM), quadratic Bayes normal classifier, and adaboost, which are applied on these features to determine whether there is a secret information hidden within an image. Gaussian-Neuron CNN-based approach[104] has been proposed for capturing the complex dependencies, which are useful for detecting the presence of hidden information. The authors have pointed out that with the conventional approach, the feature extraction and classification steps are separately performed. Hence, the classification step cannot refer to useful information extracted from the extraction step. Their network consists of three kinds of layers: an image processing layer, several convolutional layers for feature representation, and several fully connected layers for classification. The image processing layer contains filtering operation with a predefined high-pass filter, which aims to strengthen the weak stego image and reduce the impact of image content. To better distinguish a stego image from its cover image, the authors use Gaussian activation function for their network.

Another CNN-based method[105] is similar to the work presented in[104] in which the authors have conducted several experiments to find out an appropriate architecture of CNN, and their network is greater in height than in deep and without pooling layer. Qian *et al.*[106] have put forth a method based on CNN. This network consists of one image processing layer, five convolutional layers for feature representation, and three fully connected layers for classification. The role of the image processing layer is similar to the one presented in[104]. In addition, the authors have proposed a cropping strategy which enables the CNN network to deal with arbitrary input image sizes. A hybrid deep-learning-based method[107] consists of two stages in which the first stage takes decompressed JPEG images as input, and it corresponds to the convolution phase, and the quantization and truncation phase of the rich models. Meanwhile, the second stage is a compound deep CNN network in which the network parameters are learned in the training procedure, and it is composed of three subnets with identical structure. Each subnet corresponds to one group of quantized and truncated residual maps.

Wu *et al.*[108] have proposed a method using CNN with shared normalization layer. The authors have shown that the CNN network with multiple batch normalization layers is hard to be generalized to the test data once the cover images and their stego images are not paired in a test batch. The shared normalization shares the same statistics for all training and test batches. The network consists of three sub-networks such as preprocessing, feature learning and classification. The preprocessing network is used to extract the high frequency component from input cover and stego image. It contains high pass filtering layer and truncation layer. The feature learning network is used to extract effective features for image steganalysis. The classification network is used to map the extracted features into binary labels. An unsupervised method[109] has been proposed in a combination of artificial training sets and supervised classification. The artificial training set is formed by applying the targeted steganography algorithm to the testing data, and it is used to find a boundary

between classes with remarkable accuracy. The approach is to transform the cover images into images that belong to the class of stego images. The classification technique can have a relevant impact in the manner that the steganalysis system is approached. Thus, it allows to classify the images without a real training set. The classifier used in this approach is SVM with a Gaussian kernel.

The method based on wider separate-then-reunion network[110] has been proposed for color images. In this method, the authors replace summation in normal convolution with channel-wise convolution in the bottom convolutional layer. However, in the upper convolutional layers the authors use normal convolution which retains summation and makes them remarkably wider. This network takes a color image as an input and applies channel-wise convolution to the red, green and blue channel of the input image.

2.5 Forgery detection

Apart from visibly or invisibly embedding a secret information into the documents or images for tracing their origin, determining whether an image is genuine or manipulated can also be performed by the techniques of forgery detection. These techniques are the essential objectives of image forensics.

Image forgery is divided into two categories like active approach and passive approach. For the active approach, the concept of digital watermarking or digital signature or combination of them is used in which the detector knows secret information that the image contains. For the passive approach, the tampering is detected in a way the detector requires no prior information about the original image, digital signature or digital watermark, instead the specific algorithms are designed to determine the tampered regions. The passive approach is divided into three categories: copy-move forgery, image splicing and image retouching. Copy-move forgery technique is used to hide some sensitive or important information into image by copying and pasting a portion of image such that no one can easily recognize whether an image is original or forged. Detecting copy-move forgery is based on either keypoint or block. Image splicing is created by a combination of two or more images, and it is classified into two types such as region based and boundary based. Image retouching technique is to make small change in color, background, rotation, scaling, etc. for generating a forged image. The typical techniques used to detect forgery are presented below.

A new region duplication detection method[27] has been proposed to detect duplicated and distorted regions in an image in which SIFT is utilized to extract keypoints and acquire image features from the extracted keypoints. The authors formulate region duplication detection as finding transformed identical regions in an image, and they use robust estimation to obtain the matching of correct keypoints and transforms between duplicated regions simultaneously. With the estimated transforms, the method is able to obtain precise location of duplicated regions. A two-stage feature detection[28] has been put forth to obtain better feature coverage and enhance the matching performance. The feature point detection consists of two stages for extracting feature points for common areas rather than small smooth regions, which

guarantee sufficient feature point coverage all over the image while maintaining a moderate feature point number. This is done by performing a non-maximal suppression with an appropriate radius. To extract feature for ordinary regions, the authors have decided to choose local feature descriptor as multi-support region order-based gradient histogram (MROGH) which avoids the dominant orientation assignment. To extract feature for small smooth regions, the authors construct a fused feature by combining the MROGH and hue histogram so that the discriminability of the obtained descriptor can be enhanced. The descriptor is then used to find identical regions.

Wang *et al.*[29] have come up with a keypoint-based method for detecting move-copy forgery in small smooth regions. To do so, the tampered image is partitioned into non-overalpping and irregular superpixels in which the superpixels are classified into smooth, texture and strong texture based on local information entropy. The keypoints are then extracted from each superpixel by utilizing the superpixel content-based adaptive feature point detector. For each keypoint, the local visual features are constructed, and the best bin and generalized 2 nearest-neighbor algorithm are employed to find the matching keypoints. Finally, the duplicated regions are determined by using zero mean normalized cross-correlation measure. A passive scaling robust algorithm[30] has been proposed by using normalized cross correlation. This method detects highly correlated regions from the image and image blocks. The normalized cross correlation is used as a metric to assess to the level of dissimilarity or similarity between two digital images. To detect tampering regions, the image is segmented into blocks. Coarse regions of the tampering are detected based on the computation of the correlation matrix. Each of detected coarse blocks is divided into small blocks, and the normalized cross correlation is carried out on the corresponding coarse block to determine the tampering.

Another SIFT-based method[31] has been proposed with the improvement of matching operation such that the method is capable of detecting matching pairs located in duplicated regions. The keypoints and their descriptors are extracted from the image by utilizing SIFT. An improvement of matching operation is performed to handle both single and multiple copy-move forgeries in which a verification algorithm based on SIFT scale space representation has been proposed to select a subset of matched pairs. This subset is used to estimate geometric transformation, and the duplicated regions are localized by using the estimated transformation. Prakash *et al.*[32] have proposed a method for detecting splicing and move-copy forgery in which the authors combine block discrete cosine transform and polar Zernike moment. To extract the features of a color image, the authors utilize advanced threshold method in chroma space. A combination of block discrete cosine transform and de-correlation are first applied to reduce the influence caused by the diversity of the image content. The enhanced threshold method based on Markov random process is then applied to extract the discriminative feature for forgery detection. SVM with radial basis function (RBF) kernel as classifier and k-fold cross validation is then used for detection of forgery.

A local features-based method[33] has been proposed to localize any kind of tampering in image obtained from the social network. This is performed by detecting the inconsistencies between two images. The analyzed image is compared to the most similar images retrieved

by a content-based image retrieval (CBIR) system. Once similar images are retrieved, a local descriptor-based approach is used to identify and localize differences. The authors have chosen the 7^{th} fully connected layer of the VGG-19 CNN trained on ImageNet to build descriptors. Besides, dense SURF features are extracted from images, and RANSAC algorithm is then applied to estimate the affine transformation between the two images. This estimation enables to filter the false positives returned by the CBIR. This method is able to be robust to various transformations such as rotation, illumination changes, crop and translation.

2.6 Evaluation of feature point stability

In the scope of this research, we take advantage of pattern recognition techniques to develop data hiding systems in the spatial domain for the purpose of securing document images. From our survey, we have seen that both conventional techniques and deep learning have been utilized in developing digital watermarking and steganography. Moreover, we have also found that a diversity of these techniques is applied for designing data hiding schemes wherein several schemes are dedicated to natural images, and fewer schemes for document images. Thus, we would like to evaluate the performance of the feature points, which are effectively applied for natural images, extracted by using the conventional pattern recognition techniques when applying for document images. From which we find out suitable approaches that can be used to improve the limitations of the existing methods designed for documents as mentioned in Section 1.3. The stability of feature points is presented through the following evaluations.

2.6.1 Construction of hiding regions

Apart from using feature points to design circular regions or disks and transforming them to the frequency domain for watermarking development as presented in Section 2.1.3, the feature points are also used to construct hiding regions[2] for data hiding system in the spatial domain. To do so, the feature points are used to construct non-overlapping square regions centered at their corresponding feature points in which the unnecessary feature points are eliminated based on their stability and the size of the hiding region. The hiding regions of size $B \times B$ within an image of size $M \times N$ are depicted as in Figure 2.21 where $M_i (i = 1, ..., 6)$ represents the center of a hiding region. There are a number of well-known detectors utilized to detect the feature points from the natural images such as SIFT[111], Harris[112], Harris-Laplace (HL)[113], Laplacian of Gaussian (LoG)[114], maximally stable extremal regions (MSER)[115], speeded up robust features (SURF)[116], binary robust invariant scalable keypoints (BRISK)[117] and active contour[118]. The brief explanation of these detectors is presented as below.

- SIFT: This technique enables to detect keypoints and to compute their local descriptors by using Gaussian scale space. The algorithm is conducted by smoothing and resizing

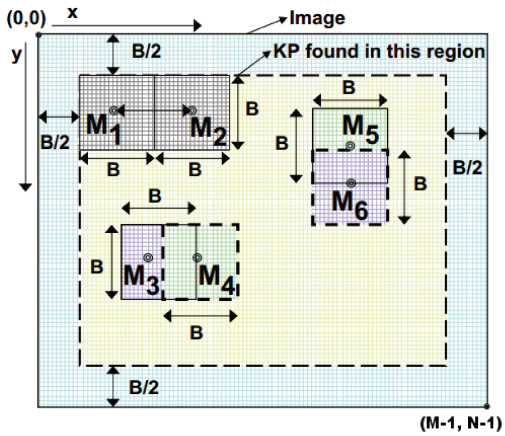


Figure 2.21: *The illustration of $B \times B$ hiding regions[2], which are constructed by using feature points.*

the input image, and takes the local extrema of the difference of Gaussian functions in the three dimensions of pixel coordinates and scales. The keypoints are then obtained from the extrema in a DoG (Difference of Gaussian) scale space in which each keypoint is assigned an orientation defined by the histogram of local gradient of the image intensity. The stability of a feature point will be determined by an intensity value in the scale space of DoG.

- Harris: The feature points are gained by computing the Hessian of the auto-correlation matrix in the gradient domain for every pixel of the image, and comparing the eigen-values and then removing flat areas and edges based on the relative magnitudes of the eigen values.

- HL: The scale adapted Harris function is used to localize points in a scale space, and the point is selected with a characteristic scale which is the extremum of the Laplacian over different scales.

- LoG: It is used for blob detection wherein the number of detected feature points is equal to a pre-assigned number of blobs.

- MSER: The obtained extremal region is the boundary pixels which have a higher or lower intensity than the pixels inside the region. The centroids of these extremal regions are regarded as feature points.

- SURF approximates the DoG with box filters. The main advantage of this technique compared to SIFT is the fast computation of operators by using the box filters. This technique enables to detect keypoints and to compute their local descriptors by using Hessian matrix approximation. The process of feature extraction is conducted by computing: integral image, Hessian matrix-based interest points and scale space representation. Integral image is an effective way of calculating the sum of pixel values in an image. The Hessian matrix is used in this approach because it gives high performance in terms of computation time and accuracy. The scale space is obtained by

scaling the filter size instead of reducing the image size. Meanwhile, the process of feature description is conducted by assigning orientation for the interest points and constructing a square region for extracting the descriptor.

- BRISK: This technique enables to detect keypoints and to compute their local descriptors. It consists of the following steps: detecting scale space keypoint, filtering keypoint, assigning orientation to the keypoint and generating descriptor. The process of detecting the keypoint is relied upon pyramid scale space which is constructed by downsampling the image into octaves and intra-octaves. The candidate keypoints are determined in the image pyramid by using features from accelerated segment test (FAST).

- Active contour is based on snakes and level sets, and it formalizes the problem as an energy minimization. It works by minimising an energy that is in part defined by the image and part by the shape of spline such as length and smoothness. The minimization is done implicitly in the shape energy and explicitly in the image energy.

In the context of data hiding, the stability of feature points is measured by the robustness of feature points-based hiding regions against distortions. Compared to the hiding regions constructed from the undistorted document, a hiding region constructed from the distorted document could fall into one of the possible cases: (i) it is extracted at the same position with the same order in a set of extracted hiding regions; (ii) it is moved to another position; (iii) it is vanished and replaced by a new hiding region. The last two cases of (ii) and (iii) are known as mismatched hiding regions. Let R_{ureg} be the set of watermarking regions constructed from the undistorted document, and R_{dreg} be the set of hiding regions constructed from the distorted document. The matching proportion P_m is estimated by:

$$P_m = \frac{\left|R_{ureg} \cap R_{dreg}\right|}{\left|R_{ureg}\right|} \tag{2.1}$$

where $\cap$ is the intersection operator, $|A|$ is the number of elements in set A.

For natural images, the works presented in[2] use various distortions including: D1- Gamma noise with $\gamma = 1.5$, D2- Gamma noise with $\gamma = 0.6$, D3- add white Gaussian noise with signal-to-noise ratio of 25 dB, D4- geometric distortions (rotation of 30° + scaling of 0.75), D5- Gaussian blur at $\sigma = 2$, D6- JPEG compression with quality factor of 10, D7- unsharp masking of 3×3 , D8- median filtering of 5×5. The average value of matching proportion P_m on 250 natural images suffered from various distortions is presented in Figure 2.22.

For document images, we evaluate the stability of feature points extracted by SIFT, SURF, BRISK, MSER, and active contour detector. The datasets used for this evaluation consist of Tobacco[119], L3iDocCopies[120] and DSSE-200[121]. The kind of noise caused by JPEG compression is selected in the assessment of feature point stability because the compression of document with low quality factor leads to significant change of document pixel intensities and much affects the outcome of feature point detection. The JPEG compression algorithm utilizes the block-based DCT for image quantization, and the image discontinuities appear at

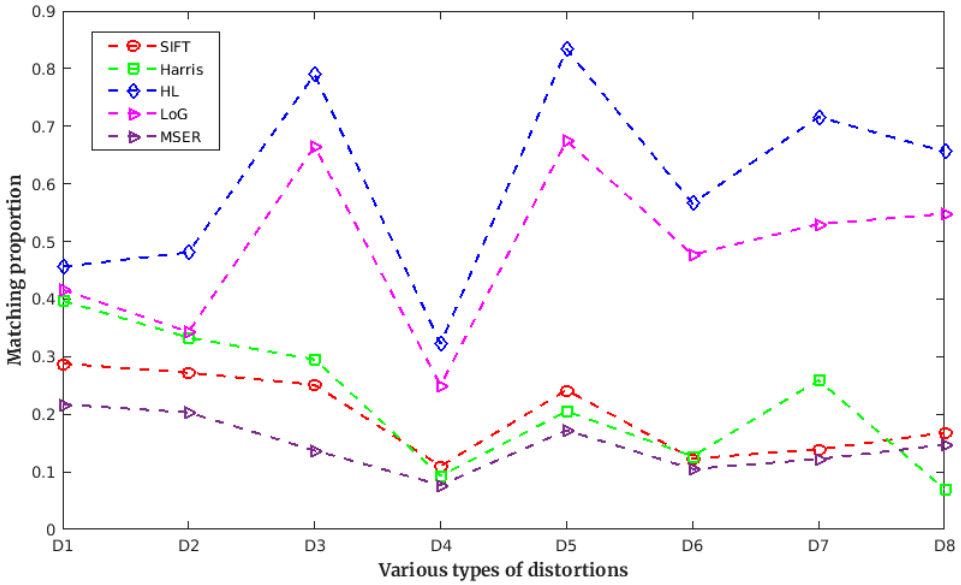


Figure 2.22: *The average value of matching proportion of hiding regions extracted from standard grayscale test images by using various feature point detectors.*

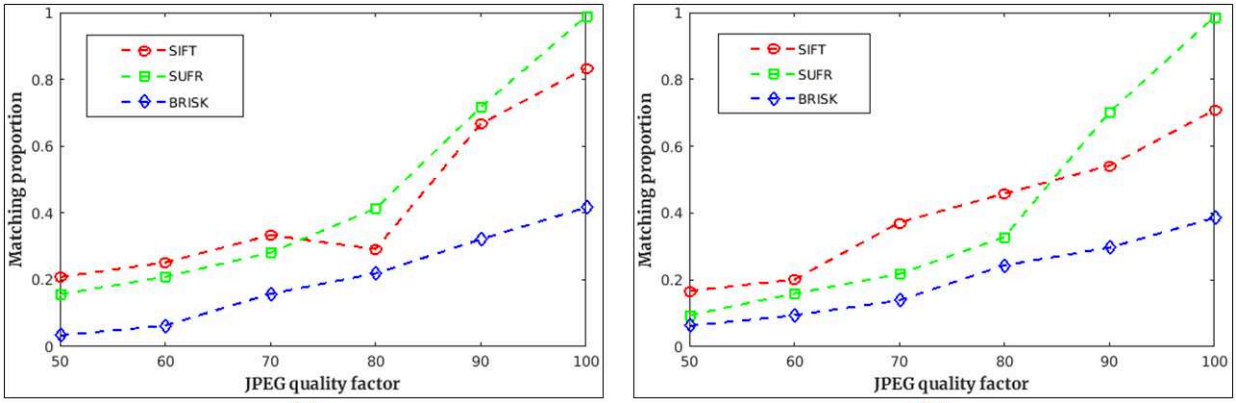


Figure 2.23: *The average result of matching proportion of hiding regions extracted from documents of (a) Tobacco and (b) L3iDocCopies dataset.*

the boundaries of each block. Thus, the sharpness of edge is reduced due to the suppression of high frequency coefficients. The average result of matching proportion P_m on 20 documents from each dataset is shown in Figure 2.23 wherein (a) for Tobacco and (b) for L3iDocCopies. Besides, we also assess the stability of feature points extracted by SURF, MSER and active contour on 60 documents from DSSE-200. The distortion used in this assessment is also JPEG compression. The average results are shown in Figure 2.24.

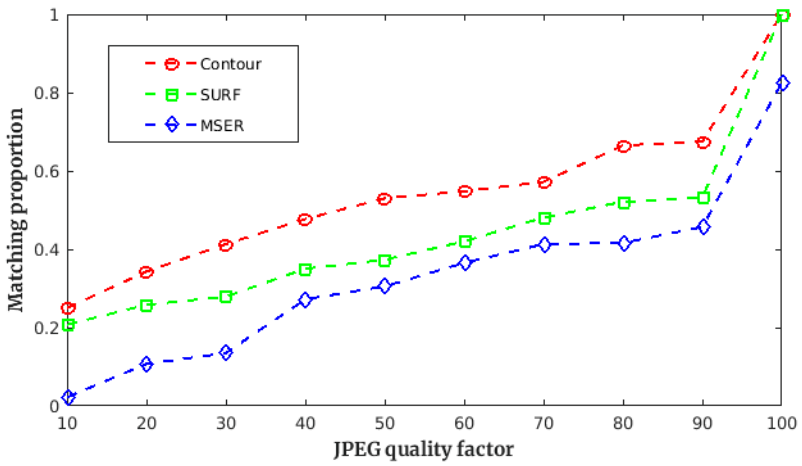


Figure 2.24: *The average result of matching proportion of hiding regions extracted from documents of DSSE-200 dataset.*

2.6.2 Image normalization

Aside from construction of hiding regions, the feature points are also employed to normalize images, this means that the input image is transformed into a standard form with the objective to obtain invariance with respect to geometric distortion. The works presented in[3;122;123] demonstrate two ways to identify parameters for image standardization including translation, rotation and scaling wherein one is based on two most stable feature points, and the other is relied upon three most stable feature points. The image normalization is briefly described as follows.

For the method based on two feature points, given a set of feature points obtained from a specific detector, these feature points are sorted according to their response values, and two feature points with the highest response value are taken for computing the normalization parameters. Regarding translation parameter, translating the input image (x, y) to the center point of a normalized image (x_c, y_c) is conducted by:

$$(t_x, t_y)^T = (x_c - x, y_c - y)^T \tag{2.2}$$

The rotation angle θ is estimated based on the line segment $\overline{P_0P_1}$ obtained from such two selected feature points as $P_0(x_0, y_0)$ and $P_1(x_1, y_1)$:

$$\theta = tan^{-1}\frac{y_1 - y_0}{x_1 - x_0} \tag{2.3}$$

Let d be the Euclidean distance between P_0 and P_1, and c be a predefined constant, the scaling factor s is calculated as a ratio between c and d:

$$s = \frac{c}{d} \tag{2.4}$$

Finally, a pixel value at a point (x, y) of the original image is transformed into its corresponding value at a point (x', y') of the standard document by:

$$(x', y') = T(t_x, t_y) + sR(\theta)(x, y) \tag{2.5}$$

where T and R are translation and rotation operations respectively.

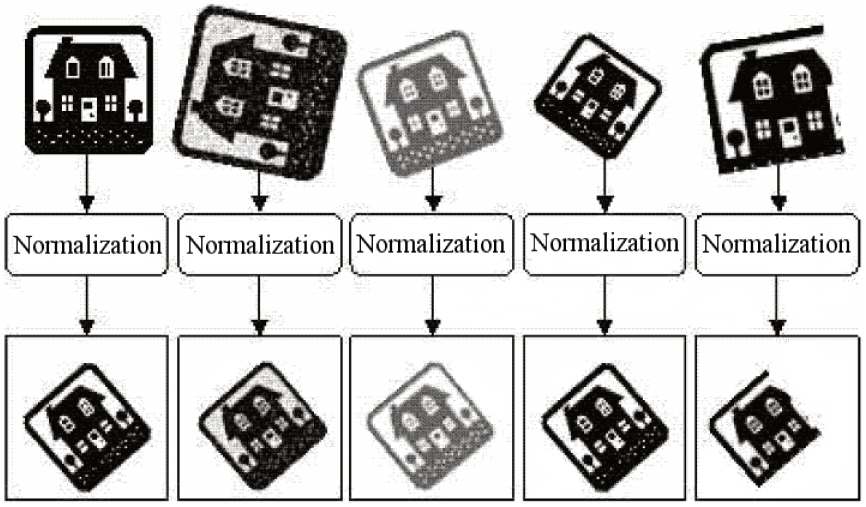
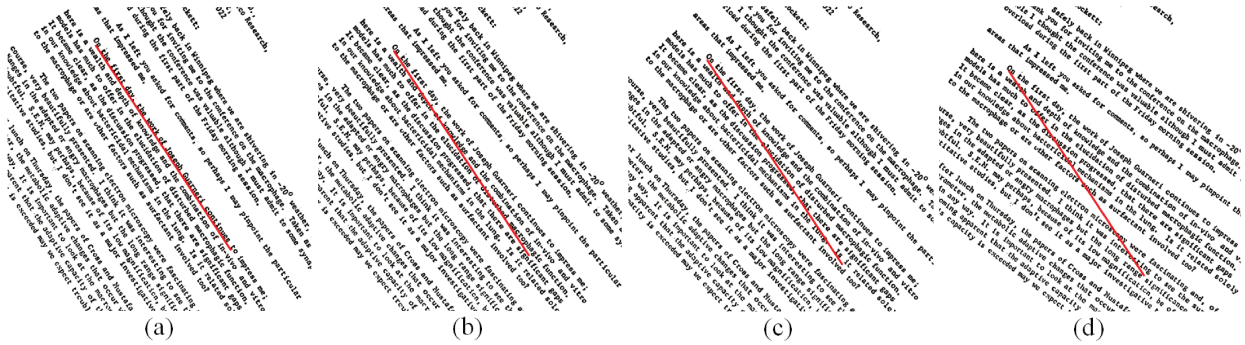


Figure 2.25: *Illustration of transforming geometrically distorted images into standard form*[3].

For the method based on three feature points, similar to the way of selecting two feature points, the three appropriate feature points (P_0, P_1, P_2) are also selected from a set of extracted feature points for estimating the normalization parameters, and these feature points form a triangle. Let the incenter point P_c of the triangle be an intersection of the three angle bisectors, the point P_l be the coordinates of one of the three feature points P_0, P_1 and P_2 (the point of triangle whose angle value is the largest), and d_l be the longest length of a side among the three sides of the triangle. The rotation angle is then calculated based on the line segment $\overline{P_c P_l}$ obtained from these two points and the horizontal line, and scaling parameter is estimated as a ratio between a predefined constant c and d_l. With these parameters, transforming an input image into a normalized form is conducted similar to Equation 2.5. The demonstration of image normalization is depicted as in Figure 2.25.

Although the feature points-based normalization approach is relatively effective for natural images, it is less effective when applying for document images. The three most stable feature points extracted from the original document do not match with the three most stable feature points extracted from geometrically distorted document. We have implemented this technique for documents and obtained results as shown in Figure 2.26 in which the red line demonstrates the normalized direction which is parallel with the text line of normalized doc-



(a) (b) (c) (d)

Figure 2.26: *Document normalization based on three most stable feature points: (a) is a normalized form of a document without geometric distortion. (b), (c) and (d) are normalized forms of documents with rotation of 15, 25 and -35 degrees respectively.*

ument without geometric distortion. Apparently, we can see that the normalized direction of geometrically distorted documents (with a rotation of 15, 25 and -35 degrees) is inconsistent with the direction of the undistorted document.

2.7 Summary

In this chapter, we have presented a thorough review of various techniques that can be used to protect the genuine images or tracing the image origin. Depending on these reviews and combined with the problems set out in Chapter 1, we have observed that the pattern recognition-based data hiding schemes designed for document images should be further investigated. Besides, the evaluation of applications based on the feature points extracted from conventional techniques is also a motivation for us to propose more efficient data hiding schemes. The details of our data hiding schemes are presented in the next chapters.

Chapter 3

Securing document images via conventional approaches

This chapter introduces feature points-based steganography scheme and stable regions-based watermarking scheme. In the former, we use SURF detector to extract feature points for constructing hiding regions, local binary pattern and local ternary pattern for determining potential hiding positions. Hiding data into document is based on odd and even feature of gray level values, and we also utilize error correction code to enhance the accuracy of hidden data detection. Besides, we also present another proposed detector that can be applicable to detect more stable feature points in the context of distorted document images. In the later, we combine common image processing operations and non-subsampled contourlet transform to detect stable hiding regions. The correction of geometric distortion is also integrated into this scheme to enhance its performance. Meanwhile, the watermarking algorithm is developed depending on each group of consecutive pixel values assigned with weights in which we have eliminated the pixel values that are vulnerable to distortions.

3.1 Introduction

As discussed in Section 2.1.3 and Section 2.6, the feature points extracted from natural images have been widely used to develop data hiding system in the transform domain. Taking advantage of this idea, we develop a digital steganography in the spatial domain for the purpose of securing document images. The feature points-based steganography scheme is the beginning of our work, which deals with the security issue of legal documents. With the evaluation of feature point stability in Section 2.6, we can see that the feature points are less robust in case of distorted images and documents. Thus, another stable feature point detector, which is relied upon non-subsampled contourlet transform and distance transform, has been introduced to improve the precision of detecting the hidden information. Although the new feature point detector has slightly improved the stability of the feature points extracted from the distorted documents, the scheme still has not met the property of robustness in

order that it can be applied in the environment of real distortions caused by print-and-scan process, print-photocopying-scan process and so on. Because of more focusing on the robustness of a data hiding system designed for securing document images, so we have decided to replace steganography with watermarking, which is presented from Section 3.3 onwards. For this reason, the watermarking scheme has been developed in which the document features used to develop this scheme are stable regions instead of the feature points. As mentioned in Chapter 1, there are two main challenges in designing a data hiding system including the features extracted from document is sufficiently stable, and the algorithm for data hiding and detection is robust enough when the watermarked documents are suffered from distortions. The solutions to solve these challanges will be in turn presented in the remaining sections of this thesis.

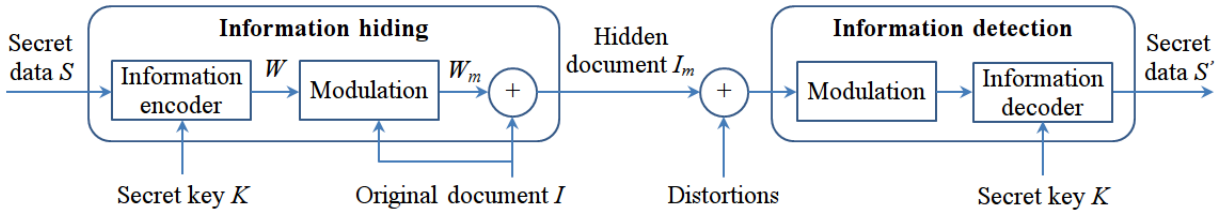


Figure 3.1: *The components of a blind data hiding framework.*

The general data hiding framework proposed in our research is presented in Figure 3.1. For information hiding process, we need to provide a legal document I, a secret information S used for document verification, a secret key K to encode the secret information or to generate ramdom positions in the document for security enhancement, a method to detect document features and a hiding algorithm, which are used to generate a stego or watermarked document. The block "Modulation" refers to the detection of document features and hiding algorithm. For the process of information detection, the system requires the stego or watermarked document I_m, the secret key K, the method to detect document featues and data detection algorithm, which are employed to extract the hidden information S'. The detail of detecting document features and the algorithms of data hiding and detection will be presented in this chapter and next chapters.

3.2 Steganography scheme based on feature points

Steganography is an effective way to hide a secret information into a document image with the objective of providing authenticity of transmitted documents. We introduce a novel data hiding scheme that enables to embed a secret information with a moderate length in the document by taking advantage of pattern recognition techniques. The main functions of this method consist of: (i) the potential feature points used for constructing the hiding regions are identified by using speed up robust features (SURF) detector; (ii) local binary pattern (LBP) is utilized to figure out content regions where the pixel values can be changed to hide data, and local ternary pattern (LTP) are then effectively exploited to locate the possible hiding positions within the content regions where the secret information bits are hidden

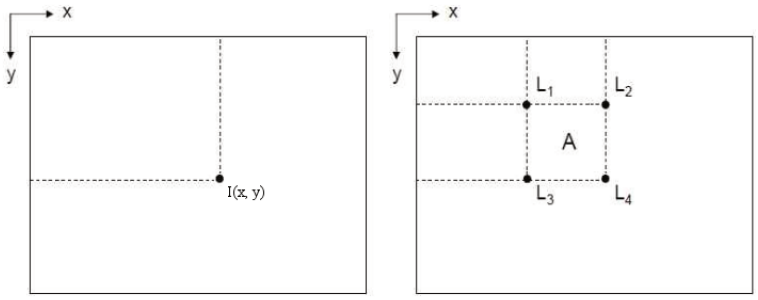
into; (*iii*) to make the scheme robust against rotation caused by distortion of printing and scanning process, Hough transform[124] is applied to compute the rotation angle for restoring a rotated document to the direction of its original document. Besides, the repetition code and other improved methods are also implemented to possibly enhance the precision of secret data detection.

The brief explanation of techniques used to extract document features is presented as follows.

SURF has been proposed as a fast and robust algorithm in comparison with SIFT. The SURF algorithm consists of three main processes including feature point detection, repeatable angle calculation and feature description. This technique depends on a scale-space representation, associated with differential operators such as first and second order box filters. The process of detecting the feature points is relied upon Hessian matrix approximation. Extracting the feature points is conducted by obtaining the determinant of the Hessian matrix and extracting the local maxima. To do that, the integral image is applied to calculate the sum of pixel values in a given image. This technique is also used to compute the average intensity within a given image. By definition, the entry of an integral image $I_{\sum}$ at a location (x, y) represents the sum of all pixels in the input image I within a rectangular region formed by the origin of an image and the location (x, y). Given an image I with size of $m \times n$, the integral image $I_{\sum}$ can be calculated by:

$$I_{\sum}(x, y) = \sum_{i=0}^{x} \sum_{j=0}^{y} I(i, j) \tag{3.1}$$

It can be shown that once the integral image has been computed, it only requires three additions to calculate the sum of the intensities over any upright, rectangular area as shown in Figure 3.2, regardless of the size of the region.



(a) (b)

Figure 3.2: *The representation of integral image: (a) is the integral image. The region A in (b) is computed by* $L_4 + L_1 - (L_2 + L_3)$.

To obtain good performance in terms of computation time and accuracy, the SUFR algorithm utilizes the Hessian matrix. Instead of using different measures for determining the location and the scale which are based on Hessian-Laplace detector, this technique is

depending on the determinant of the Hessian matrix for selecting the location and the scale. To adapt for any scale, the image are filtered by using a Gaussian kernel, so given a point $p(x, y)$ from the original image I, the Hessian matrix $H(p, \sigma)$ of a point p belonging to scale σ is defined as follows:

$$H(p, \sigma) = \begin{bmatrix} L_{xx}(p, \sigma) & L_{xy}(p, \sigma) \\ L_{xy}(p, \sigma) & L_{yy}(p, \sigma) \end{bmatrix} \tag{3.2}$$

where $L_{xx}(p, \sigma)$ is the convolution of the Gaussian second order derivative with the image I in point p, and σ is the standard variance of the Gaussian, and similarly for $L_{xy}(p, \sigma)$ and $L_{yy}(p, \sigma)$.

The extreme values are obtained by using Hessian matrix, and the stable location and scale of a feature point are obtained by utilizing interpolation operator. The approximated Hessian matrix $\hat{H}(p, \sigma)$ is defined by:

$$\hat{H}(p, \sigma) = \begin{bmatrix} D_{xx}(p, \sigma) & D_{xy}(p, \sigma) \\ D_{xy}(p, \sigma) & D_{yy}(p, \sigma) \end{bmatrix} \tag{3.3}$$

where $D_{xx}(p, \sigma)$, $D_{xy}(p, \sigma)$ and $D_{yy}(p, \sigma)$ are the convolutions of the approximated filters with image I.

Once the feature points and their scales are obtained, each feature point is assigned a repeatable angle before obtaining the invariant descriptor vector. The angle of the gradients around the feature point is calculated, and the maximum angular response is selected as the direction of the feature point. This process depends on the calculation of Haar wavelet responses. The obtained direction is then applied to generate a rotated square surrounding the feature point. The gradients within this square are then combined to form the final invariant descriptor vector.

LBP has been initially proposed for characterizing the spatial structure of a texture, which is invariant to transformations of intensities or color. Due to its high performance, LBP has become a widely used operator for image processing in real application. By applying LBP to an image, each pixel value is depicted by an integer label which is robust to monotonic illumination change. There are 256 different labels in a 3×3 neighborhood, and each of these labels is regarded as a LBP pattern. The LBP pattern for a pixel is computed by comparing neighboring pixels with the center pixel in terms of their intensities. The neighboring pixels whose intensities are larger than the central pixel are assigned to 1 while the other intensities are assigned to 0. The comparison will result in a bit string with eight elements. The binary weights, which come from a geometric sequence, are set to the bits corresponding to their positions in the bit string. The bit string associated with its weights is then converted into a decimal valued index. A pixel c with gray level value g_c is labeled by:

$$S(g_p - g_c) = \begin{cases} 1, & \text{if } g_p \geq g_c \\ 0, & \text{if } g_p < g_c \end{cases} \tag{3.4}$$

where pixels p belong to a 3×3 neighborhood with gray levels g_p

The LBP pattern of the 3×3 neighborhood is computed by summing the corresponding values $S(g_p - g_c)$ weighted by a binomial factor of 2^i:

$$LBP = \sum_{i=0}^{7} S(g_p - g_c).2^i \tag{3.5}$$

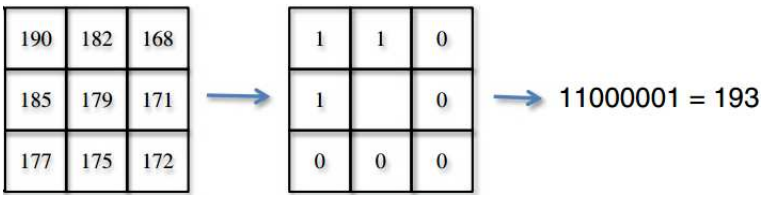


Figure 3.3: *The illustration of LBP computation for a pixel.*

After computing the labeling for each pixel of the image, a 256-bin histogram of the resulting labels is used as a feature descriptor for the texture. The illustration of LBP computation for a pixel is depicted in Figure 3.3.

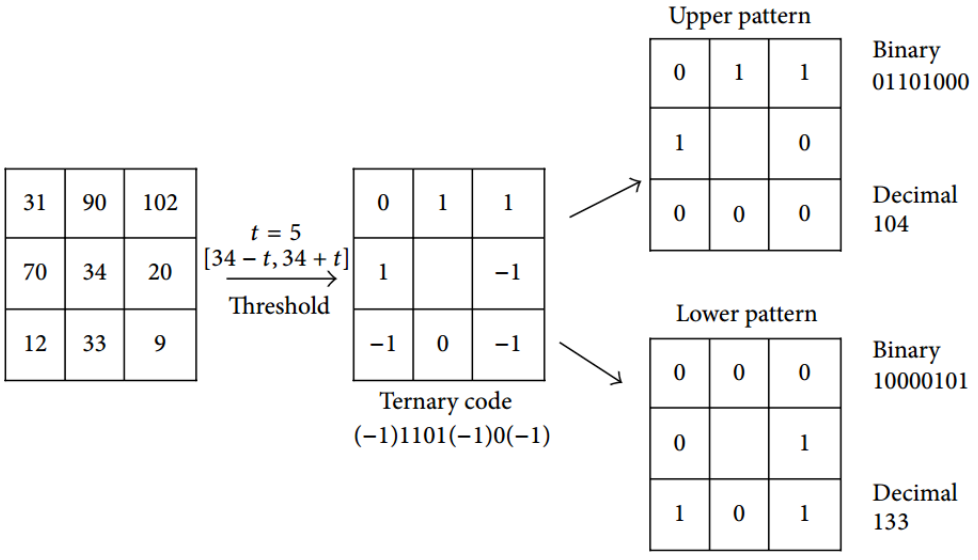


Figure 3.4: *The illustration of LTP computation for a pixel.*

LTP is an extension of LBP to a three-valued code $(-1, 0, 1)$, which deals with the noise sensitivity issue of LBP. The LTP pattern for a pixel is labeled by using a threshold function

around zero to evaluate the local grayscale difference. The gray level values in the zone of width t (predefined threshold) are assigned to 0, the pixels whose intensities above the threshold t are assigned to 1, and those whose intensities below t are assigned to -1. A pixel c with gray level value g_c is labeled by:

$$S(g_p - g_c) = \begin{cases} 1, & \text{if } g_p \geq g_c + t \\ 0, & \text{if } -t < |g_p - g_c| < t \\ -1, & \text{if } g_p \leq g_c - t \end{cases} \tag{3.6}$$

where g_p is a pixel value in neighborhood, g_c is the center pixel value.

The LTP pattern can be decomposed into two binary patterns which are upper and lower patterns, and two binary patterns are converted into two decimal values and replaced back at the center pixels similar to LBP. The LTP operator is the concatenation of the code of the upper pattern and the lower pattern. The illustration of LTP computation for a pixel is depicted in Figure 3.4.

Hough transform maps a line in the spatial domain to a point in the Hough parametric space, and it is used to find lines, curves or parametric curves. The simplest case of Hough transform is the linear transform for detecting straight lines. The slope intercept model of a straight line is defined by:

$$y = mx + c \tag{3.7}$$

where m is the slope, and c is the y intercept.

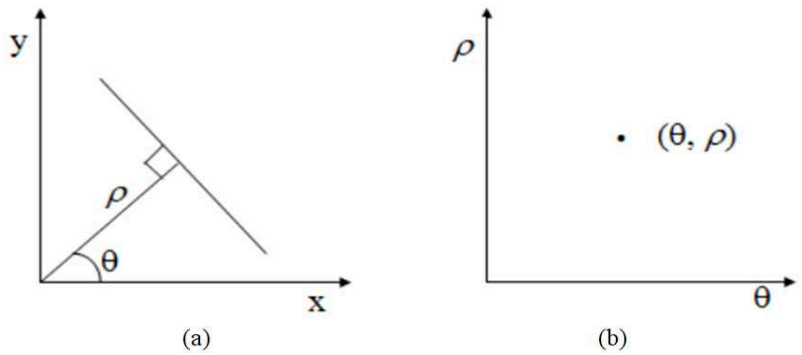


Figure 3.5: *Hough transform: (a) is the image space, and (b) is the parametric space.*

The straight line can be written in the form of parametric space by:

$$\rho = xcos\theta + ysin\theta \tag{3.8}$$

where ρ is the distance of line from origin, and θ is the angle of ρ with respect to x axis.

The line in the image space is just a point in a parameter space as depicted in Figure 3.5. Hough transform uses two dimentional arrays which are regarded as accumulator array for detecting the presence of lines in image space where each row and column correspond to θ and ρ values respectively. Peak point is the strong point in the accumulator array which represents straight line in the image space.

3.2.1 Pattern analysis for hiding patterns and hiding positions

In this section, we present how to identify content regions by using LBP and to determine hiding position within content regions by utilizing LTP. Hiding information into documents (based on pixel level approach) is simply the process of changing pixel intensities such that it has to keep the quality of the stego or watermarked documents, and this change has to be imperceptible to visual perception. In order to keep the imperceptibility, we need to find out the suitable features of document where any small change does not significantly affect the visibility of a document. For this reason, with the support of LBP, the corner and non-uniform patterns are considered as potential features that can be used to identify the regions of document content for data hiding because changing pixel values in these regions are less sensitive in terms of human visual perception. We consider the corner and non-uniform patterns as hiding patterns. Meanwhile, the edge and uniform patterns are eliminated from this context because changing pixel intensities in this regions is easily noticeable and could draw much attention by human perception. By taking advantages of LBP as mentioned in Section 3.2, we can easily identify such document features as corner, edge, uniform and non-uniform features. To hide much information bits, we decide to choose the hiding pattern with size of 3×3, and then the document features are determined by considering the neighboring pixels of a pattern where corner or nonuniform features come under. It is noticeable that the number of secret information bits that be hidden into a document depends on the characteristic of its content.

Depending upon the binary code of a LBP pattern, the document features are determined as follows. The dark corner features consist of three successive bits 1 and five successive bits 0. The bright corner features consist of five successive bits 1 and three successive bits 0 whereas the edge features include four successive bits 1 and four successive bits 0. To determine whether a pattern is uniform or non-uniform, an uniformity measure of a pattern is defined as the number of bitwise transitions from 0 to 1 or vice versa. A local binary pattern is called uniform if its uniformity measure is at most 2 (e.g. 00000000 - 0 transition, 11111111 - 0 transition, 01110000 - 2 transitions), otherwise it is called non-uniform pattern (e.g. 11001001 - 4 transitions, 01010011 - 6 transitions). The dark and bright corner patterns are depicted in Figure 3.6 in which (a) is dark corner features, and the extracted dark corner features are depicted in (b). Similarly, the bright corner features are depicted in (c) and (d).

After using hiding patterns to identify appropriate content regions for data hiding, LTP is then applied on these regions to locate which positions are relevant for carrying the concealed message bits. LTP produces 3-valued code $(-1, 0, 1)$ depending on a predefined and fixed threshold t. In the context of our work, the gray values corresponding to valued-code "1"

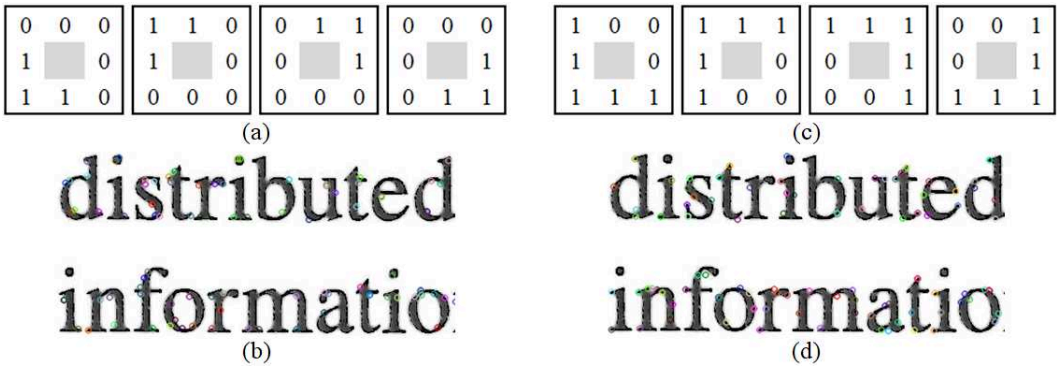


Figure 3.6: *Hiding patterns: (a) and (b) are dark corner features. (c) and (d) are bright corner features.*

in each of LTP patterns are selected for carrying the secret bits. With this method, the number of bits that can be hidden into a 3×3 pattern reaches five bits. For instance, the LTP pattern as illustrated in Figure 3.7(c) enables to change pixel intensities at appropriate positions (corresponding to code 1 of LTP) to hide four secret bits. If we replace the gray level value of 232 corresponding to code "0" of LTP with 235, the LTP then enables to change the gray level values to hide five secret bits. The relationship between pixel intensities, LBP and LTP is shown in Figure 3.7 in which (a) is a sample of pixel intensities from undistorted document, (b) and (c) are LBP binary code and LTP code.

236	223	221
238	232	228
242	238	232

(a)- Gray values

1	0	0
1		0
1	1	1

(b)- LBP code

1	-1	-1
1		-1
1	1	0

(c)- LTP code *(t = 3)*

Figure 3.7: *The demonstration of identifying hiding region and positions.*

By applying corner and non-uniform features to find out the hiding patterns, and LTP to determine the hiding positions as described in Figure 3.6, we have encountered lots of hiding patterns and hiding positions which are not robust against pixel intensity changes caused by distortions, e.g. JPEG compression. Based on our experiments, we have seen that once the pixel values of document changed, there are two cases leading to loss of integrity in identifying the positions of pixels whose gray values are adjusted to hide the secret data including: LBP binary code generated from undistorted document is disrupted compared to LBP binary code generated from distorted document; the hiding positions corresponding to LTP code are inconsistent because of the fixed threshold t. Figure 3.8 demonstrates a sample of pixel values from distorted document (a), the corresponding LBP code (b) and LTP code (c). Apparently, we can see that the LBP code and LTP code generated from distorted document is inconsistent with the ones generated from undistorted document. This inconsistency will affect the robustness of data hiding scheme.

235	223	220
237	233	228
242	238	232

(a)- Gray values changed

1	0	0
1		0
1	1	0

(b)- LBP code

0	-1	-1
1		-1
1	1	0

(c)- LTP code *(t = 3)*

Figure 3.8: *The illustration of hiding region and positions from distorted pixel intensities.*

To reduce the inconsistency of LBP and LTP code generated between undistorted and distorted document, we choose the median binary pattern (MBP), which is a variation of LBP, as an alternative method. For LTP, instead of using a fixed threshold for whole document, we estimate a dynamic threshold t_i for each of LTP patterns by:

$$t_i = c \times m_i \tag{3.9}$$

where c is a predefined constant, m_i is the median value of all gray values of each LTP pattern.

235	223	220
237	233	228
242	238	232

(a)- Gray values changed

1	0	0
1		0
1	1	1

(b)- MBP (mean = 232)

1	-1	-1
1		-1
1	1	0

(c)- LTP *(t = 0.01x232 = 2)*

Figure 3.9: *The illustration of hiding region and positions from the distorted pixel intensities using MBP and dynamic threshold.*

Figure 3.9 shows that the MBP and dynamic threshold have improved the consistency of detecting the hiding region and positions from the distorted pixel intensities.

3.2.2 Rotation correction using Hough transform

To meet the requirement of robustness against real distortions, by observation, we have seen that the scanned documents are more or less suffered from geometric distortions including rotation and scaling. The rotated document will result in considerably reducing the performace of data hiding scheme. Thus, the correction of rotated document partly improves the scheme performance. To address geometric distortions, in the spatial domain, feature points-based method[3;122;123] enables to transform natural image into a standard form in which the direction of the standard image is rotated with a certain angle instead of vertical direction. This method has been proved to provide good performance for natural image, and it gives less efficiency when applying on document images because the appropriate feature

points obtained to estimate parameters for geometric correction are inconsistent between the distorted and undistorted document. The evaluation of this method for natural images and documents is also presented in Section 2.6.2. Therefore, we have decided to utilize Hough transform to deal with the rotated documents, and this technique is very effective to detect text lines in the documents, which are used to indentify the rotation angle.

With the Hough transform, a point in the original (x, y) image space is mapped to all points in the (ρ, θ) parameter space of lines through (x, y) in which ρ is the distance of a line from origin, and θ is the angle of ρ with respect to x axis. This technique computes the values for the parameters (ρ, θ) of all curves of straight lines that can pass through each black pixel of a document. Votes are then casted for each curve in an accumulator matrix. Each dimension of this matrix corresponds to one of the parameters. After the entire document has been processed, the accumulator matrix is inspected for local maxima. Each such maximum indicates the existence of a curve in the original document given by the corresponding parameter values on the axes. The rotation angle is then determined by (ρ_i, θ_i) corresponding to the maximum values in the accumulator matrix.

With the rotation angle θ, an affine transformation is applied to the rotated document to obtain the corrected document by:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} cos\theta & sin\theta & 0 \\ -sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{3.10}$$

where (x', y') is a point of the corrected document, (x, y) is a point of the rotated document, and $\begin{pmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{pmatrix}$ is the rotation transformation matrix.

With this approach, we can correctly estimate the rotation angle in case of documents rotated within $[-90°, +90°]$. Figure 3.10 demonstrates the estimation of rotation angle with the help of Hough transform technique in which the estimated rotation angle is -14.9951 and 20.0027 corresponding to -15 and 20 degrees, and the appropriate blue line is used to estimate the rotation angle.

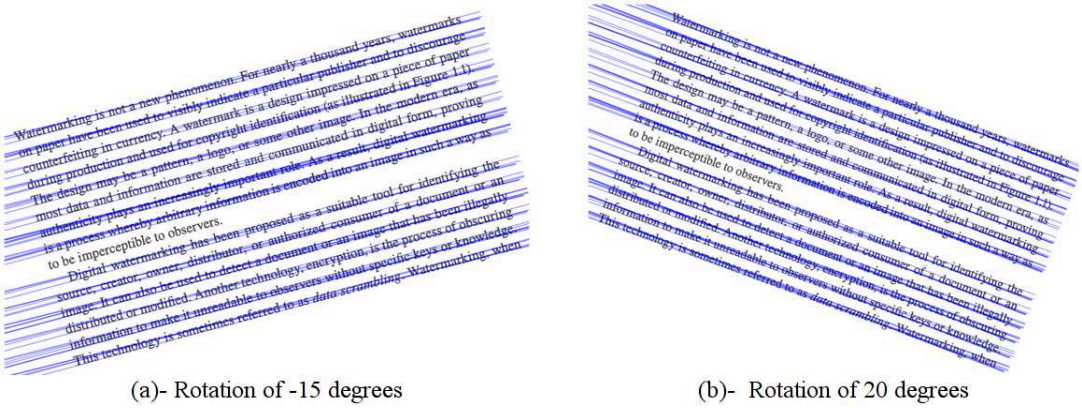


Figure 3.10: *Hough lines used for estimation of rotation angle.*

3.2.3 Data hiding process

The data hiding scheme in our research is designed to hide a meaningful text information into the documents, so this information is converted into a series of binary digit as secret bits. The hiding process is conducted based on the following main steps:

Step 1 - Extracting feature points from grayscale document

Given an input document I size of $M \times N$. A set of feature points $F = \{p_i(x_i, y_i)\}_{i=1}^{n}$ is extracted by using SURF detector, where n is the number of feature points. We sort them in descending order based on their response value, and we then obtain a sorted list. Here, we prioritize to hide the secret information into regions around the feature points with strong reponse first and in turn to the feature points with weaker response because the higher the response value is, the more the stability obtains.

Step 2 - Constructing hiding regions and eliminating overlapping regions

The secret bits can not be hidden by directly changing the pixel values at positions corresponding to the feature points because this results in losing the consistency of feature poins extracted from the original document and document whose pixel values changed at positions of feature points. This is why we need to construct hiding regions surrounding the appropriate feature points. The hiding region B with size of $L \times L$ as illustrated in Figure 3.11 is centered at an appropriate feature point, and the size of hiding regions determines the performance of our scheme. If L is too small, lots of feature points detected from the original document could mismatch with the feature points detected from the stego document. Otherwise, the number of hiding regions will be insuffcient, and this results in having less data to be hidden. To make sure the border of the hiding region B is not overflowed outside the document, a feature point $p_i(x_i, y_i)$ is appropriately opted if and only if $\lceil L/2 \rceil \leq x_i \leq (N - \lceil L/2 \rceil)$ and $\lceil L/2 \rceil \leq y_i \leq (M - \lceil L/2 \rceil)$.

SURF extracts a lot of feature points that densely cover the whole document content. Thus, the hiding regions centered at these feature points are definitely overlapped each other. To apply feature points for constructing the hiding regions, the feature points making the overlapping hiding regions should be eliminated because if data is hidden into the overlapping regions, it will results in robustness reduction. To eliminate the feature points making the overlapping regions, we begin with the first feature point in the sorted list, the distance d between every pair of feature points is measured by Euclidean distance. If d is less than L, one of these two feature points has to be removed out of the list. In this case, we prioritize to keep the feature point with higher response value. This process is iterated until reaching the end of the sorted list. Eventually, the feature points remained in the list are ready to construct the hiding regions.

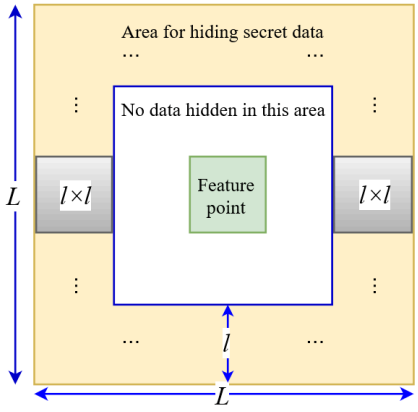


Figure 3.11: *The hiding region $B(L \times L)$ contains corner patterns of $l \times l$.*

Step 3 - Hiding secret information into the document

As mentioned above, the secret information is converted into a sequence of bits denoted as $W = \{w_1, w_2, ..., w_m | w_i \in \{0, 1\}\}$, where m is the length of a secret information. The message bits are sequentially hidden into possible positions of the 3×3 hiding patterns, which are located inside the hiding regions B by:

$$p'_i = p_i - p_i \, mod \, 2 \; + \; w_i \tag{3.11}$$

where w_i is the i^{th} secret bit, p_i and p'_i are the gray level values in the original and stego document respectively.

Although the robustness against lossy compression has been implemented by using MBP instead of LBP, and dynamic threshold for LTP, the accuracy for detecting the hidden information bits is still not as high as expected. Thus, we have employed error correction code in our steganography scheme. Correcting the corrupted bits can be performed by various ways such as implementing each error correction code separately, or combining a type of correction code with another one. Basically, the technique of error correction code enables to add extra data to the transmitted information, and the extra data help detection of error and reconstruction of original information. In this work, we apply repetition code on the secret bits by repeating each bit three times, which allows us to correct one error in each group of three bits. This will generate a new sequence of bits with lots of repeated bits. The generated sequence of bits is then hidden into the document.

Besides, we also improve the hiding method as presented in Equation 3.11. Changing pixel intensities in a range $[-1, 1]$ for data hiding scheme is possibly considered as the best way for imperceptibility because the quality of stego document is least affected. However, this range is easily destroyed by lossy compression algorithm where the gray level values are much changed even with the highest quality factor. For this reason, we extend this range to $[-4, 4]$ for adjusting the pixel intensities, the wider range of changing gray values for data hiding is able to raise the robustness of scheme against distortions. With addition or subtraction of 4 pixel units, the adjusted gray level values is hardly perceived by the human

visual system whereas the quality of stego documents still remains at an acceptable level.

3.2.4 Data detection process

Before detecting the hidden information, the stego document is first restored to its correct form as described in Section 3.2.2 for enhancing the accuracy of data detection because the stego documents may be subjected to rotated distortion caused by the printing and scanning process. The secret information bits w_i' are simply extracted by:

$$w_i' = p_i' \, mod \, 2 \tag{3.12}$$

The accuracy ratio of extracted secret bits is measured by:

$$StegoR = \frac{\sum_{i=1}^{m} \neg(W(w_i) \oplus \tilde{W}(w_i))}{m} \tag{3.13}$$

where $W(w_i)$ and $\tilde{W}(w_i)$ are the i^{th} secret information bit corresponding to original and extracted bit, m is the length of a secret information, $\neg$ depicts the NOT operator and $\oplus$ denotes the exclusive-OR operator.

The detected message bits are divided into groups of three bits, and the secret information is recorvered by extracting bit 0 or 1 from each group based on the occurrence of bit 0 and 1 within the group.

3.2.5 Improvement of feature point detection

As discussed in Section 2.6, the well-known feature extraction methods such as SIFT and SURF give good performance even when images contain distortions like view point change, blur and rotation[125]. However, their performance declines significantly when images are compressed using the discrete Cosine transform-based algorithm, especially when they are applied on documents. It means that the number of feature points extracted from uncompressed images do not match with the feature points extracted from compressed images. Hence, the detectors give too many false feature points to design a robust data hiding scheme. For this reason, to improve our hiding scheme, we propose a new feature points detector which is more stable than the existing methods in the context of detecting feature points for documents. The distorted document images caused by JPEG compression could fall into the following possible cases[126]: (*i*) For pure texture components, the distortion can blur the texture of dark regions and make an appearance of some new blocks or effects; (*ii*) For edge components, the weak edges in the document image may be distorted by this noise; (*iii*) For edge components in the texture regions, some edges are lost, some new edges are inserted to the document. These cases will lead to change significantly the intensities of the document and certainly cause instability of feature points. To reduce these distortions which

affect the feature point detection, the proposed detector is operated in the following steps:

Step 1 - Eliminating weak components of document

The document image is first converted to binary form, and opening morphological operator is then applied on binary document to remove small regions. This task is to eliminate the content regions of document which are sensitive to intensity changes.

Step 2 - Normalization of generated binary document

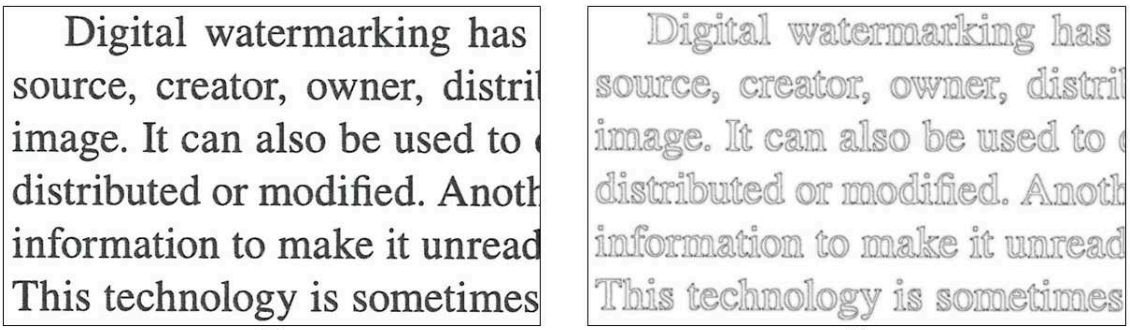


Figure 3.12: *Illustration of normalization: The original document (a) and normalized document (b).*

The binary document is transformed into an intermediate form by utilizing contour detection. The methods of transformation are often used in feature detection consisting of skeleton and contour in which skeletonizing document image will preserve the connectivity of object regions, but reduce most of foreground intensities. In addition, skeleton is also sensitive to boundary deformation. In contrast to skeleton, contours are more stable in the presence of noises regardless of image categogy[127]. Hence, the contourlet transformation[128] is applied on the binary document, and this results in a contourlet document. The contourlet document is considered as a normalized document. The normalized document is illustrated in Figure 3.12(b).

Step 3 - Detection of local maxima

We apply distance transform[129] on the normalized document which contains boundary intensities. The distance transform is often used to improve the performance of feature detection. The result of this transform generates a graylevel document in which every intensity is assigned by a value corresponding to the distance $L2$ which is nearest to foreground objects. The regions of interest of documents are depicted as the foreground objects O. Let p_1 be a given pixel of the document, and p_2 be a pixel belonging to the object O. The distance transform $dt(p_1, p_2)$ is calculated by:

$$dt(p_1, p_2) = \begin{cases} 0 & \text{if } p_1 \in O \\ min_{p_2 \in O} d(p_1, p_2) & \text{if } p_1 \notin O \end{cases} \tag{3.14}$$

where $d(p_1, p_2)$ is the Euclidean distance between p_1 and p_2.

Next, we find the local maxima in the transformed grayscale document by using a kernel with size of $k \times k$. The factor k determines the number of feature points that will be extracted. The larger k is, the fewer the feature points are obtained.

Step 4 - Computation of feature points

The approach of filtering feature points using scale space filtering[130] is applied in this step. This produces consecutive blurred documents by using Gaussian filter, which is defined by:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x+y)^2/2\sigma^2} \tag{3.15}$$

where σ is the smoothing factor which controls the scale, and x and y are pixel coordinates.

The filtered documents are also binarized in order to generate documents which are used for feature point detection. The extracted feature points are depicted as small color circles in Figure 3.13(b).

Figure 3.13: *Detection of feature points: The distance transform document (a) and extracted feature points (b).*

Besides, we also demonstrate the feature points which are extracted from well-known detectors such as SIFT, SURF and BRISK as shown in Figure 3.14(a), (b) and (c) respectively.

In this section, we have introduced a steganography scheme for document authentication, which is based on the feature points extracted by using the well-known SURF detector. By experiments, we have observed that the stability of the feature points has significantly been mitigated when the documents undergone distortions. To overcome this, we have proposed

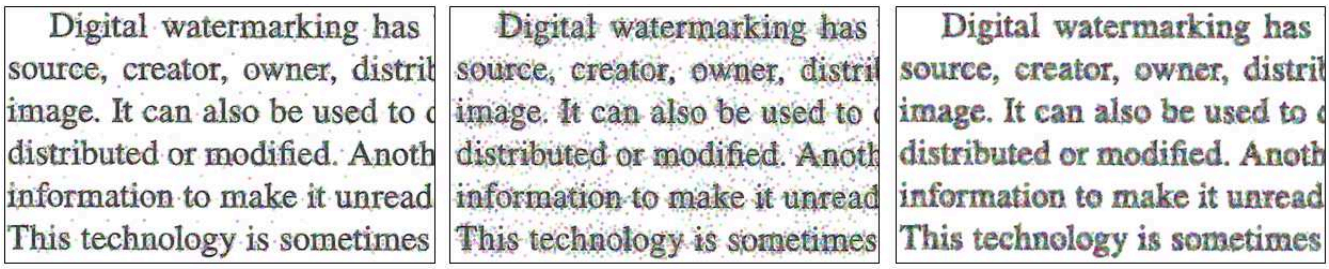


Figure 3.14: *Feature points extracted from (a) SIFT, (b) SURF and (c) BRISK detector.*

another detector for feature point extraction, which provides better stability in terms of distortions. In addition, the data hiding and detection processes have been developed by making use of LBP and LTP. The robustness of the scheme has also been improved by utilizing MBP and the dynamic threshold used for LTP. The experimental results of this approach are detailed in Section 5.2.

However, the performance of the scheme still has not met the requirement of high robustness to be applied for real applications. It means that the scheme has ability to detect the hidden information from the documents which are subject to high distortions such as JPEG compression with low quality factor, print-and-scan operations, print-photocopy-scan operations, etc. Thus, the feature stability extracted from the documents and the robustness of the scheme have to be further investigated and improved to meet these requirements. The scheme improvement is presented in the next section.

3.3 Watermarking scheme based on stable regions and object fill

Depending on the performance of feature points-based steganography scheme as presented in Section 3.2, we have seen that it is quite difficult to extract feature points for developing data hiding scheme in spatial domain, which is robust enough to practical distortions such as printing and scanning noises, print-photocopy-scan noises. Thus, we have proposed to extract stable regions from the documents for developing data hiding scheme instead of feature points. In our research, we wish to develop a framework being able to resist to various practical noises, so the robustness of scheme is prioritized to deal with. According to the characteristics of a data hiding system as discussed in Section 1.2, the robustness property is always concentrated in designing a watermarking system.

In this section, we introduce a watermarking system in which the stable regions are extracted from documents by making use of image processing operations and non-subsampled contourlet transform (NSCT)[65]. The watermarking algorithm is developed based on a group of successive pixel intensities within the stable regions.

3.3.1 Stable region detection

By observation, we have seen that the extracted bounding boxes surrounding document content are not highly robust to some distortions when applying NSCT directly on document images. Thus, the input document needs to be transformed into another form such that it is less affected by noises as much as possible. Figure 3.15 depicts the general steps for detecting document regions which are likely stable against noises.

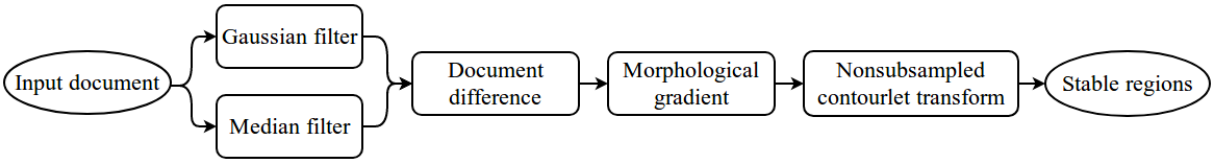


Figure 3.15: *Steps of detecting stable regions.*

First, we apply two filtering operations to remove noises. Gaussian filter (GF) is known to reduce noise and straighten the edges of the document content, but this operator makes document blurred. Meanwhile, median filter (MF) is known to reduce noise and does not blur document. After removing noises, the difference between GF and MF is made by subtracting each other to locate the positions in the document that fluctuate in intensity change due to the distortions. To enhance variations of intensity pixel in the document difference, morphological gradient operator is applied. These variations correspond to the edges of the document content. It detects either the internal or external boundary of an edge. The document produced at this step is considered as a transformed document.

Next, we take advantage of the properties of non-subsampled contourlet transform such as translation invariant, multiscale, multidirection and anisotropy to identify contour from the transformed document. NSCT first decomposes document into several pyramidal levels ranging from finer to coarser scale and different directions with the same size of the original document in which each pixel in the original document corresponds to subbands in the same location. To detect object contour, the coefficients corresponding to pixels of document are classified into three categories like strong edges, weak edges and noise. The strong edges depict those pixels with large magnitude coefficients in all subbands. The weak edges represent those pixels with large magnitude coefficients in some directional subbands but small magnitude coefficients in other directional subbands within the same scale. The noises illustrate those pixels with small magnitude coefficients in all subbands.

By observation of the coefficients in the multiscale decomposition, we have observed that the coefficients in some directional subbands are able to maintain the document structure as strong edge, and background pixels as weak edges or noises. These edges are classified by:

$$\begin{cases} \text{strong edge, if } mean \geq c\sigma \\ \text{weak edge, if } mean < c\sigma, max \geq c\sigma \\ \text{noise, if } mean < c\sigma, max < c\sigma \end{cases} \tag{3.16}$$

where *mean* and *max* are the mean and the maximum magnitude of the coefficients for each pixel across directional subbands, c is a constant between 1 and 5, σ is the noise standard deviation of the subbands of directions at a specific pyramidal level.

As discussed above, we implement four levels of NSCT decomposition on the transformed document to determine which level is suitable for contour detection. The notation $co_{i,j}$ is used to demonstrate the variance of coefficients at the i^{th} directional subband of the j^{th} pyramidal scale, $co_{i,j} = \{cval_k | k \in (1, ..., n)\}$ where n is the number of coefficients, $cval_k$ is the k^{th} coefficient value. The directions at various levels are obtained as follows: one direction at the first level ($co_{1,1}$), one direction at the second level ($co_{1,2}$), two directions at the third level ($co_{1,3}$, $co_{2,3}$) and eight directions at the fourth level ($co_{1,4}$, $co_{2,4}$, $co_{3,4}$, $co_{4,4}$, $co_{5,4}$, $co_{6,4}$, $co_{7,4}$, $co_{8,4}$). The result of contour detection is depicted in Figure 3.16.

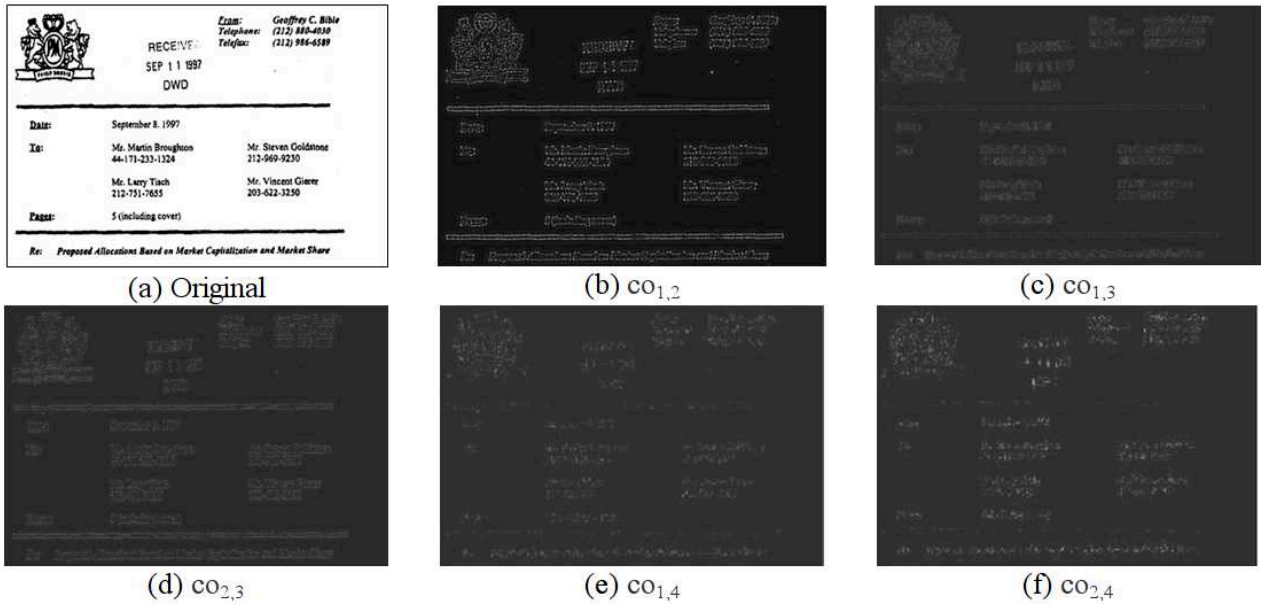


Figure 3.16: *Illustration of four levels of NSCT decomposition for stable hiding region detection.*

With the obtained coefficients in Figure 3.16, we have observed that the coefficients are blurred at level 1. Most of object edges are maintained at level 2 but many noises are presented. Coefficients at level 3 (co_{13}, co_{23}) give better object contours of document. Meanwhile, there are too many edges lost at level 4. Thus, the variance of coefficients at level 3 is suitably opted for detecting the objects contour of documents. However, there are two set of coefficients acquired at this level, so these coefficients need to be integrated to generate a united set of coefficients for describing objects contour of the document. To integrate coefficients from two directions at level 3, we have eliminated unnecessary edges and made the integrated contours keeping more their edge features by:

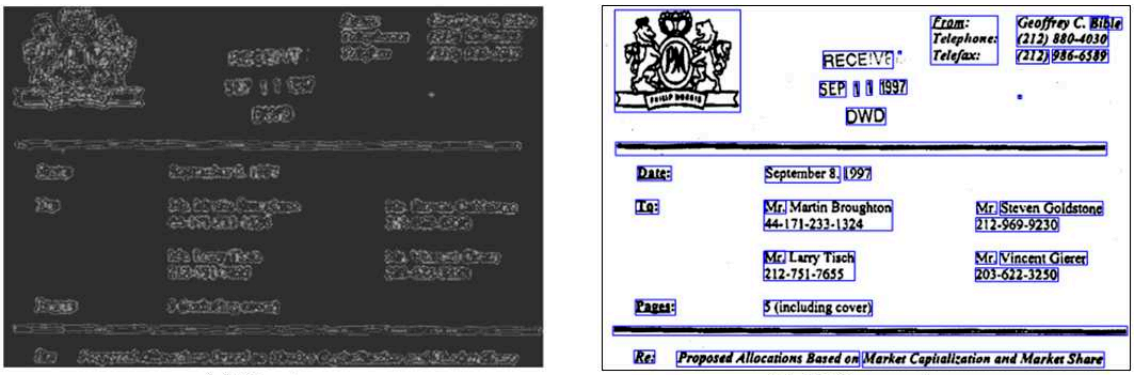
$$co_{13} = co_{13} \setminus \{cval_k < (mean(co_{13})/stdev(co_{13}))\} \tag{3.17}$$

$$co_{23} = co_{23} \setminus \{cval_k < (mean(co_{23})/stdev(co_{23}))\} \tag{3.18}$$

where *stdev* is the standard deviation of corresponding coefficients.

$$contours = \sqrt{co_{13}^2 + co_{23}^2} \tag{3.19}$$

The result of integrated contour pixels is depicted in Figure 3.17(a). These contours are used to construct stable regions of document with the support of convex hull. The blue rectangles in Figure 3.17(b) represent the stable regions which are employed to hide secret information. We consider contents extracted separately from document as objects.



(a) Contours (b) Hiding regions

Figure 3.17: *The integrated coefficients (a) and bounding boxes (b) representing the stable regions.*

3.3.2 Identification of potential positions for watermarking

Due to the essence of documents, there are a lot of empty spaces contained inside objects or interlaced among them. The empty regions are unusable in data hiding and detection. The document's objects are constituted by stroke and filling part as illustrated in Figure 3.18 in which red and green points represent the stroke and filling part of an object respectively. To determine the positions of the object's filling part, we separate the objects situated inside stable regions, and these objects are obtained by detecting their connected components. Then, the detection of stroke and filling positions of an object is performed by checking the eight connected neighborhoods of its pixels. The document difference obtained in Section 3.3.1 shows that the object's stroke part is easily affected by noises. Therefore, only positions of object's filling part are appropriately selected, and these positions are used to map to their corresponding gray level values where the watermarking process will be conducted.

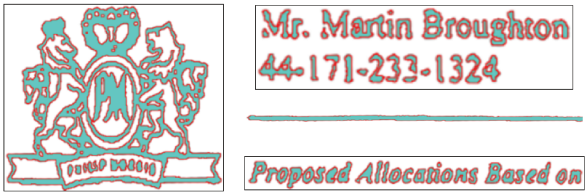


Figure 3.18: *The object's stroke and filling part located inside each stable region.*

3.3.3 Geometric correction based on points of object stroke

After printing and scanning, the watermarked document could rotate a certain angle, and its size might change due to scanning with different resolutions. To obtain high performance of the watermarking scheme, the input document needs to be transformed into a standard form to minimize these kinds of distortion prior to hiding and extracting the secret data. The direction of standard document is vertical wherein the text line of a document is parallel to the axis X. The estimation of parameters for correcting geometric distortion is based on the minimum bounding box which contains a set of points extracted from documents. The set of points here corresponds to the integrated coefficients obtained from the step of contourlet detection above. We assume that the minimum bounding box surrounding the whole document is depicted as in Figure 3.19. The rotation angle θ is calculated from the top edge of this minimum box with respect to the axis X, and the top edge $\overline{AB}$ is used to calculated the rotation angle as similar to Equation 2.3.

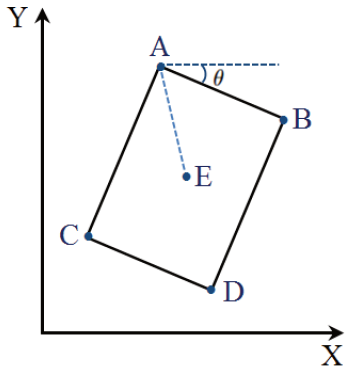


Figure 3.19: *The minimum bounding box used for estimation of scaling and rotation angle parameter for document standardization.*

Next, we estimate the scale factor based on the Euclidean distance d of such two points as a top left point A and an intersection point E of two diagonal lines of the minimum rectangle as illustrated in Figure 3.19. The scale factor s is computed by:

$$s = \frac{c}{d} \tag{3.20}$$

where c is a predefined constant, and it is determined based upon the minimum bounding

box of the original document. With the obtained parameters θ and s, the input document is then transformed into its standard form by affine transformation as in Equation 2.5.

3.3.4 Watermark hiding process

The overall hiding process is depicted in Figure 3.20. The watermark used in this work is a text message, so it has to be converted into a sequence of bits. The hiding process basically consists of the main following steps.

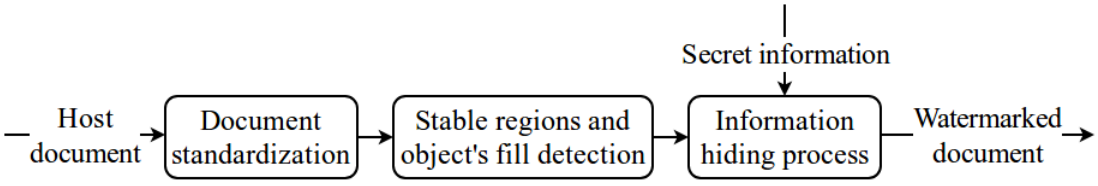


Figure 3.20: *Information hiding process.*

Step 1 - Transforming the input document into its standard form

This task improves the accuracy of extracted secret message in case the watermarked documents are rotated a certain degree, or scaled up or down due to the distortion of printing and scanning process. The transformation is performed as in Section 3.3.3.

Step 2 - Detection of stable regions

The stable regions are obtained by Section 3.3.1. We sort these regions based on their area in descending order. There are three possible kinds of extracted regions including: (c_1) Small regions that are unstable and need to be eliminated; (c_2) Nested regions are also eliminated. If we hide watermark bits into a nested region, the hidden data within an inner region will be overwritten by the data hiding operations carried out in an outer region. In this case, we only keep the outermost region; (c_3) Overlapping regions also overwrite some of the previously hidden data so that the intersection part of a smaller region is merged to a larger region.

Step 3 - Identifying positions of the filling part of the objects

As discussed above, the pixel values belonging to the stroke part of an object are considerably fluctuated in case noises like print-and-scan, so we only keep the filling part of document objects for watermarking scheme. The detection of object's filling part is descrided in Section 3.3.2.

Step 4 - Hiding watermark bits into document

We sort the positions of each object fill by y-coordinate in ascending order. Then, for each object, we split the pixels belonging to this object into groups of m successive pixels. Then, for a given group G, the m pixel values are used to carry one watermark bit. A weight $W(w_i)$ is assigned to each pixel of the group G as illustrated in Figure 3.21(a).

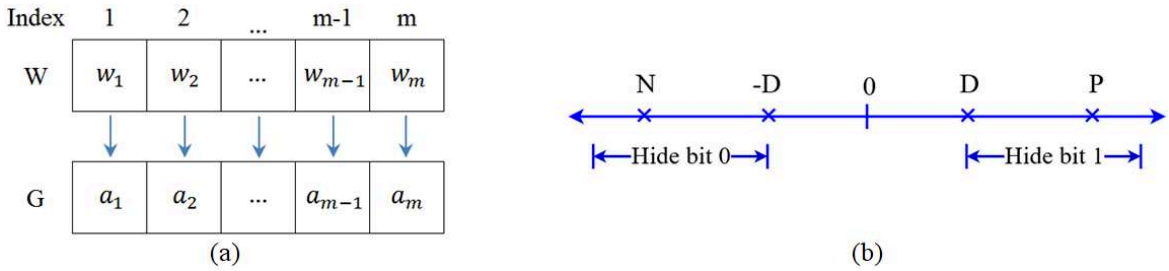


Figure 3.21: *Demonstration of assigning weight to a group of pixel values (a) and ranges of hiding factor (b).*

Each element of weighted array W holds a value by:

$$w_i = \begin{cases} -1, & \text{if } i \leq m/2 \\ 1, & otherwise \end{cases} \tag{3.21}$$

where m is an even number ($m > 0$)

To determine which gray level values in G are adjusted to hide watermark bit, the hiding factor f for the k^{th} group is calculated by:

$$f_k = \sum_{i=1}^{m} w_i \times a_i \tag{3.22}$$

where m is the number of elements in W_k and G_k

The set of hiding factors of whole document ($F = \{f_1, f_2, ..., f_n\}$) includes negative ($N$), zero and positive ($P$) values and is distributed in the range $[N, P]$. We use the range of negative values of F to hide bit 0 and positive range of F to hide bit 1. However, when the watermarked documents undergo noises, the hiding factor might move from the negative range to the positive one and vice versa. Thus, these two ranges need to be separated by a certain distance D for improving the accuracy of watermark detection. Depending on the hiding factor of each group and watermark bit, its gray level values will be adjusted such that its hiding factor lies on the range with $f_k \leq -D$ or $f_k \geq D$ as described in Figure 3.21(b). A watermark bit is hidden into an appropriate range by:

- if the j^{th} watermark bit $wm_j = 0$:
  - ☐ $f_k \leq -D$: gray level values in G keep unchanged.

- ☐ $f_k > -D$: adjust gray level values in G such that $f_k \leq -D$.

– if the j^{th} watermark bit $wm_j = 1$:

- ☐ $f_k \geq D$: gray level values in G keep unchanged.
- ☐ $f_k < D$: adjust gray level values in G such that $f_k \geq D$.

The adjustment is done by: (i) swapping the first block of $m/2$ gray level values with the second block of $m/2$ gray level values; or (ii) modifying gray level values of the first or second block of $m/2$; or both of (i) and (ii).

3.3.5 Watermark detection process

The steps for watermark detection is conducted in the same manner with the watermark hiding process, it just differs in the data extraction process. The watermark bit wm_j is extracted by:

$$wm_j = \begin{cases} 0, & \text{if } f_k \leq 0 \\ 1, & otherwise \end{cases} \tag{3.23}$$

The accuracy of information detection is measured by bit error rate (BER), which is defined as a ratio between the number of detected error bits and the total number of hidden bits.

3.4 Summary

In this chapter, we present the feature points-based steganography scheme and stable regions-based watermarking scheme. For the method based on feature points, we make use of SURF detector for extracting the feature points from the documents. Although these kinds of feature points give high performance like for natural images (the schemes are designed in the transformed domain), they give low performance when applied on the documents, as shown in the experiments for detecting the hidden information detailed in Section 5.2. In addition, we also proposed another detector for detecting the feature points, which is more stable than the well-known detectors in terms of distorted document images. However, the steganography scheme based on the feature points extracted from our proposed detector still has not meet the requirement of the robustness against practical distortions like printing and scanning, or print-photocopy-scan distortion.

For this reason, we have proposed to improve the robustness of the scheme by designing another watermarking scheme based on the stable regions extracted from the documents instead of the feature points. To achieve this objective, we take advantage of common image

processing operations and non-subsampled contourlet transform to detect stable regions from the documents. The new watermarking algorithm has been proposed, which is relied upon a group of sucessive pixel values associated with weights for carrying one watermark bit. As a result, the stable regions-based watermarking scheme has significantly improved the robustness, especially the scheme is capable of tolerating the real distortions like printing and scanning at high resolution of 600 dpi. However, the scheme fails to detect the hidden information from the watermarked documents, which are scanned at lower resolutions of 400, 300 and 200 dpi (as presented in Section 5.3). Thus, the performance of the scheme needs to be improved in order to meet higher requirement of practical applications.

Chapter 4

Securing document images via deep learning

In this chapter we introduce five other watermarking schemes by making use of deep learning with the aim of improving the scheme robustness including: three schemes for grayscale typewritten documents; one scheme for both handwritten and typewritten documents; and one scheme for binary documents. To improve the stability of extracted hiding regions which has been presented in Section 3.3.1, we make use of fully convolutional networks, which has been initially proposed for semantic image segmentation. This kind of network is also utilized to detect the variations of document characters and symbols from the watermarked documents. Besides, we also take advantage of generative adversarial networks to:

- Generate an intermediate document from an input document, which is used as a reference for the watermarking process.
- Produce new fonts of a document, or variants of document characters and symbols from their skeleton. With respect to watermarking algorithms, we develop them based on pixel intensity level, and shape of characters and symbols.

4.1 Introduction

Although the stable regions-based watermarking scheme has considerably improved the performance of information detection in comparison with the steganography scheme based on the feature points, the robustness didn't reach the expected level. Thus, we have proposed to take advantage of deep learning to enhance the scheme robustness. Deep learning is a subfield of machine learning research, which is utilized to design models and learning algorithms for deep neural networks. It has been incorporated in a large number of applications of pattern recognition and artificial intelligence including character and text recognition, image segmentation, object detection and recognition, traffic sign recognition and so on.

Deep learning has its roots from conventional neural networks, and it enables computational models of multiple processing layers to learn and represent data with multiple levels of abstraction mimicking how the brain perceives and understands multimodal information. Hence, deep learning implicitly captures intricate structures of large scale data. The recent researches in deep learning have shown that it outperformes previous state-of-the-art techniques in several tasks. Specifically, in our work, we make use of such two kinds of neural networks, namely fully convolutional networks (FCN) and generative adversarial networks (GAN). The principle of FCN and its application are presented below while the one of GAN is depicted in Section 4.4.

Fully convolutional network is a contemporary technique that provides very good performance for several applications in the field of pattern recognition, which is adjusted to solve the issue of detecting document content regions for watermarking. FCN has been initially proposed to solve the problem of semantic image segmentation[66;131], and recently this advanced technique has been efficiently applied for scene text detection[132–134], document structure segmentation[135;136] and document image binarization[137]. Compared to convolutional neural networks which only work with the fixed size of an input images and only generate one label per image, FCN is designed to provide pixel level predictions, and it can take the input documents with arbitrary size and generate outputs as feature maps with the same size of the inputs.

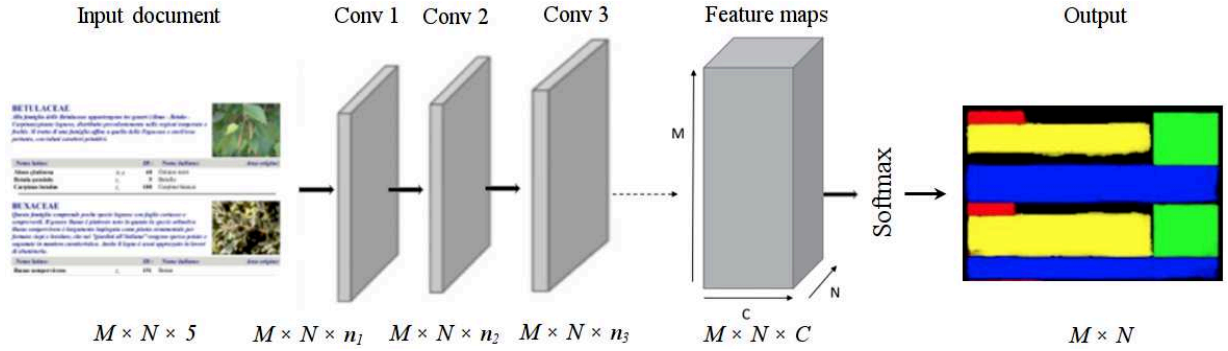


Figure 4.1: *Fully convolutional network architecture.*

The FCN is designed by replacing fully connected layers with convolutional layers which enable it to preserve coarse spatial information that is essential for specific tasks. The convolutions are chosen in such a way that the input document is transmitted without any change in the spatial dimension, it means that the height and width of the output document keep the same dimension than the input. The fully convolutional network predicts all the pixel labels at once rather than having individual patches from a document independently evaluated for pixel labeling. The output layer of this kind of network consists of C feature maps, where C is the number of labels, including the background, in that each pixel can be classified. If the height and width of the original document are h and w respectively, then the output of the network comprises $h \times w \times C$ feature maps. For the ground truth document, there should be C number of segmented documents corresponding to the C classes. For any spatial coordinate (h_1, w_1), each of the feature maps contains the score of that pixel

pertaining to the class that the feature map is associated with. These scores across the feature maps for each spatial pixel location (h_1, w_1) are obtained from the softmax function over the different classes.

The architecture of fully convolutional networks is presented in Figure 4.1 in which the number of output feature maps corresponding to five classes are displayed. Assume that the score at the spatial coordinate (i, j) for the k^{th} class is depicted by $s_k(i, j)$, the probability of the k^{th} class for the pixel at spatial coordinate (i, j) under the Softmax activation function is given by:

$$P_k(i,j) = \frac{e^{s_k(i,j)}}{\sum_{k'=1}^{C} e^{s_{k'}(i,j)}} \tag{4.1}$$

Assume that the ground truth labels at the spatial coordinate (i, j) for the k^{th} class are given by $g_k(i, j)$, the cross-entropy loss of the pixel at spatial coordinate (i, j) can be computed by:

$$\mathcal{L}(i,j) = -\sum_{k=1}^{C} g_k(i,j) log P_k(i,j) \tag{4.2}$$

Assume that the input document of size $M \times N$ used to feed to the network, the total loss for a document is given by:

$$\mathcal{L}_D(i,j) = -\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\sum_{k=1}^{C} g_k(i,j) log P_k(i,j) \tag{4.3}$$

The output class $\hat{k}$ for a pixel at spatial coordinate (i, j) can be determined by taking the class k where the probability $P_k(i, j)$ is maximum as below.

$$\hat{k} = \underbrace{\text{Arg max}}_{\text{k}} P_k(i,j) \tag{4.4}$$

The output feature maps of the network for document structure segmentation in Figure 4.1 correspond to five classes of document regions including subsection headings, paragraphs, tables, pictures and background. We can see that there is a feature map associated with each class or label, and the spatial dimensions of the feature maps are the same size of the input document.

With the FCN presented in Figure 4.1, all convolutional layers in the network retain the spatial dimensions of the input document. However, the documents with high dimension or resolution could make computational time of the network becoming more intensive, es-

pecially if the number of feature maps in each convolution is high. To deal with this issue, the operations of downsampling are applied to the convolutional layers during the process of feature extraction to reduce the size of feature maps, and the feature maps with dimension reduction will be restored to the dimension of original document by using upsampling operations. It is important that the downsampling and upsampling tasks should retain the spatial information of the documents. The downsampling task can be performed through stride and pooling (max pooling or average pooling) operations, which are applied to the convolutional layers. Meanwhile, the techniques that are commonly used to upsample the document or feature map consisting of unpooling, max unpooling and transpose convolution. For upsampling operations, few convolutional layers are required. An example of downsampling and upsampling operations for this network is depicted in Figure 4.2.

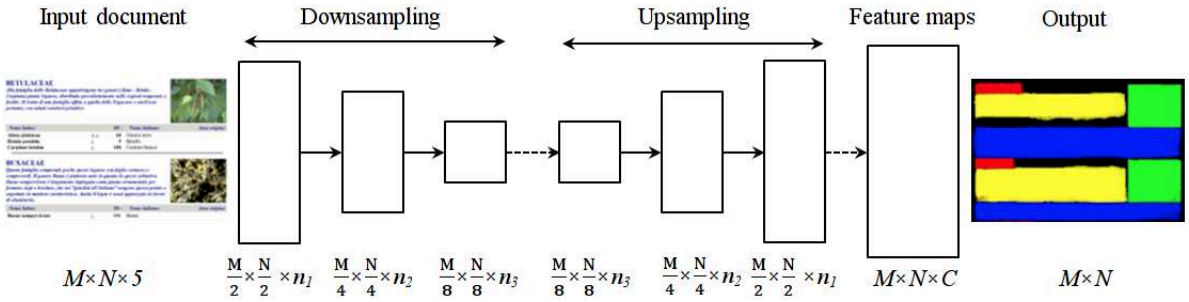


Figure 4.2: *Illustration of FCN with downsampling and upsampling operations.*

This kind of network is adjusted to solve the problem of detecting document content regions for watermarking system, and its performance is demonstrated through watermarking schemes detailed below.

4.2 Watermarking scheme for typewritten documents

Depending on our survey, we have noticed that deep learning, specifically convolutional neural networks, generative adversarial networks or CNN-based autoencoder, has been exploited to develop data hiding schemes for natural images as presented in Chapter 2. However, the FCN has been employed yet in the field of data hiding. Although the watermarking scheme presented in Section 3.3 has improved the performance as compared to the existing approaches, the extracted regions for watermarking development is still unstable against high distortions. For this reason, we propose another FCN-based watermarking approach for document images with mixed content and introduce this advanced technique to the field of document watermarking for security concern. Unlike the CNN-based approaches in which the authors leverage weight parameters of deep learning framework, or the output image of the network for watermarking process on the fixed size images, here we train the FCN so that the trained network can be used to generate a salient map describing watermarking regions of a document with arbitrary size. The process of our watermarking development is separately designed with the phase of training the network.

4.2.1 Detection of stable hiding regions using FCN

In reality, the document images are represented under several forms with various size. After watermarking process, the watermarked document is expected to keep a similar shape with the one of host document. The FCN[66;131] is adopted for developing our watermarking scheme because this network is able to take an arbitrary sized document and generate an output with the same dimension (this network has no any fully connected layer which is replaced by convolutional layer). The FCN is initially designed for semantic image segmentation, and this kind of network is very powerful to solve the problem of document structure segmentation[135;136] and document binarization[137]. By experiments, we have shown that the segmented content regions of document such as running text, headline, picture, table, etc. are stable against distortions. Thus, we transform the problem of document structure segmentation, which includes many labels describing various segmented content regions, to the problem of watermarking regions detection which consists of two labels: one describes background region (black color in Figur 4.3(b)), and the other depicts all segmented content regions (white color in Figure 4.3(b)). It means that in the context of our work, the segmented content regions are assigned with the same label, e.g. a segmented region containing the running text does not differentiate from the other segmented content regions and so on.

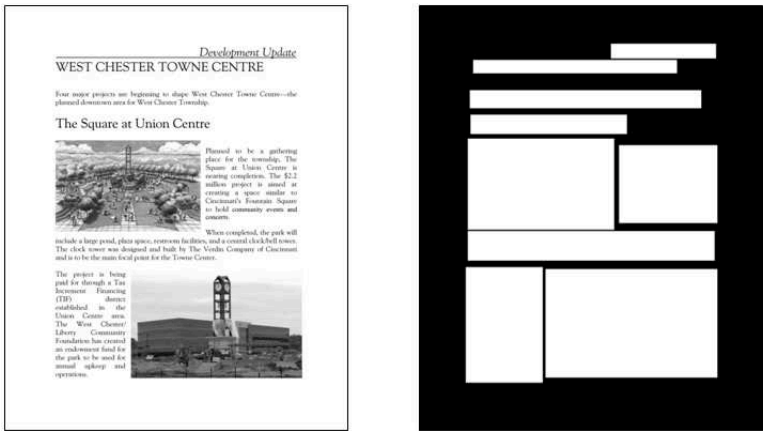


Figure 4.3: *The illustration of a mixed document (a) and its ground truth document (b).*

Our network as depicted in Figure 4.4 is based on the principle of FCN presented in[131] in which we use the VGG 16-layer[138] network and replace its three fully connected layers with convolutional layers (two blocks of additional layers) for preserving spatial information. The feature maps generated in the phase of convolution operations (downsampling) are reduced in dimension, so they need to be reconstructed by performing upsampling through a few transposed convolutional layers. This task produces the feature maps with the same dimension of original document in the output layer. Besides, we apply Softmax layer on the output layer to transform the result of network into a two-class problem for representing the probability of document's watermarking regions.

Next, we describe briefly our network as follows. The first five convolutional blocks are

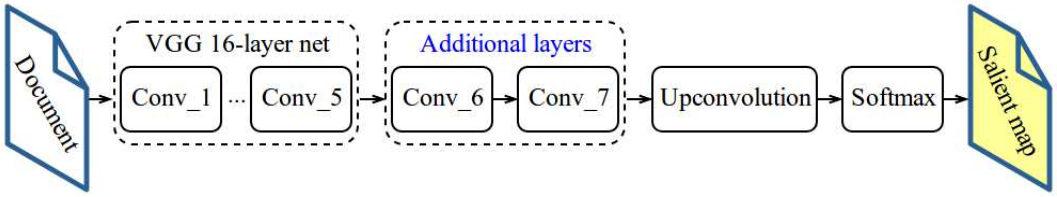


Figure 4.4: *The architecture of FCN for detecting watermarking regions.*

analogous to the VGG-16 net in which the first two blocks (Conv_1, Conv_2) contain two convolutional layers in each, while the next three blocks (Conv_3, Conv_4, Conv_5) contain three convolutional layers in each. The kernel size used in the first five convolutional blocks is set to 3×3. Each additional block (Conv_6, Conv_7) contains two convolutional layers including one with a kernel size of 3×3, and the other with a kernel size of 2×2. The step size used for shifting convolution kernel is of 1×1. The activation function used in all layers of network is a rectified linear unit (ReLU). Besides, the max pooling layer with a kernel size and stride of 2×2 is applied after the last convolutional layer of each block except for block of Conv_7. We apply dropout layer after Conv_6 and Conv_7 for preventing the problem of overfitting. The phase for reconstructing the spatial dimension of the original document is represented by the "Upconvolution" block (upsampling operations) in Figure 4.4 wherein a few transposed convolutional layers are utilized for retaining the spatial information. By our experiment, we have observed that the transposed convolution restores better spatial information than other upsampling operations. With the transposed convolution, the filter values at a specific location are weighted by the input value at which the filter is placed, and the weighted filter values are populated in the corresponding locations in the output. The outputs corresponding to each of the input values are added to produce the final output.

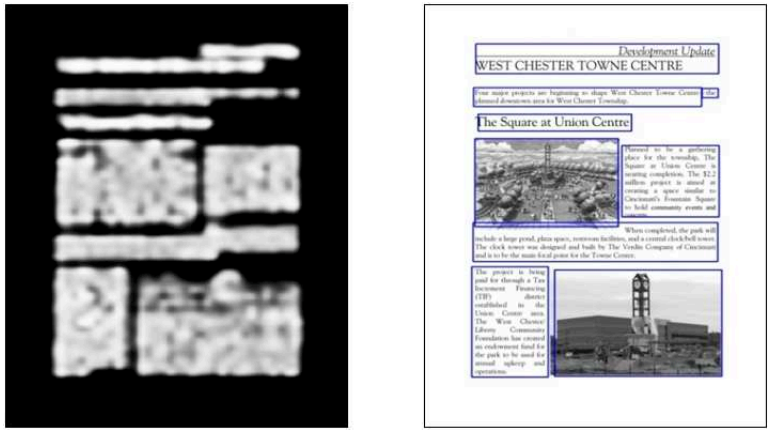


Figure 4.5: *The generated salient map (a), and the bounding boxes surrounding the document content regions (b).*

The annotating documents are performed by creating the ground truth of document's content regions. The input documents (Figure 4.3(a)) along with their ground truth segmentation as described in Figure 4.3(b) are used to feed the network for training purpose. By

observation, we have seen that the feature maps generated from low blocks of convolutional layers represent overall shape of document content while regions-specific information at high blocks. The salient map is computed based on the scores of output feature maps obtained from our trained network, and it is represented under grayscale form as in Figure 4.5(a). Based on the output salient map, which is converted into binary form, the bounding boxes as blue rectangles on Figure 4.5(b) surrounding the content regions of document are easily detected by utilizing connected components. These regions are considered as watermarking regions.

4.2.2 Feature points-based geometric correction

During the process of printing and scanning, the watermarked document is possibly subjected to distortions such as rotation and scaling. The distortions could make the watermarked document rotating a certain angle, or changing its dimension due to scanning with different resolutions. Thus, these distortions need to be mitigated before conducting watermarking operations by transforming the input document into a standard form. In this work, we adopt BRISK[64] detector for detecting keypoints, and these keypoints are used for determining the minimum rectangle surrounding the content of entire document, which is illustrated as blue rectangle in Figure 4.6(a). The small color circles in Figure 4.6(a) are feature points extracted by BRISK detector. Then, the rotation angle θ is estimated based on this minimum rectangle.

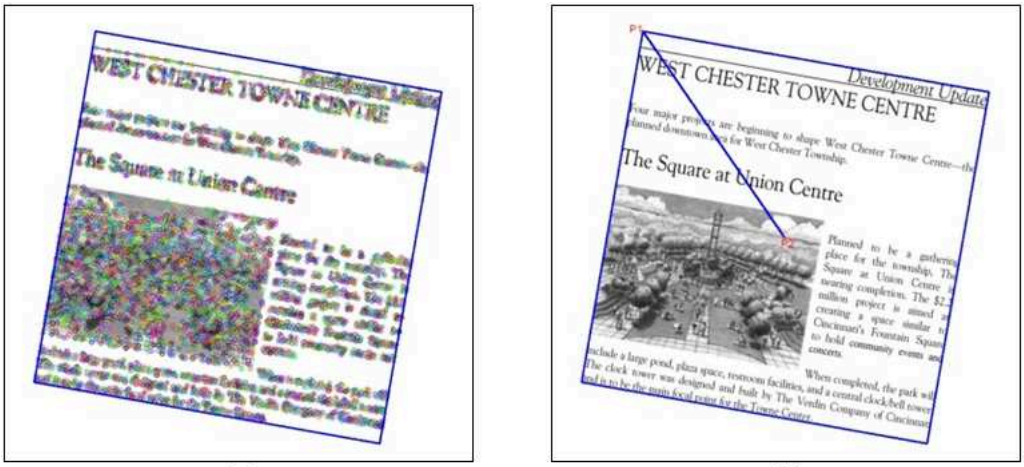


Figure 4.6: *Estimation of geometric parameters.*

Next, the scale factor s is calculated as a ratio between a predefined constant and the Euclidean distance of such two points as a top left point P_1 and a center point P_2 as illustrated in Figure 4.6(b). With the acquired parameters of rotation θ and scale s, the input document is then transformed into its standard form by affine transformation as in Equation 2.5.

4.2.3 Data hiding process

The main steps of information hiding process is illustrated in Figure 4.7. The watermark hiding process basically comprises the following main steps:

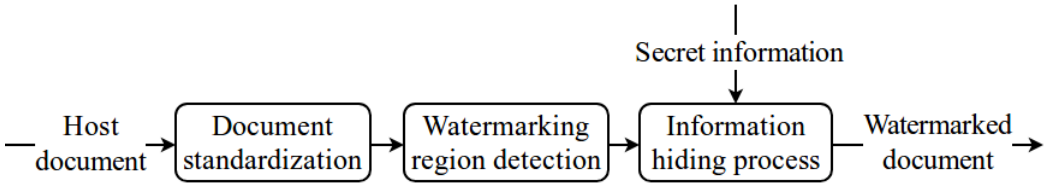


Figure 4.7: *The main steps of watermark hiding process.*

Step 1 - Transforming the input document into its standard form

The information hiding algorithm in this work is designed based on a $m \times n$ pattern of pixel values for carrying one information bit. Thus, the pattern of pixel values extracted from an undistorted document is possibly different from the one extracted from geometrically distorted document. To minimize this inconsistency, the document is transformed into its standard form before starting the procedure of information hiding and detection. This task will result in increasing the performance of detecting hidden information from the distorted documents. The document standardization is carried out as described in Section 4.2.2.

Step 2 - Detection of watermarking regions

The watermarking regions are detected by utilizing FCN as detailed in Section 4.2.1. With this approach, we can easily localize any type of document contents such as text, picture, table and so on where we want to hide the secret information. In addition, this method is able to efficiently identify text and non-text elements of documents so that it can be applied to address the shortcoming of state-of-the-art schemes designed for pure text documents. To hide as many information bits as possible into a document, we take all its extracted content regions for information hiding. Similar to the stable regions in Section 3.3.1, the extracted watermarking regions based on FCN could be nested or overlap. This issue has been solved similarly to the method presented in Section 3.3.4.

Step 3 - Identifying watermarking positions

Depending on the natural shape of document content, we use a watermarking pattern B with size of $m \times 2$ as on Figure 4.8(a) to scan through each of watermarking regions for locating appropriate positions where the watermarking process will be conducted. As we know that, for each of document characters, there are many pixels whose grey values with high intensity are interlaced either inside individual character or around it. These positions

are unusable for watermarking process, so they have to be eliminated. To make sure that the watermarking process is not fallen into these positions, all gray level values $(p_1, p_2, .., p_{m\times 2})$ of the watermarking pattern B have to be less than a threshold δ.

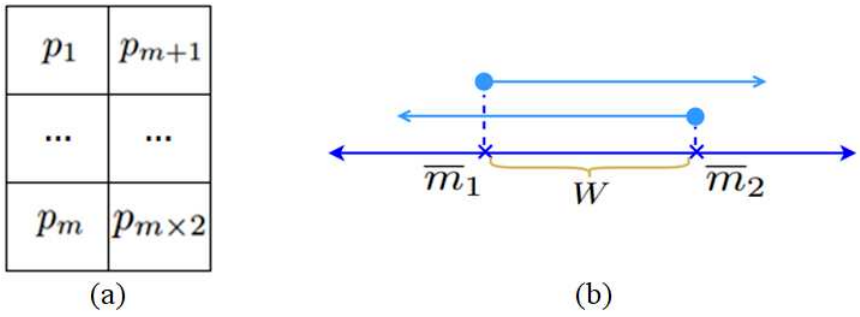


Figure 4.8: *The watermarking pattern B_i with size of $m \times 2$ (a), and the distribution of mean values $\overline{m}_1$ and $\overline{m}_2$ calculated from B_i (b).*

The watermarking pattern is used to carry a watermark bit, so the number of watermark bits that a document can carry is depending on the number of satisfied watermarking patterns described above.

Step 4 - Hiding watermark bits into document

We divide the gray level values of each of watermarking patterns B into two groups of m values such as $(p_1, ..., p_m)$ and $(p_{m+1}, ..., p_{m\times 2})$. The idea of hiding an information bit into the document is based on two mean values $(\overline{m}_1, \overline{m}_2)$ corresponding to two groups of pixel values of B, and the absolute difference d_i between these mean values. It is possibly either $\overline{m}_1 \leq \overline{m}_2$ or $\overline{m}_1 > \overline{m}_2$, and we use $\overline{m}_1 \leq \overline{m}_2$ for carrying bit 0 and $\overline{m}_1 > \overline{m}_2$ for bit 1.

$$\begin{cases} \overline{m}_1 = \frac{1}{m}\sum_{k=1}^{m} p_k; \overline{m}_2 = \frac{1}{m}\sum_{k=m+1}^{m\times 2} p_k \\ d_i = |\overline{m}_1 - \overline{m}_2| \end{cases} \tag{4.5}$$

However, when the watermarked documents suffer from distortions, the mean value $\overline{m}_1$ might move passed towards the mean value $\overline{m}_2$ and vice versa as illustrated in Figure 4.8(b). Thus, these two means need to be separated by a certain distance of W for improving the precision of watermark detection. The value of W will determine the robustness of the scheme. A watermarking pattern B_i carrying a watermark bit wm_i is given by:

- if the i^{th} watermark bit $wm_i = 0$:
  - ☐ $\overline{m}_1 < \overline{m}_2$ and $d_i \geq W$: the gray level values in B remain unchanged.
  - ☐ Otherwise: adjust the gray level values in B such that $\overline{m}_1 < \overline{m}_2$ and $d_i \geq W$.
- if the i^{th} watermark bit $wm_i = 1$:

- ☐ $\overline{m}_1 > \overline{m}_2$ and $d_i \geq W$: the gray level values in B remain unchanged.
- ☐ Otherwise: adjust the gray level values in B such that $\overline{m}_1 > \overline{m}_2$ and $d_i \geq W$.

The adjustment is performed by: (i) permuting the first group of gray level values $(p_1, ..., p_m)$ with the second groups of gray level values $(p_{m+1}, ..., p_{m\times 2})$; or (ii) modifying gray level values of the first or second group; or both of (i) and (ii).

4.2.4 Data detection process

The watermark extraction is the inverse of the watermark hiding process. The watermarked document is first transformed into its standard form in order to minimize the geometric distortions. The geometrically corrected document is then fed into the fully convolutional networks, and this network will produce a salient map with the same dimension of the input document. The generated salient map is used to help identify the content regions of document, which is regarded as watermarking regions, where we wish to detect the secret information hidden inside. The watermarking pattern of size $m \times 2$ is positioned within each of these watermarking regions, and the pixel values of each of watermarking patterns is partitioned into two sets. The mean values $\overline{m}_1$ and $\overline{m}_2$ corresponding to these two sets of pixel values will determine either bit 0 or 1 is extracted, and this is estimated by:

$$wm_i = \begin{cases} 0, & \text{if } \overline{m}_1 \leq \overline{m}_2 \\ 1, & otherwise \end{cases} \tag{4.6}$$

The precision of information detection (bit error rate) is measured by the proportion between the total number of incorrectly extracted bits and the total number of hidden information bits.

To conclude, we have improved the stability of watermarking regions extracted from the documents by using FCN. With the FCN networks, the scheme can easily provides a flexible way to hide the secret data into a specific region of the document. This region can be textutal content or a picture. During the experiments, we have observed that the watermarking regions extracted by FCN networks give better stability than the method presented in Section 3.3. In addition, the precision of hidden data detection has been improved compared to our previous schemes. However, the scheme gives low performance when detecting the watermark from the watermarked documents undergone the scanning operation with the resolution of 400 dpi or lower. The experimental results of this method are presented in Section 5.4.

4.3 Watermarking scheme for handwritten documents

With the fast-growing advanced technologies, the action of writing with pen has been considerably decreased in our daily activities. However, the handwriting documents still remain an important role in the digital age. The handwriting documents are really inevitable and in use at various sectors such as notarized agreements, judicial documents, sworn statement, bank transfer forms, etc. There are several existing researches dealing with the integrity and authenticity of digital material like signature determination, handwriting identification, ink verification and other document characteristics in the field of forensic specialty. Before going into our approach for securing handwriting documents, we would like to introduce a couple of typical works related to this field.

Signature determination: It can be used for the purpose of individual identification and document authentication. Signature detection for document image retrieval has been proposed by Zhu *et al.*[4]. To detect and segment signature from document images, the authors have proposed a multiscale approach which captures structural saliency using a signature production model and computes the dynamic curvature of 2D contour fragments over these multiple scales. The signature production model is based on the incorporation of two degrees of freedom in the Cartesian coordinates: when producing a signature the pen moves in a fashion with reference to a sequence of shifting baselines. Within a short curve segment, the baseline remains unchanged; the locus of the pen maintains a propositional distance from the local center point to the local baseline. Thus, a fragment of signature can be equivalently considered as concatenations of small elliptic segments. Regarding signature recognition and retrieval, the authors measure the shape dissimilarity based on anisotropic scaling and registration residual error where a supervised learning framework is utilized to combine complementary shape information from different dissimilarity metrics with the support of linear discriminant analysis. Figure 4.9 shows the infinite number of geometric curves that pass two given end points E_1 and E_2 on a signature, very few are realistic (solid curves) whereas the dotted line is an unrealistic curve.

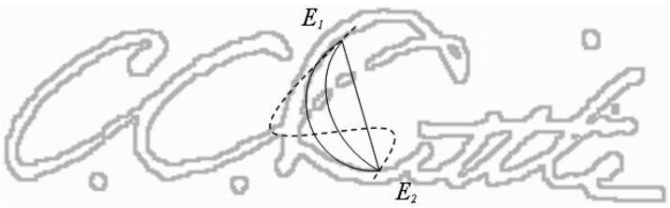


Figure 4.9: *The geometric curves*[4] *passing through the two points E_1 and E_2: solid lines are realistic curves, and dotted line is unrealistic curve.*

Ahmed *et al.*[139] proposed a method for signature extraction whereas SURF is utilized to differentiate the machine printed text from signatures. To do so, for all connected components from the printed text images, the corresponding extracted keypoints and descriptors are added to a database for printed text features. For connected components of signature images, their extracted keypoints and descriptors are also added to another database for signature features. These two databases provide a reference for the matching of features, which

is used for the purpose of signature segmentation. Another method[140] detects signature using hyper-spectral imaging. In this work, the authors have pointed out the characteristics of hyper-spectral response of document images in which the response of printer ink and background is almost consistent across all the bands whereas the response of signature pixels varies a lot, especially in the bands near infrared region. To extract signature from documents, the connected components are first extracted from the printed text documents. Then the corresponding extracted keypoints and descriptors are stored in a database for printed text features. With respect to connected components of signature images, their keypoints and descriptors are also stored in another database. These databases are used for signature segmentation, and the authors utilize k-nearest neighbours algorithm to classify the connected components instead of Euclidean distance metric as method presented in[139].

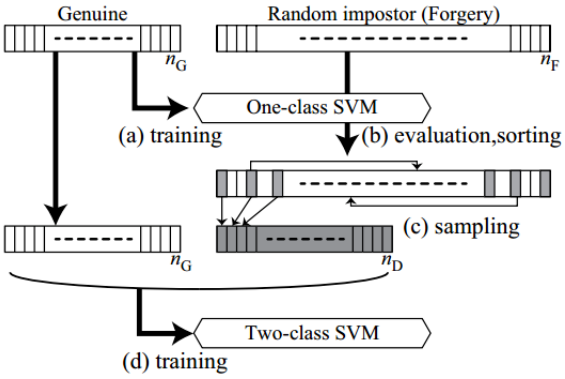


Figure 4.10: *The architecture of SVM training[5] wherein the forgery signatures are generated by verifying signatures of different signers.*

The multi-cript signature verification method[5] uses the genuine signatures of third signers as training samples of the forgery class for support vector machine (SVM) training in which the authors have proposed an effective sampling method that uses one-class SVM to reduce the sample number for the training dataset. This method provides procedures for both online and offline signature verification. For the offline verification procedure, the system consists of signature image generation for which the grayscale values reflect the velocity, and the thickness of strokes reflects pressure of pen movement. To generate the signature image, the authors have combined grayscale values and width of trokes to enhance the appearance difference between genuine and forgery signatures. Meanwhile, the online verification task processes time series data of the signature by using a procedure similar to offline verification for signature image generation. The architecture of this method is depicted in Figure 4.10.

Writer verification: This kind of application is used to authenticate whether a given document is written by a certain individual or not, and this is performed by matching the writers to their handwriting specimens. The general framework of this application is presented in Figure 4.11.

Adak *et al.*[141] have proposed a method that is based on handcrafted features and auto-derived features extracted from intra-variable writing. The handcrated features are extracted by using SVM whereas the auto-derived features are extracted by using CNN network. The

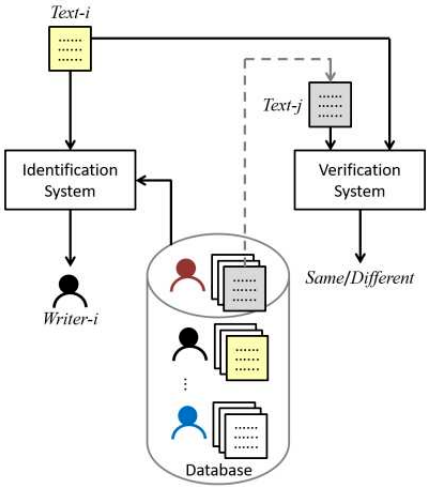


Figure 4.11: *A framework of writer identification and verification*[6].

handcrafted features consist of:

- The macro and micro features: (i) the macro feature vector contains 11 features such as gray-level entropy (f_1), gray-level threshold (f_2), count of black pixels (f_3), interior/exterior contour connectivity $(f_4 - f_5)$, vertical/negative/positive/horizontal contour slope $(f_6 - f_9)$, average slant and height of the text-line $(f_{10} - f_{11})$. The authors define that the features $f_1 - f_3$ are related to the pen pressure, the features f_4 and f_5 are relevant to writing movement, the features $f_6 - f_9$ are related to strokes, the feature f_{10} depicts the writing slant, and f_{11} represents the text proportion; (ii) two paragraph-level macro features include the width and height ratio of paragraph f_{12} and margin width f_{13}. The word-level macro features are upper zone ratio (f_{14}), lower zone ratio (f_{15}) and length (f_{16}); (iii) character-level micro features consist of 192-bit gradient, 192-bit structural and 128-bit concavity features.

- The contour direction and contour hinge features of handwriting strokes.

- The features of the writing direction and the curvature of a writing stroke.

The auto-derived features are computed based upon two types of patches: character-level patch is acquired by calculating the center of gravity of a segmented character, and another patch is obtained as a window of appropriate size centered at a keypoint on writing strokes.

The Bengali writer verification system[142] is also based on handcrafted features and auto-derived features. The authors perform classification for writer verification based on the handcrafted features and then feed it into a CNN network for generating the auto-derived features. The generated features are fed into multi-layer perceptron and Siamese neural network for writer verification. Aubin *et al.*[143] have put forward a writer verification that is based on simple graphemes. The descriptors obtained from individual and simple characters have improved the recognition capacity. The descriptors are constructed in considering several factors such as the width of the stroke, the gray level of the character or grapheme

skeleton, the average of the gray level on the perpendicular line to the skeleton, and the transformation coefficients of the area of the grapheme. The system is implemented by using SVM with k-fold cross validation. Another writer verification method has been proposed by Akao *et al.*[144] in which the multi-dimensional scaling is applied to earth mover's distance (EMD). The earth mover's distance is able to represent the difference of kernel density distributions between writers, and the flow of EMD is calculated between k-means cluster centroids.

The method based on the analysis of a unique sample of a handwriting word[145] makes use of Levenshtein edit distance based on Fisher-Wagner algorithm which is used to estimate the cost of transforming one handwritten word into another. Prior to the process of measuring edit distance between handwriting words, the graphemes need to be generated as elementary components for each word. This is performed by binarization of documents, making skeletonization, and upper contour extraction. Parziale *et al.*[146] have developed a writer verification by using the stroke level measurements. The features of documents are computed by building a triangle between every two consecutive strokes, and they are used to measure the similarity between the genuine and the questioned handwriting document.

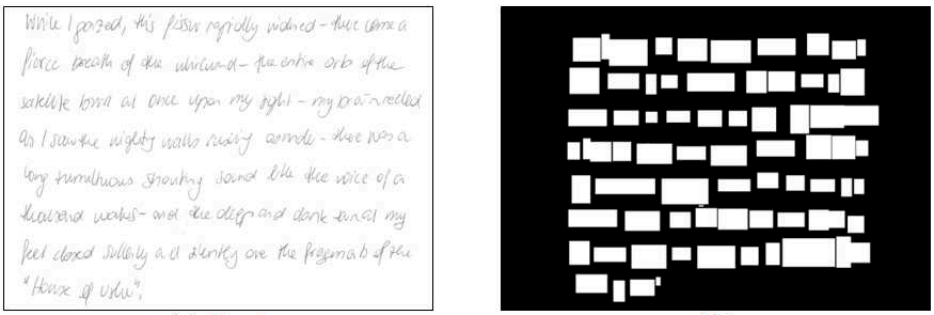
Ink verification: This kind of application is also utilized to assist in making decision whether a given document is genuine or not, specifically it helps answer the question of whether some texts or strokes are added with a different pen or ink. Howe *et al.*[147] have proposed a method of ink verification in which the authors utilize inkball models to generate a varying feature set which is used to train the hidden markov model for character recognition. With this model, the hidden stages correspond to characters in the target language, and each of these characters has a corresponding inkball model produced from a prototype of each character. The prototype of character is based upon k-medoids and information gain, the k-medoids algorithm operates in the same manner as k-means, except that at each step the cluster centers are not the actual centroid but the cluster member which is closest to it. For information gain, it is computed as the difference in entropy of a set before and after a partition into subsets (using a threshold). The maximum information gain over all possible threshold is computed and stored for each candidate prototype, and the candidate with the greatest information gain within each character class becomes the prototype. The obtained character prototypes are then converted into inkball models. The best matching between inkball model and a sample of handwritting text is estimated by the model and the hidden markov model. Another ink verification system[148] is based on the gradients of the spectral power histogram. The input documents have to be pre-processed in order to extract measure points which contain parts of the stroke. The task of pre-processing consists of removal of measurement drop-outs, building of reference vector, wavelength normalization and extraction of measure points. The features used for ink verification are gradients in which the absolute value norm or the Euclidean norm is used to measure the difference between two measure points.

Instead of using the aboved techniques to verify the genuine handwriting documents, we would like to introduce a watermarking method that can be effectively used to secure the legal handwritten documents, and can provide high performance for the purpose of

document verification. The main ideas of our approach are as follows. The handwritten document is pre-processed by replacing gray level values holding high intensity with the mean value of whole document content. The document is then transformed into standard form to minimize geometric distortion. Next, fully convolutional networks (FCN) is leveraged to detect document's content regions used for hiding secret information. Lastly, the data hiding process is conducted by dividing the gray level values of each separated object situated within content regions into two sets for carrying one watermark bit. To our knowledge, there is currently no watermarking approach designed for handwritten document protection.

4.3.1 FCN-based hiding region detection

Similar to the watermarking system presented in Section 4.2, we also make use of FCN for detecting content regions of handwriting documents, which are known as watermarking regions. To apply FCN in this work, we adjust FCN for the purpose of document layout segmentation[135;136] to tackle the problem of watermarking regions detection. The adjustment is to modify the convolutional layers for better feature extraction and representation. Instead of having many labels as applied for layout segmentation, our watermarking regions detection only requires two labels. The first one describes background region as black color in Figure 4.12(b), and the other depicts regions where document contents will be segmented as white color in Figure 4.12(b). Hence, our network to detect the watermarking regions is likely flexible to work well even for handwriting documents with mixed content like text, picture, table, etc. Moreover, this network provides a powerful way to eliminate a certain content regions of document if necessary.



(a)-Doc1 (b)

Figure 4.12: *The handwritten document annotation with two labels: (a) a sample of handwriting document, and (b) its corresponding ground truth document.*

The FCN network for detecting the watermarking regions is depicted in Figure 4.13 in which the blocks (B1-B7) contain convolutional layers for feature extraction (downsampling phase). The principle of downsampling and upsampling operations for FCN are following the work presented in[66;131]. Our network architecture is partly based on the VGG-16 network[138] which is converted to fully convolutional networks. We replace three fully connected layers with convolutional layers (B6, B7) for retaining spatial information. For this network, the feature maps generated in the phase of convolution operations (downsampling) are reduced

in dimension, so they need to be reconstructed by performing upsampling. The process of recovering the spatial dimension of original document is represented by "UpConv" block (upsampling operations) in Figure 4.13 wherein the transposed convolution is used for retaining the spatial information of document. Besides, we apply Softmax layer on the output to transform the result of network into a two-class problem for representing the probability of document's watermarking regions. As a result, the feature maps with the same dimension than the input handwriting document are produced at the output layer. The convolutional layers of each block of the network are briefly described in Table 4.1, and we use the notation of (kernel, stride) and (kernel) to define the convolutional layers.

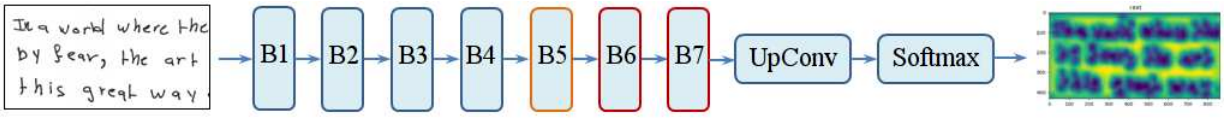


Figure 4.13: *The architecture of FCN for detecting the watermarking regions.*

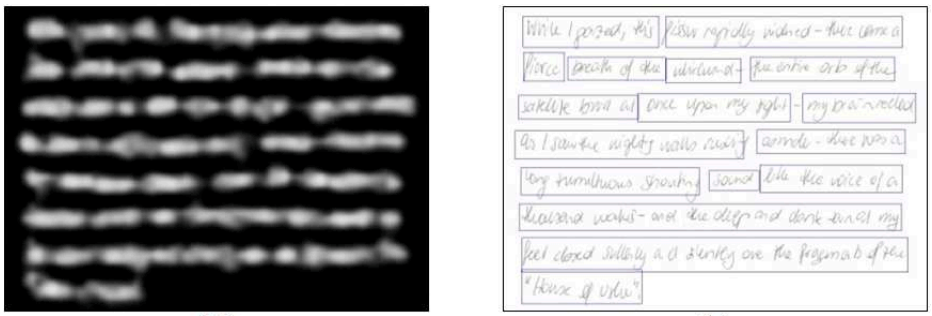
Table 4.1: *Description for convolutional operations of each block*

Blocks	Convolutional operations in each block
B1, B2	2 conv layers (3×3, 1×1), ReLU, max pooling (2×2)
B3, B4	3 conv layers (3×3, 1×1), ReLU, max pooling (2×2)
B5	3 conv layers (1×1, 1×1), ReLU, max pooling (2×2)
B6	1 conv layer (7×7, 1×1), ReLU, dropout
B7	2 conv layers (1×1, 1×1), ReLU, dropout

The ground truth corresponding to the bounding boxes around the words of the text is carried out automatically relied on gradients of text, which is presented in[149]. The documents along with their ground truth as depicted in Figure 4.12(b) are used to train the network. The feature maps generated from first blocks (B1 - B3) of convolutional layers represent the overall shape of handwriting document content while the regions-specific information is extracted from the last blocks (B4 - B7). The salient map as described in Figure 4.14(a) is then obtained by computing the scores of the output feature maps gained from our trained network. Lastly, depending on the salient map, the bounding boxes (blue rectangles in Figure 4.14(b)) surrounding the content regions of document are easily determined by taking advantage of connected components. We consider these regions as the watermarking regions, and the separated handwriting elements located inside the watermarking regions are regarded as objects. These objects are used to carry the secret information bits.

4.3.2 Watermark hiding process

Figure 4.15 demonstrates the general steps of information hiding process.



(a) (b)

Figure 4.14: *The salient map and watermarking regions: (a) the salient map depicts the content regions of document, and (b) the blue rectangles are the watermarking regions.*

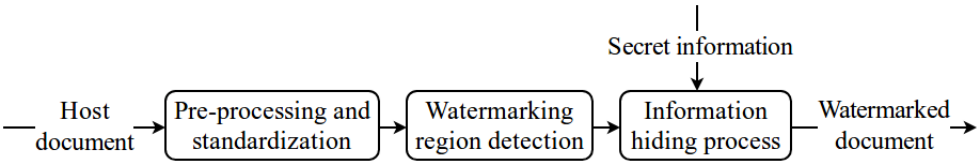


Figure 4.15: *The main steps of information hiding process.*

Step 1 - Pre-processing and standardization of the input document

Due to the stroke of pen and writing style, the content of handwriting documents is often written with irregular ink strokes. Hence, much information is lost during printing and scanning process. The loss often falls into the areas of objects containing gray level values with high intensity, and this could lead to fail in detecting the secret information. To prevent from losing much information, the document is pre-processed by updating gray level values of objects, which are greater than mean value of entire documents content, with the mean value of the document.

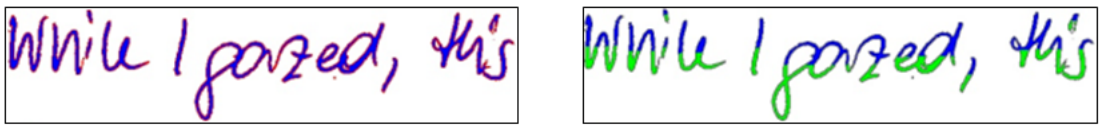
To identify the parameters for geometric correction, the rotation angle θ is effectively determined through a minimum rectangle containing the content of entire document, and the positions of the object's stroke as described in Section 3.3.2 along with convex hull are used for identifying the minimum rectangle. The scale factor s is estimated depending on the Euclidean distance d of two points like a top left point and an intersecting point of two diagonal lines of this rectangle. The document standardization is then carried out by applying affine transformation as in Equation 2.5.

Step 2 - Detection of watermarking regions

This task is conducted as described in Section 4.3.1. The extracted watermarking regions are possibly overlapped or nested together, and this issue has been solved similarly to the method presented in Section 3.3.4.

Step 3 - Hiding secret information bits into the document

By experiment, we have observed that the pixel values situated in the positions of object stroke are much changed when documents undergo the process of printing and scanning. Thus these pixel values are not selected to carry the secret information bits. Only the gray level values corresponding to the filling part of the objects (as detailed in Section 3.3.2) are used to hide information, and they are illustrated as blue color in Figure 4.16(a). To hide a watermark bit into a separated object, we divide the pixel values of the object fill into two sets by processing pixels from left to right and from top to bottom, which is depicted as blue and green color in Figure 4.16(b): $P = \{p_1, p_2, ..., p_m | m = n/2\}$ and $Q = \{p_{m+1}, p_{m+2}, ..., p_n | m = n/2\}$, where n is the number of pixel values of an object fill. Let s_1 be the sum of all pixel values of P, s_2 be the sum of all pixel values of Q, d be the absolute difference between s_1 and s_2, and δ_1 be the minimum distance required between s_1 and s_2. The adjusment of gray level values for carrying the watermark bit is performed by:



(a) (b)

Figure 4.16: *The objects stroke and fill are depicted in red and blue color respectively (a). The two sets P and Q (blue and green color) of each of separated handwriting elements are used for carrying watermark bit (b).*

- if the i^{th} watermark bit $wm_i = 0$:
  - ☐ $s_1 < s_2$ and $d \geq \delta_1$: the pixel values in P and Q keep unchanged.
  - ☐ Otherwise: decreasing pixel values of P by a threshold δ_2, and increasing pixel values of Q by δ_2 such that $s_1 < s_2$ and $d \geq \delta_1$.
- if the i^{th} watermark bit $wm_i = 1$:
  - ☐ $s_1 > s_2$ and $d \geq \delta_1$: the pixel values in P and Q keep unchanged.
  - ☐ Otherwise: increasing pixel values of P by δ_2, and decreasing pixel values of Q by δ_2 such that $s_1 > s_2$ and $d \geq \delta_1$.

4.3.3 Watermark detection process

The detection of watermark bits is conducted in a similar fashion as the watermark hiding process wherein the extraction of hidden data does not require the pre-processing task. The watermarked document is first transformed into its standard form in order to minimize the geometric distortions. The geometrically corrected document is then fed into the fully convolutional networks, and this network will produce a salient map with the same dimension

of the input document. The generated salient map is used to help identify the content regions of document, which are known as the watermarking regions, where the secret information will be extracted. Each separated handwriting element within the watermarking regions is divided into two sets, and the sums of pixel values s_1 and s_2 corresponding to these two sets are used to detect the hidden information bits. The information bit wm_i is extracted by:

$$wm_i = \begin{cases} 0, & \text{if } s_1 \leq s_2 \\ 1, & otherwise \end{cases} \tag{4.7}$$

The accuracy ratio of watermark extraction is measured by Equation 3.13.

In comparison with the existing techniques such as signature determination, writer verification and ink verification, the data hiding technique also gives high performance for tracing the origin of genuine documents. The proposed scheme is capable of verifying the document origin without requiring any reference information as the state-of-the-art approaches presented above. However, our approach can not resist to the distortions caused by the process of scanning with low resolutions or other complicated distortions. Thus, the scheme robustness needs to be further improved. The experimental results of this method are detailed in Section 5.5.

4.4 A robust watermarking scheme using generative adversarial networks

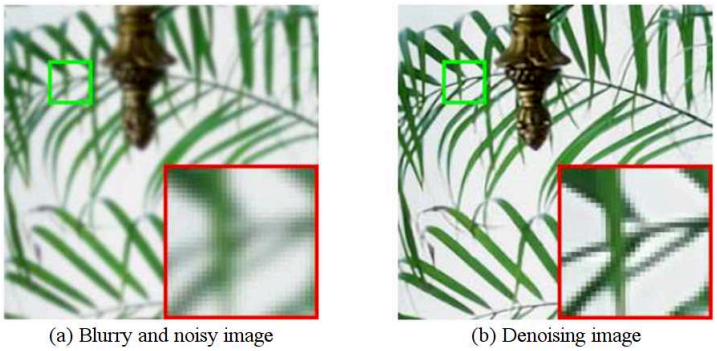
As discussed in the previous works, although the approaches presented in Section 3.3, Section 4.2 and Section 4.3 have significantly improved the robustness against common distortions compared to the feature points-based method presented in Section 3.2, they still have failed in detecting the hidden information from the watermarked documents, which are scanned at the resolution of 400 dpi or lower, or subjected to complicated distortions like print-photocopy-scan. Thus, we are motivated to come up with another solution for scheme improvement. The main idea of this work is to generate a good quality document from an input document, which is robust enough to distortions, and this document is regarded as a reference for building the watermarking system. The good quality documents are obtained by utilizing generative adversarial networks (GAN).

Prior to presenting the detail of our watermarking scheme, we would like to introduce some prominent techniques which are recently applied to generate a good quality image from an input one. These techniques are image restoration, image reconstruction, super-resolution image reconstruction and image-to-image translation.

Image restoration is a process for reconstructing a clean image from a degraded observation (e.g. noise, blurring, and sampling). Kim *et al.*[150] have proposed a CNN-based scheme for learning the regularizer of the alternating minimization algorithm. In this work, the

authors introduce the aggregated mapping in the alternating minimization algorithm, which produces better restoration model than the conventional pixel-wise mapping. The idea of alternating minimization algorithm is to solve the problem of formulating a data term for the degraded observation and a regularization term for image to be reconstructed. Their network consists of three sub-networks such as guidance network, parameter network and deep aggregation network. The guidance network is accountable for considering structures of both input and guidance images. Another framework based on fully convolutional and deconvolutional layers[151] is used for image restoration in which the convolutional layers work as feature extraction, which obtain the spatial information of image contents while eliminating noises, and the deconvolutional layers are used to recover the image details. However, when the network goes deeper or using operations like max pooling, the deconvolutional phase does not work well because lots of information are lost during the convolutional operations. To address this issue, the authors utilize skip connection between corresponding convolutional and deconvolutional layers. The feature maps passed by skip connection hold much image detail, so this helps the deconvolutional process recover a better image.

Chen *et al.*[152] have proposed a learning framework based on learning optimal nonlinear reaction diffusion model. With this approach, the training parameters such as filters and influence functions are learned from training data in a supervised manner. The authors apply feedback step in their diffusion network rather than conventional feed-forward networks. The nonlinearity (influence function) used in their network is trainable whereas the activation function (ReLU or sigmoid function) used in conventional convolutional networks is fixed. Zhang *et al.*[7] have put forward a deep CNN denoiser prior based optimization method in which the authors make use of dilated convolution, instead of normal convolution, to make a tradeoff between the size of receptive field and network depth. The dilated convolution is known for expanding capacity of the receptive field while maintaining the merits of normal convolution. Besides, the batch normalization and residual learning have been adopted in their network, which enable the network to result in better denoising performance. The authors have pointed out their network to be applicable to image denoising, image deblurring and single image super-resolution. Figure 4.17 illustrates an example of image deblurring.



(a) Blurry and noisy image (b) Denoising image

Figure 4.17: *The illustration of image deblurring*[7].

Another method based on regularization[153] is capable of handling arbitrary degradations such as blur, missing pixels, etc. To do so, the authors rely on the tendency of small patches to recur within natural images, and they do that by adopting the weighted nuclear norm

minimization which has been shown to obtain better results in image denoising. In addition, the authors have pointed out the fact that the small patches tend to recur not just within the same scale but across different scales in the natural images. Finally, instead of formulating an independent reconstruction for each patch group, the authors apply regularization that take all patch groups by using the expected patch log-likelihood approach. Yoo *et al.*[8] have proposed a method for restoring the details of corrupted image, which is due to the loss caused by JPEG compression. The authors consider the task of image restoration as a task of classification. The frequency distribution of target image is directly estimated from the input image by using the cross-entropy loss function. The network for image restoration consists of three subnets namely a classifier, an encoder and a decoder. The classifier network produces a discrete distribution containing probability of frequency coefficient class for each frequency channel. The cross entropy loss is used to train the classifier network. The encoder network is used to generate feature map, and the decoder network is used to produce output image. Figure 4.18 demonstrates the restoration of compressed image with JPEG quality factor of 10.

Figure 4.18: *The restoration of JPEG compressed image*[8].

Image reconstruction: Similar to image restoration, it refers to recovering the clean image from the corrupted image (e.g. blur, low resolution). CNN-based method[154] has been proposed to reconstruct image from incomplete measurement data. This network is trained to learn the mapping between the true (original) image and the artifact (degraded) image. This mapping is embbed within an iterative reconstruction method that could result in reducing artifact levels. Instead of learning a direct mapping, this network predicts a residual image which is added to the original image to get the output image. Besides, the authors train their network with two-stage process in which the network is first trained with zero-initialized reconstructed images, the weights are then fine-tuned with the weights obtained in the first stage. Another CNN-based method[9] has been proposed to reconstruct high dynamic range (HDR) images from low dynamic range (LDR) images which are captured with arbitrary and low-end cameras. This network is designed in the form of autoencoder network that is adjusted to operate on the LDR images and produce the HDR images in which the encoder side operates directly on LDR input image, and the decoder side is accountable for producing HDR output image. In addition, the authors use skip connection in their network to transfer each level of information from the encoder side to the corresponding level of information from the decoder side. The reconstruction of corrupted image is shown in Figure 4.19.

Figure 4.19: *The reconstruction of corrupted image*[9].

Cheng[155] *et al.* have proposed a CNN-based method for reconstructing image from sub-sampled acquisition in the spatial-frequency domain in which the network is applied to patches of data in the frequency domain. It means that a bandpass filter is used to select and isolate the reconstruction to small localized patches in the frequency space (k-space). With this approach, the training and inference process are carried out on localized patches of k-space. The network consists of two different blocks such as update block and de-noising block. The update block is critical to ensure that the final reconstructed image accords with the measured data samples. The de-noising block consists of a number of convolutional layers to de-noise effectively images.

Super-resolution image reconstruction aims to reconstruct a high resolution image from a low resolution input image. Kim *et al.*[156] have come up with a deeply-recursive convolutional network. The feature maps after each recursion, which are supervised, are used to reconstruct the output high resolution image. The authors use skip connection to transfer information from input to the reconstruction layer. Their network consists of three sub-networks: embedding, inference and reconstruction network. The embedding network takes the input image and generate a set of feature maps, which is passed to the inference network. The inference network is the main component to solve the issue of super-resolution, and it contains a recursive layer. The feature maps obtained from the final application of recursive layer represent the high resolution image. The reconstruction network is responsible for transforming these feature maps back to the original image space. Laplacian pyramid network-based method[157] has been proposed for reconstructing the sub-band residuals of high resolution images in which a cascade of convolutional layers is used to extract feature maps. This network takes an input image and predicts residual images at $log_2(s)$ level where s is the scale factor, and it consists of two main processes. For feature extraction, at a level s, the network consists of a number of convolutional layers and a tranposed convolutional layer to upsample the extracted features by a scale of 2. The output of each transposed convolutional layer is connected to a convolutional layer for reconstructing the residual image at level s, and a convolutional layer for extracting features at the finer layer $s+1$. For image reconstruction, the upsampled image is combined with the predicted residual image from the feature extraction process to produce a high resolution image. The output image at level s is fed into the image reconstruction of level $s+1$.

The dual-state recurrent network-based method[158] adopts two current states that enables to use features obtained from both low resolution (LR) and high resolution (HR) space. With this approach, the bottom state captures information at LR space while the top state operates

in HR space. There is a connection from the bottom to the top state via deconvolutional operations. Besides, the authors use a delayed feedback mechanism to allow information flow from the top state to the bottom one. This allows LR and HR signals joint together to learn the mapping. Wang *et al.*[159] have proposed a method based on spatial feature transform layer which is capable of transforming the features of some intermediate layers of the network. The spatial feature transform layer is conditioned on semantic segmentation probability maps, so it can generate a pair of modulation parameters to apply affine transformation spatially on feature maps of the network. The authors turn out that reconstructing high resolution image with rich semantic regions can be obtained by a single forward pass through transforming the intermediate features of a single network.

Another method based on a combination of residual block and dense block[160] enables to extract abundant local features via dense connected covolutional layers. This network consists of four parts: shallow feature extraction, residual dense blocks, dense feature fusion and the upsampling network. The residual dense block consists of dense connected layer and local feature fusion with local residual learning. Besides, the authors use mechanism of contiguous memory for passing the state of preceding residual dense block to each layer of current block. The output of one block has direct access to each layer of the next block. Each convolutional layer in block has access to all the subsequence layers and passes on the information that needs to be maintained. CNN-based method[161] has been proposed to solve the problem of multiple degradations via a single model. This network enables to take low resolution image along with its degradation maps as input in which the degradation maps are obtained by a simple dimensionality stretching of the degradation parameters like blur kernel and noise level. The authors use a cascade of convolutional layers to perform the non-linear mapping in which each layer is composed of three operations such as convolution, rectified linear units and batch normalization except for the last convolutional layer. In addition, a sub-pixel convolutional layer is followed by the last convolutional layer, which is used to convert multiple high resolution subimages to a single high resolution image. Figure 4.20 shows the result of high resolution image obtained from a degraded one.

(a) Degraded (b) High resolution image

Figure 4.20: *The high resolution image constructed from the low resolution one with variant degradation*[10].

The method based on generative adversarial network (GAN)[162] is capable of inferring photo-realistic natural images for 4× upscaling factors. The authors have adopted a deep residual network with skip connection and diverge from mean squared error. To enhance

the performance of their generator network, the authors have proposed a perceptual loss function which consists of an adversarial loss and content loss in which the content loss is calculated by using high level feature maps of the VGG network. The objective of this network is to train a generator network with the goal of fooling a differentiable discriminator that is trained to distinguish super-resolved images from real images. With this approach, the generator network can learn to produce images which are similar to the real images, and thus it is difficult to classify by the discriminator network.

Image-to-image translation aims at learning a mapping that can convert an image from a source domain to a target domain, while preserving the main presentations of the input images. Isola *et al.*[11] have proposed a method based on conditional adversarial network. This network enables to learn the mapping from input image to output image, and it is also learn a loss function to train this mapping. The network consists of generator and discriminator, and both of them uses modules of the form convolution-batch normalization-ReLU. Figure 4.21 depicts the result of converting a satellite image into a map. The conditional dual GAN method[163] makes use of the strengths of GAN network and dual learning. The architecture of this network is based on the principle of encoder-decoder network in which the encoder side is responsible for extracting the domain-independent and domain-specific features, while the decoder side is accountable for merging the two kinds of features to generate images. The dual learning helps extract and merge the domain-independent and domain-specific features by minimizing the construction errors.

Figure 4.21: *An example of image translation*[11].

A method based on deep attention GAN[164] enables to decompose the task of translating samples from two independent sets into translating instances in a highly structured latent space. This network consists of four components such as a deep attention encoder, a generator and two discriminators. The deep attention encoder decomposes the original image into instances, which makes it possible to find correct semantic alignments and exploit geometric changes. The constraint of instance level enables the mapping function to find the meaningful semantics. Another method for inferring three dimensional plant branch structures has been proposed in[165]. The inference of the probability of branch existence is conducted by using Bayesian deep learning framework, which is applied to each of multiple view images. Choi *et al.*[166] have proposed a method that can perform translation for multiple domains using only a single model. This method is based on GAN in which the generator network is capable of learning mapping among multiple domains, and the auxiliary classifier is used to enable the discriminator network to control multiple domains.

From this short survey of existing approaches, we have adopted the generative adversarial network to produce a high quality document from a degraded input document, which is used to enhance the performance of detecting document content as well as to be a reference for developing watermarking algorithm. This kind of network has been proposed by Goodfellow *et al.*[167]. GAN gives high performance because it can learn to mimic any distribution of data. The network presented in Figure 4.22 comprises two sub-networks: a generator (G) and a discriminator (D). The generator network's goal is to produce data that is indistinguishable from the training dataset. The discriminator network's goal is to correctly determine whether a particular sample is real which comes from the training dataset or fake which is generated by the generator network. For each classification it makes, the network is given feedback whether its decision was correct or not. The whole process of the network is like these two networks competing each other while cooperating at the same time.

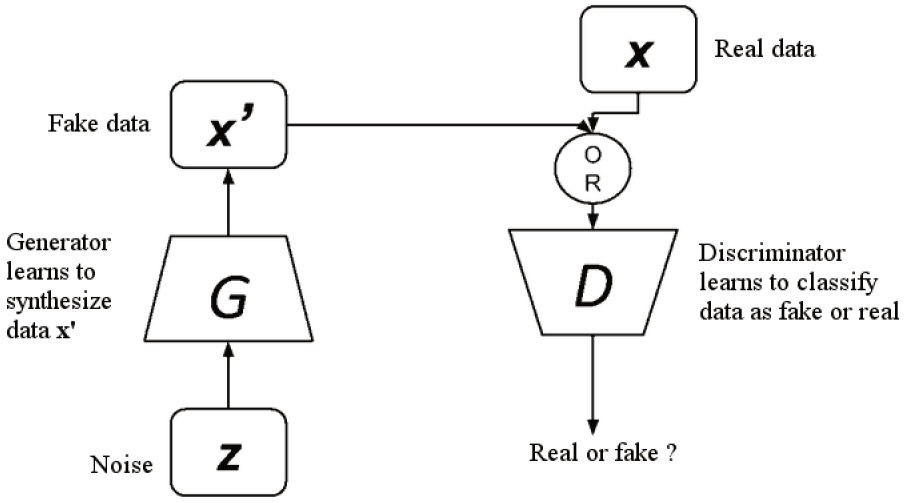


Figure 4.22: *GAN architecture: the generator (G) learns to generate fake data that can eventually fool the discriminator, and the discriminator (D) is trained to distinguish between real and fake data.*

The input of the generator network is noise, and the ouput of this network is synthesized data. Meanwhile, the input to the discriminator network is either synthesized or real data. The genuine data come from the true sampled data whereas the fake data are produced by the generator network. It means that all valid data are labeled by 1 (100 percent of probability to be real), and all synthesized data are labeled by 0 (0 percent of probability to be real).

The process of training the generator network consists of the following steps: The network takes a random real data from the training dataset and random noise vector, or a distorted data from the training dataset. It tries to produce a fake sample as its output. This network uses the discriminator network to classify the generated sample (fake sample), and it is trained by receiving feedback from the discriminator network. The generator network computes the classification error and backpropagates the error to update the generator weights and biases in order to maximize the discriminator error. Meanwhile, the process of training the discriminator network comprises of the main steps as follows: The network

takes a random real sample from the training dataset and the generated data obtained from the generator network. It tries to classify the real sample and generated sample. The discriminator network computes the classification errors and backpropagates the total error to update the discriminator weights and biases in order to minimize the classification errors. In general, the GAN network works similar to the situation of a zero-sum game in which one player's gains are equal to another's losses. It means that when one player gets better off by some amount, the other player gets worse off by the exact same amount. The generator and discriminator networks can reach their situation at which neither player can improve his or her situation when the fake samples generated by the generator network are not distinguishable from the real data, and the discriminator network can at best classify whether a particular sample is real or fake.

The discriminator network can be trained by minimizing the loss function by:

$$\mathcal{L}^{(D)}(\theta^{(G)}, \theta^{(D)}) = -E_{x\sim P_r}\left[logD(x)\right] - E_{z\sim P_z}\left[log(1 - D(G(z)))\right] \tag{4.8}$$

where $E_{x\sim P_r}$ is an expected value from the real data distribution, and $E_{z\sim P_z}$ is a value from the noisy data distribution. $\theta^{(G)}$ and $\theta^{(D)}$ are the parameters of generator and discriminator network respectively.

The loss function of the generator network is simply the negative of the discriminator loss function, and it is given by:

$$\mathcal{L}^{(G)}(\theta^{(G)}, \theta^{(D)}) = -\mathcal{L}^{(D)}(\theta^{(G)}, \theta^{(D)}) \tag{4.9}$$

However, the main problem with this generative model is vanishing gradient because the gradient of the discriminator network not only provides to itself but also provides to the generator network as a feedback. If the discriminator network gets stronger quickly, the gradient of the loss function is down to 0. Otherwise, if the discriminator network gets too weakly, the generator network does not provide a good feedback to the discriminator network. To address these problems, the loss function of the generator network can be adjusted by:

$$\mathcal{L}^{(G)}(\theta^{(G)}, \theta^{(D)}) = -E_{z\sim P_z}\left[logD(G(z))\right] \tag{4.10}$$

The loss function is used to maximize the performance of the discriminator network in which the network recognizes that the generated data is real by training the generator network. The parameters of the generator network are updated only when the whole adversarial network is trained where the gradients are passed from the discriminator network to the generator network.

4.4.1 Document generation for watermarking process

Our observations have shown that the content detection of a document and the performance of the watermarking algorithm decrease when the distorted documents are processed. This is why we are motivated to come up with a solution in order to produce a good quality document from the distorted document. The possible solution to address this concern is to generate an intermediate document from the input document, which is robust enough to distortions, and this document is regarded as a reference for building the watermarking system. The research fields which have inspired us are super-resolution and image denoising as presented above wherein the former enables to recover a high-resolution image from a low-resolution input image, and the latter is to remove noises from the input images. Taking advantage of these techniques, we construct a network based on the principle of GAN[167] for generating a desired document from a given document with or without distortions. Our network consists of three main parts including a generator network G, a discriminator network D and a combination of modified loss functions. The brief explanation of our network is presented below.

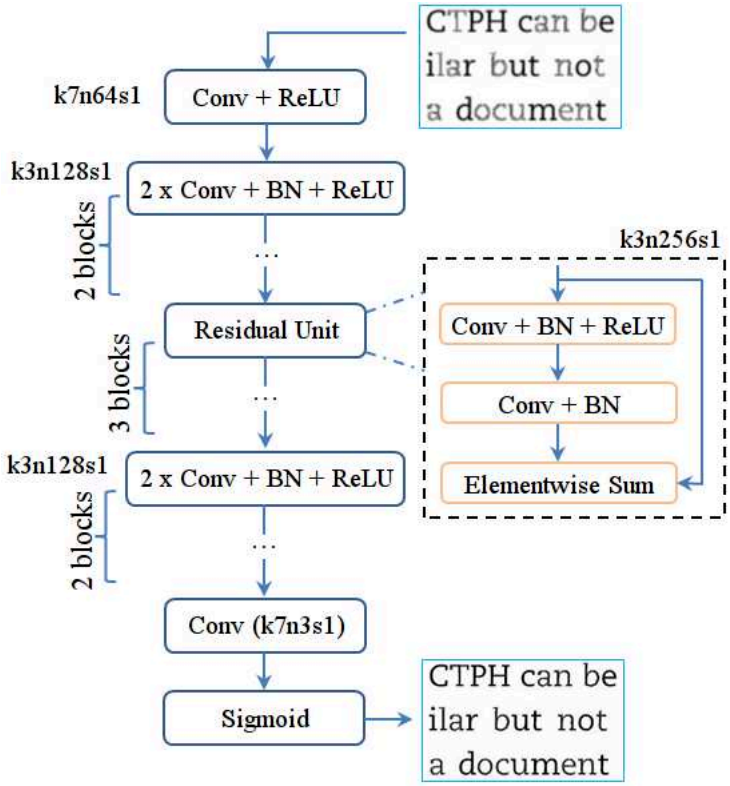


Figure 4.23: *The architecture of generator network in which k is the kernel size, n is the number of feature maps, and s is the stride in each convolutional layer (e.g. k7n64s1 means the convolutional layer has* 64 *kernels with size* 7 *and stride* 1*).*

The generator network as shown in Figure 4.23 is used to map the input document to its manifold from the training distribution. This kind of network in our model is trained in an end-to-end manner by using a combination of adversarial loss, Euclidean loss and feature loss. The network begins with a block of one convolution followed by two blocks

of convolutional operations in order to spatially downsample and encode the documents. Afterwards, three blocks of residual unit with identical structure are used to generate the content and manifold features. The structure of residual unit is presented as a dash line rectangle on the right side of the network. The batch normalization (BN) is utilized to keep the deep model working without falling into collapse mode. It is a situation in which the generator network creates samples with very low diversity. In other words, this network returns the same aspect samples for different input signals. Finally, the generated document is reconstructed from the obtained features by two blocks of de-convolution, a convolutional layer with kernel of 7×7 and a sigmoid layer (for pixel-level prediction). The combination of convolutional, residual and de-convolutional layers allows the network to preserve the features of the document while eliminating possible distortions. For instance, "k7n64s1" means the convolutional layer has 64 kernels with size 7 and stride 1 where k, n and s are stand for kernel, the number of filters and stride respectively.

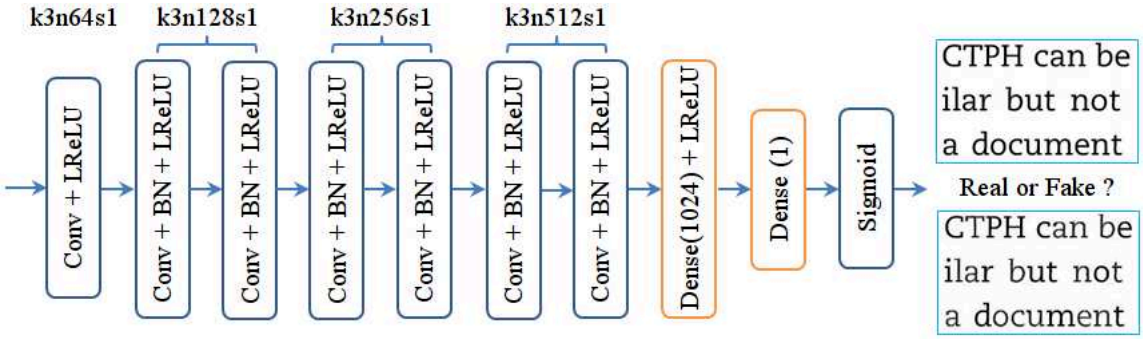


Figure 4.24: *The architecture of discriminator network: from left, the number of kernels of covolutional layer 2 and layer 3 is 128; 256 for layer 4 and layer 5; 512 for layer 6 and layer 7.*

The discriminator network takes either real document or document generated from the generator network as an input. It tries to predict whether the input is a real or generated document. This network solves a problem of binary classification and gives an output with a scalar value between 0 and 1. The architecture of the discriminator network consists of convolutional layers, fully-connected layers and a sigmoid activation function as presented in Figure 4.24.

Given a pair of documents (X, Y) with width W and height H in which X is the watermarked or distorted document, and Y is its corresponding real document. The loss function used for obtaining better gradient behavior is defined as follows:

$$\mathcal{L}_G = -\frac{1}{N}\sum_{i=1}^{N} log(D(G(X_i))) \tag{4.11}$$

$$\mathcal{L}_D = -\frac{1}{N}\sum_{i=1}^{N} (log(D(Y_i)) + log(1 - D(G(X_i)))) \tag{4.12}$$

where $D(.)$ is discriminator output, $G(.)$ is generator output, and N is a set of generated documents.

The feature-based loss $\mathcal{L}_F$ is defined by the squared and normalized Euclidean distance of the high level feature maps extracted at layer relu1_2 as in the pretrained model presented in Section 4.2.1 between the generated and real documents (transformed by a non-linear activation function). This layer level is selected because the feature representations obtained from low layers (relu1_2) tend to produce an output that keeps content and spatial structure close to the input document.

$$\mathcal{L}_F = \frac{1}{WH}\sum_{i=1}^{W}\sum_{j=1}^{H}\|\phi_i(G(X^{i,j})) - \phi_i(Y^{i,j})\|_2^2 \tag{4.13}$$

where ϕ_i is the activation at the i^{th} layer of the network, and it represents a non-linear transformation.

The pixel loss (per-pixel Euclidean loss) $\mathcal{L}_E$ is defined by the normalized Euclidean distance between the intermediate document generated by the generator network and its real document.

$$\mathcal{L}_E = \frac{1}{WH}\sum_{i=1}^{W}\sum_{j=1}^{H}\|G(X^{i,j}) - Y^{i,j}\|_2^2 \tag{4.14}$$

The total loss function $\mathcal{L}$ for G is then defined by:

$$\mathcal{L} = \alpha_g\mathcal{L}_G + \alpha_e\mathcal{L}_E + \alpha_f\mathcal{L}_F \tag{4.15}$$

where α_g, α_e and α_f are weighting parameters.

Figure 4.25 shows the generated documents by using our proposed network: (a) and (b) are input documents, and (a1) and (b1) are their corresponding generated documents. We can see that the quality of generated document from distorted document has been significantly improved compared to the input one ((a) vs (b): peak signal to noise ratio (PSNR) = 18.41, structural similarity index measurement (SSIM) = 0.93; (a1) vs (b1): PSNR = 25.46, SSIM = 0.97). The generated documents are then used as reference for developing our watermarking algorithm.

4.4.2 Data hiding process

Figure 4.26 depicts the general process of watermark hiding. Hiding information into document is sequentially conducted as follows:

– Step 1 - Transforming an input document into its correct form (I_c) is carried out as

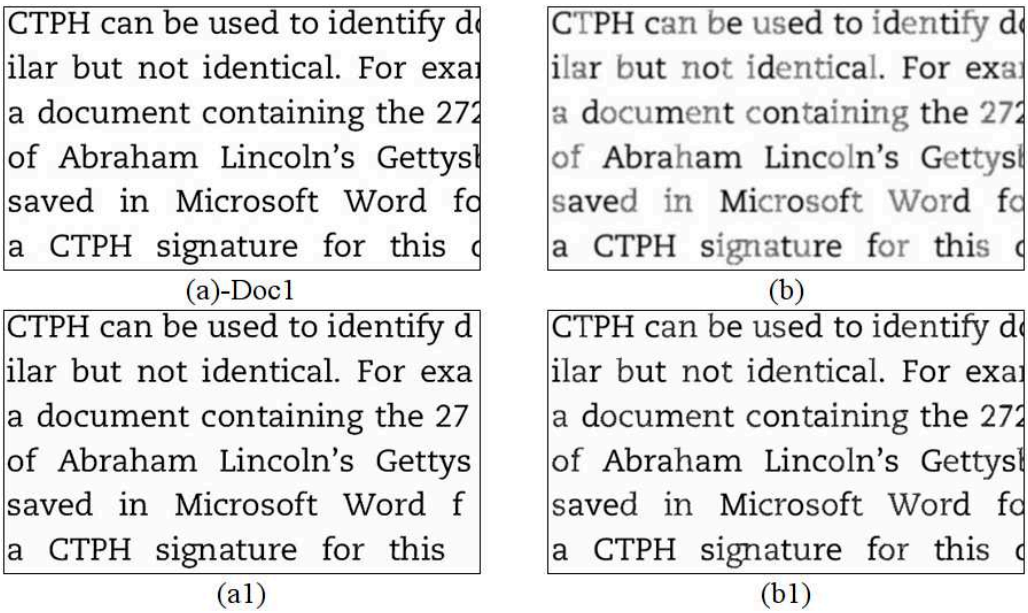


Figure 4.25: *The illustration of generated documents: (a) and (b) are input documents, and (a1) and (b1) are generated documents.*

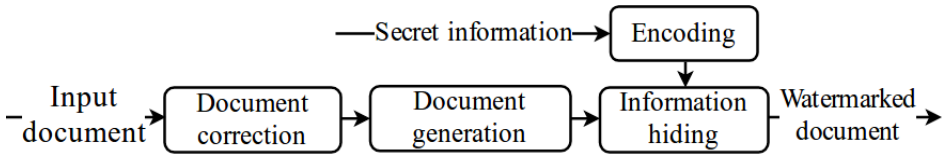


Figure 4.26: *The main steps of information hiding process.*

described in Section 3.2.2. We make use of Hough transform to detect lines which are parallel with the text lines of document, and the rotation angle of document is determined based upon an appropriate line.

- Step 2 - Generating a referenced document (I_g) from an input document, which is used as an enhancement of detecting document content and a reference for watermarking process, is detailed in 4.4.1.

- Step 3 - Information encoding for enhancing security feature is conducted in a similar manner as presented in Section 4.6.3.

- Step 4 - To hide watermark bits into document, we make use of optical character recognition (OCR) to detect bounding boxes of objects within document. By experiments, we have observed that OCR gives the extraction of bounding boxes more stable than other methods in terms of distorted documents. Each bounding box containing an object is used to carry a watermark bit by:

  - if the i^{th} watermark bit $wm_i = 0$:

    ☐ Replacing the i^{th} bounding box within the document I_c with its corresponding

bounding box within the document I_g.

- if the i^{th} watermark bit $wm_i = 1$:
  - ☐ Replacing the i^{th} bounding box within the document I_c with its corresponding bounding box within the document I_g.
  - ☐ Increasing the gray level values of the i^{th} object in the document I_c by a threshold λ.

4.4.3 Data detection process

The main steps of watermark detection are conducted in a similar manner as the watermark hiding process just different in extracting hidden information and decoding the extracted information for recorvering it to a meaningful information. To do so, for each eight bounding boxes, we measure the absolute distance of pixel values between each object in I_c and its respective object in I_g. We choose groups of eight bounding boxes because it is enough to balance the minimum, maximum and average pixel values of corresponding characters within these boxes for extracting the secret bits. The absolute distance d_i of the i^{th} pair of objects is computed by:

$$\begin{cases} s_1 = \sum_{j=1}^{n} p_j; s_2 = \sum_{j=1}^{n} q_j \\ d_i = |s_1 - s_2| \end{cases} \tag{4.16}$$

where p and q are pixel values of an object in I_c and I_g respectively, and n is the number of object's pixel values.

The set of absolute distances of each eight bounding boxes is depicted by $D = \{d_1, d_2, ..., d_8\}$. Assume d_{min} and d_{max} are the minimum and maximum values in D, d_{avg} is the average value of d_{min} and d_{max}. The i^{th} watermark bit is then determined by:

$$wm_i = \begin{cases} 0, & \text{if } d_i < d_{avg} \\ 1, & otherwise \end{cases} \tag{4.17}$$

The accuracy ratio of watermark extraction is measured by a ratio between the number of correctly extracted watermark bits and the total number of hidden watermark bits.

In summary, we have proposed another watermarking scheme based on the idea of generating a reference document from the input one for watermarking process. The watermark hiding algorithm is developed relied upon the changing of pixel intensities. Our approach has significantly improved the scheme robustness compared to our previous approaches. Specifically, the scheme can detect the hidden information when the watermarked documents are scanned at low resolutions, or subject to two rounds of photocopying before scanning at

various resolutions. The details of our experiments are presented in Section 5.6.

4.5 Watermarking scheme based on font generation

In this section, we introduce another watermarking framework for securing legal documents, which is relied upon font generation or variations of document characters. Unlike the data hiding schemes based on changing pixel intensities, a few approaches have been proposed by using variants of characters as presented in the works[53;54] and SOOD company[1]. The schemes based on variations of characters[53;54] give high resistance to complicated distortions like PS and print-and-photograph. Howerver, this scheme requires a codebook to store characters and their variations, which are used as a reference for prediction (CNN network) of extracted information. The variations of a character are generated by using font manifold presented in[72]. Another concern with this method is that it seems to be hard to build a general codebook for characters with different styles and fonts. Different from these approaches, we generate variations of characters by utilizing generative adversarial networks (GAN) for hiding secret information, and we have adopted fully convolutional networks (FCN) to detect the hidden information from the watermarked documents (without any reference information). As a result, our proposed method is able to detect the hidden information in case of the watermarked documents scanned at low resolutions and subjected to two rounds of photocopying.

4.5.1 Generation of character variations using GAN

To address concerns mentioned in the existing schemes, we are going to develop a watermarking framework which is independent of fonts, styles of characters and symbols used in the documents. For simplicity, we refer to the document characters and symbols as characters. The research field which has inspired us is the image-to-image translation as presented in Section 4.4. Taking advantage of these techniques, we construct a network based on the principle of GAN[167] for the purpose of generating a desired variation of characters from a given document. This kind of network is applicable to produce new font or variation of characters from the input document with arbitrary font and style. The information of a character can be described by its skeleton and normal shape. Thus, these two features are used to train the network, and the trained network is capable of generating the variants of a character from its skeleton. We utilize the thinning algorithm presented in[168] to make the skeleton of a character. Our network consists of three main parts including a generator network G, a discriminator network D and a combination of modified loss functions. The skeleton feature and shape feature are obtained at the output of the last block of downsampling convolutional layers in the network G, and these features are then concatenated as the input of the remaining part of the network, which has several upsampling convolutional layers.

[1] http://sood.fr/en/patents/sood_tattoing.html

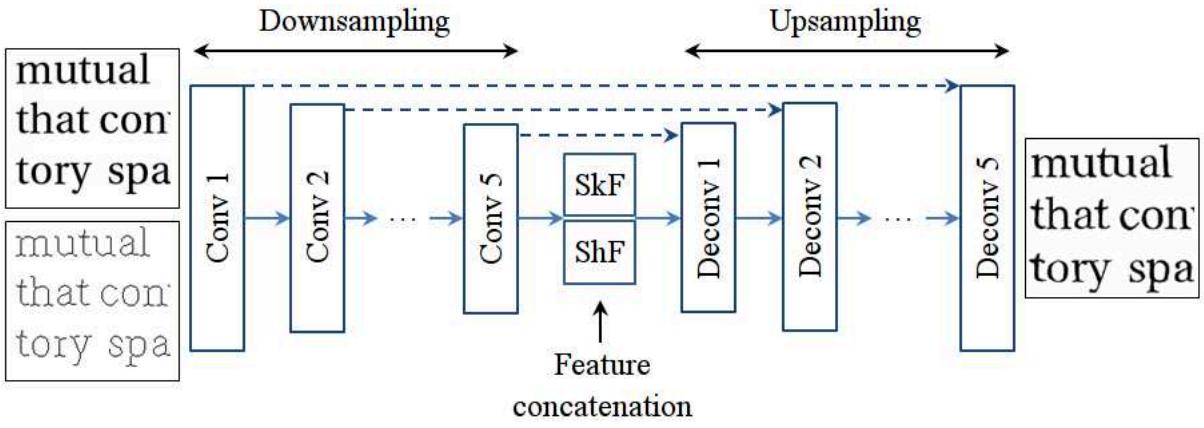


Figure 4.27: *The architecture of generator network for producing character variation in which "SkF" and "ShF" stand for skeleton feature and shape feature respectively.*

The generator network as shown in Figure 4.27 is used to map the input document to its manifold from the training distribution. This kind of network in our model is trained in an end-to-end manner by using a combination of modified adversarial loss and feature loss. The structure of our network is symmetrical by applying successively downsampling and upsampling process. There is a connection between each pair of processes (dash arrow in Figure 4.27) which enables the feature maps of downsampling block to its corresponding upsampling block. The convolutional layers in each downsampling block are partly based on the VGG 16-layer network[138], and they are briefly described in Table 4.2. We use notation of (kernel, stride) and (kernel) to define the convolutional layers.

Table 4.2: *Convolutional operations of downsampling blocks*

Downsample	Convolutional operations in each block
1, 2	2 conv layers (3×3, 1×1), ReLU, max pooling (2×2)
3, 4	3 conv layers (3×3, 1×1), ReLU, max pooling (2×2)
5	3 conv layers: (1×1, 1×1), (3×3, 1×1), (1×1, 1×1) ReLU, max pooling (2×2)

The discriminator network is used to distinguish between the real documents and generated documents synthesized by the generator network. The architecture of this network is similar to the discriminator network presented in Section 4.4.1.

In the process of network training, the downsampling blocks of the generator network are used to generate the feature of character's skeleton and the feature of character's normal shape. These two features are then concatenated and fed the upsampling blocks of the network for generating character variation. Meanwhile, in the inference process, the network takes only skeleton document as an input. The loss function for the discriminator network is defined by Equation 4.12. Meanwhile, the total loss function for generator network is defined by:

$$\mathcal{L} = \alpha_a \mathcal{L}_G + \alpha_f \mathcal{L}_F \tag{4.18}$$

where α_a and α_f are weighting parameters to determine a tradeoff between the adversarial loss and the feature loss. $\mathcal{L}_G$ and $\mathcal{L}_F$ are adversarial and feature loss corresponding to Equation 4.11 and Equation 4.13.

Figure 4.28 demonstrates the generated characters by using our proposed network in which (a) is the original document, and (b), (c) and (d) are the corresponding character variations. These variants are obtained by combining the skeleton feature with the various shape features.

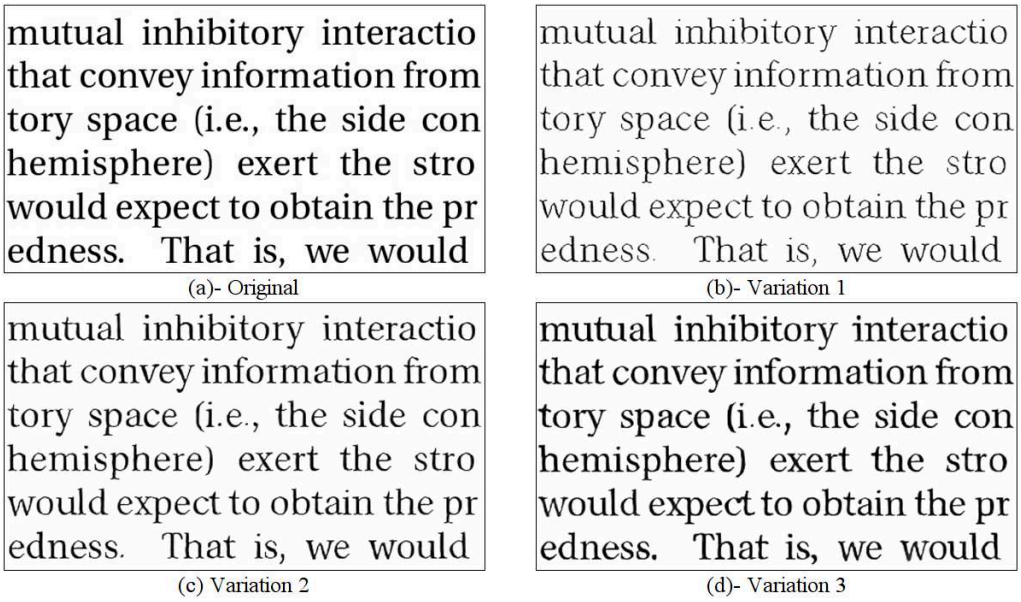


Figure 4.28: *The generated characters: original characters (a), and their variants (b), (c) and (d).*

4.5.2 Detection of character variations using FCN

To detect the regions of character replaced by their variations from the watermarked documents, we adjust FCN[66] for the purpose of image semantic segmentation to deal with the problem of detecting character variation. The adjustment has been performed by rectifying convolutional layers for better feature extraction and representation from the watermarked documents. The network presented in Figure 4.29 consists of two main stages: downsampling and upsampling operations. The downsampling process contains convolutional layers, elementwise activation function (ReLU) and max pooling as described in the architecture of generator network, and followed by two blocks of convolutional layers: (1 × conv layer (7×7, 1×1) + ReLU + dropout) and (2 × conv layers (1×1, 1×1) + ReLU + dropout). Meanwhile, the upsampling process ("UpConv") consists of few convolutional layers instead of

symmetrical convolutional layers as in the generator network. It is responsible for recovering the spatial dimension of the original document wherein the transposed convolution layers are used for maintaining the spatial information. Besides, we apply Softmax function at the output layer to transform the result of the network into a two-class problem for representing the probability that the characters are replaced by their variations. As a result, the feature maps with the same dimension of its input document are produced at the output layer.

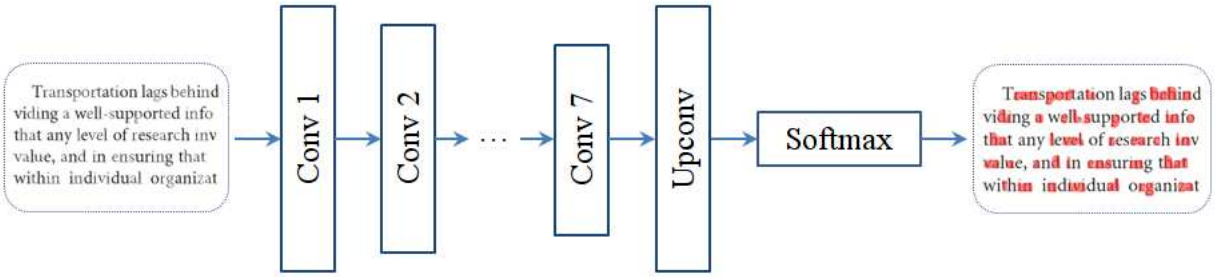


Figure 4.29: *The architecture of FCN for detecting the character variations.*

The watermarked documents (Figure 4.30(a)) and their ground truths corresponding to the bounding boxes around character variation (Figure 4.30(b)) are used to train the network. We generate the ground truths by marking the positions of group of adjacent variations. This helps the FCN learn features from the character variations better than the ground truths made by marking the positions of single variation.

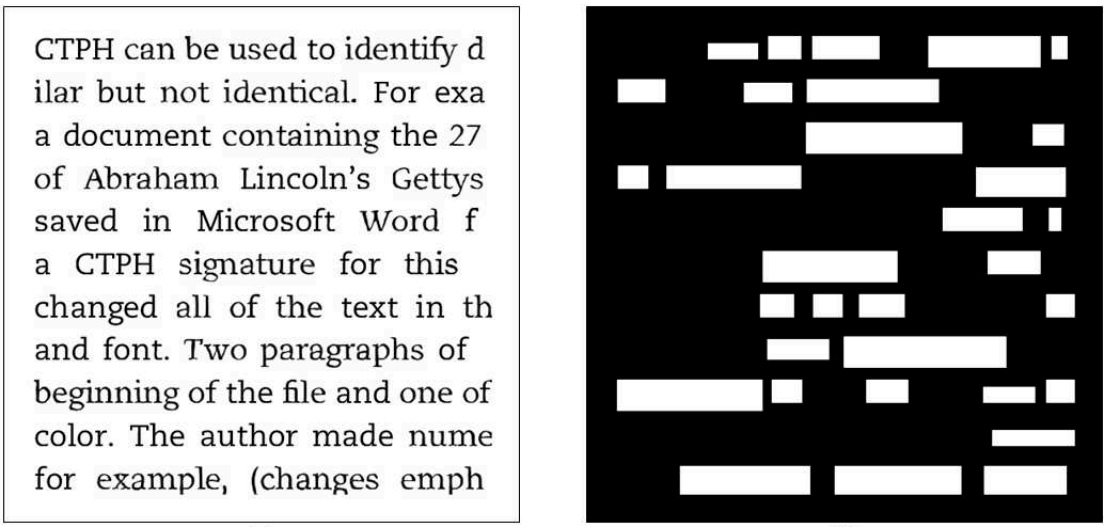


Figure 4.30: *An example of a watermarked document (a) and its corresponding ground truth regions (b).*

Figure 4.31(a) depicts in color the salient regions, which are obtained by computing the scores of the output feature maps gained from the trained network, that describes the positions of characters substituted by their variations.

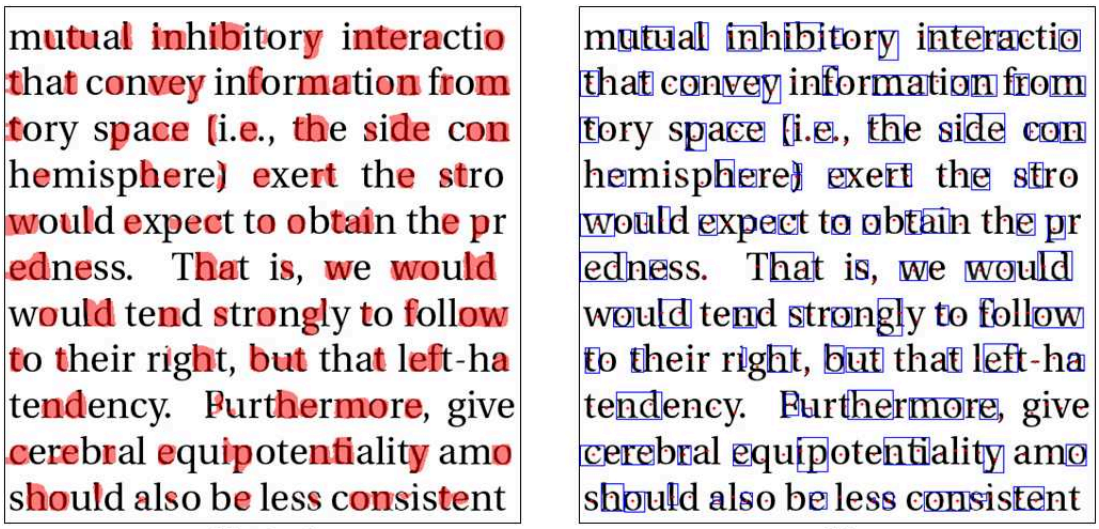


Figure 4.31: *(a) The color regions mark the regions of character variants. (b) The bounding boxes (blue rectangles) correspond to the regions of character variants, the red points correspond to the center of the bounding boxes surrounding each character (note that these bounding boxes are not represented here).*

4.5.3 Generating random positions for watermarking process

In order to enhance the security feature, instead of selecting characters extracted from document in sequence for watermarking process, we randomly chose characters by using pseudo-random numbers. With a seed which is regarded as a private key, this technique enables to generate a series of positive numbers, and these numbers correspond to the positions of the characters in the document, which are used for data hiding and detection. With the same generator of pseudo-random numbers and the same seed, we always obtain the same series of positive numbers. The number of positive numbers is selected to be equal to the number of characters extracted from the document. This approach enables to hide secret information into document in a manner that the secret bits are distributed over the entire document even that the number of secret bits is much less than the document content.

4.5.4 Watermark hiding process

The secret information is converted into a string of bits, and hiding information into the document is sequentially conducted as follows:

- Step 1 - Skeletonizing the document by using thinning algorithm[168].
- Step 2 - Generating the character variants from the skeleton document as described in Section 4.5.1.

- Step 3 - Producing a set of random positions corresponding to document characters as presented in Section 4.5.3.
- Step 4 - To hide secret bits into document, we make use of optical character recognition (OCR) to detect bounding boxes of characters within document. Each bounding box containing a character at respective random position is used to carry an information bit by:
  - ☐ if the i^{th} watermark bit $wm_i = 0$: The character remains unchanged.
  - ☐ if the i^{th} watermark bit $wm_i = 1$: The character is replaced with its corresponding variation, which is generated in step 2.

4.5.5 Watermark detection process

The process of detecting the hidden information is performed with the following steps:

- Step 1 - Transforming the document into its correct direction as depicted in Section 3.2.2.
- Step 2 - Generating the salient regions I_s from the watermarked document I_w as presented in Section 4.5.2.
- Step 3 - Producing a set of random positions with the same seed used in the hiding process.
- Step 4 - Detecting the bounding boxes B_s ($B_s = \{b_j\}_{j=1}^{n}$, where n is the number of bounding boxes) of the salient regions from I_s by utilizing connected component algorithm (blue rectangles in Figure 4.31(b)), and the bounding boxes of characters from I_w by OCR. For each bounding box of a character at its corresponding random position, we compute its center point p_c (red points in Figure 4.31(b)), and the hidden information are then extracted by:

$$wm_i = \begin{cases} 1, & \text{if } p_c \in b_j (b_j \in B_s) \\ 0, & otherwise \end{cases} \tag{4.19}$$

- Step 5 - Converting the extracted bits into text to obtain the meaning information.

The accuracy ratio of watermark extraction is measured by a ratio between the number of correctly extracted watermark bits and the total number of hidden watermark bits.

In this section, we have proposed a new watermarking scheme relied upon the variation of characters, which has been little studied in the literature. Unlike the existing variation-based works, we generate the variants of characters by using GAN, and detect the hidden information from the watermarked documents by utilizing FCN, which does not required

any reference information. The results presented in Section 5.7 show that our approach has ability to detect the hidden information in terms of complicated distortions. However, the watermarking scheme needs to be further improved to obtain higher precision when detecting the hidden information from the watermarked documents whose contents are replaced by the variants closer to the shape of normal characters.

4.6 Watermarking for securing binary documents

As discussed in the previous chapters, the possibility to convert the genuine documents into their digital and readable formats has become a necessity. Scanning or capturing document is a way of changing the printed document into its digital format. The digital versions of documents commonly suffer from various degrations. Specifically, the document images can be heavily degraded due to ink bleed-through, faded ink, wrinkles, stains, missing data, constrast variation, warping effect, and noise due to lighting variation during the process of scanning or capturing. A common problem encountered when scanning documents is distortions which could occur in documents because of paper quality, or quality of scanners used during the scanning process. For instance, the distortions in a scanned document could come from the page rule line which can be a source of noise and can affect text objects. The marginal noise often occurs in a large dark region around the document, which could be textual or non-textual. For captured documents, some regions of background appear in the documents, and these undesired regions of documents are referred to as border noise[169].

There are two types of border noises including textual and non-textual noise. For example, when the regions of textual noise are fed to a character recognition engine, some extra characters will be obtained in the the ouput of this system along with the actual content of the document. Meanwhile, the non-textual noise would make further processing of document, such as text line extraction or removal of warp, a difficult task. These degradations would reduce the accuracy of tasks of document analysis and recognition, so transferring the documents into binarized forms or removing noise, and storing them in a binarized format is a pervasive solution to deal with these issues. As a result, the genuine documents which exist in the binarized format for meeting the requirement of various purposes are inevitable in the real world, and this is why we have decided to develop a watermarking scheme for securing binary documents.

As presented in Chapter 2, most of the previous watermarking schemes have been put forward for natural and document images, which are in color or grayscale format whose pixel intensity has a wide range of value. There are several techniques utilized in grayscale and color images to extract image features which are used to construct the watermarking system. These techniques consist of keypoint detector, edge detector, local binary pattern, contour detection, etc. However, the schemes for grayscale and color images can not be directly applied for binary images because the pixel values of binary images are represented by only back and white value. Besides, the approaches presented in[1;48–53;71] can be applied for both grayscale and binary documents. However, these methods are only dedicated to text

documents or a specific text language like Indian or Chinese. They might not perform well when applying for documents with mixed content because these approaches do not provide strategies on how identify and separate text and non-text elements.

Compared to color and grayscale images, there are few data hiding and watermarking methods[14–16;91;96;97;100–102] designed for binary images. Majority of existing techniques for binary images partition images into non-overlapping blocks with a predefined window of size $m \times n$ in order to retrieve image content where the secret information is hidden into. In the the work[15;97] the authors utilize contour tracing technique to detect object contours that are flipped to hide a secret information. Furthermore, the approaches presented in[14–16] satisfy the properties of imperceptibility and capacity, but they do not meet the requirements of robustness and security. Meanwhile, apart from satisfying the essential properties of imperceptility and capacity, other schemes[91;96;98;100–102] meet one of the other critical properties either robustness or security, but not both of them.

To address the shortcomings of existing approaches, we apply pattern recognition technique, which is very less exploited in designing data hiding and watermarking system for binary images, to develop a new watermarking framework for security issue of binary documents. This technique is FCN, which has been proposed in our previous works presented in Chapter 4 for grayscale documents. Unlike CNN-based method[63] wherein the authors make use of weight parameters of deep learning framework for watermarking process on the fixed size images, the FCN-based method can be applied on the arbitrary sized documents. With this approach, we train the network in such a way the trained network can be used for producing a salient map describing watermarking regions, and the process of watermark hiding and detection is separately designed during the training of the FCN. As discussed earlier, our proposed network provides a flexible way to identify a certain content region of document. With a given document as in Figure 4.32(a), the network can accurately identify the text region and picture region corresponding to Figure 4.32(b) and Figure 4.32(c).

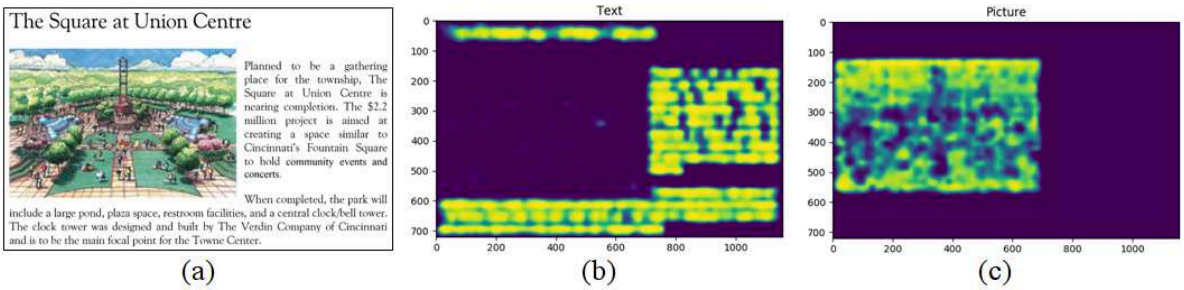


Figure 4.32: *The content regions of document: the text regions (b) and picture region (c).*

A binary document image of size $M \times N$ is defined by a matrix $I(x, y)$.

$$I(x,y) = \left\{ \begin{array}{ll} 1, & \text{for points on the object;} \\ 0, & \text{for background points.} \end{array} \right\}$$

where $0 \leq x \leq M-1$, $0 \leq y \leq N-1$, 1 and 0 demonstrate white (object or foreground)

and black (background) pixels.

4.6.1 Detection of hiding regions using FCN

Due to the need of work in different sectors, the documents are often presented under various dimension. After hiding secret information into document, furthermore, we expect to obtain a watermarked document retaining a dimension similar to the original document. Besides, from our observation, the extracted regions describing various types of document content such as running text, headline, picture, table, etc. using FCN[66] are relatively more stable against distortions when using FCN than other approaches such as active contour, non-subsampled contourlet transform[65], maximally stable extremal regions[115], speeded up robust features[116], etc. This is why we have adopted this kind of network in our work.

The FCN network for detecting watermarking regions is depicted in Figure 4.33 in which the blocks (B1-B7) contain convolutional layers for feature extraction, and the process of feature extraction is known as downsampling. The principle of downsampling and upsampling operations for FCN follows the work presented in[66]. The architecture of our network is partly based on the VGG-16 network[138], which is converted to fully convolutional networks. We replace three fully connected layers of the network[138] with convolutional layer blocks (B6, B7) for retaining spatial information. This replacement makes VGG-16 network becoming a fully convolutional networks. With regard to the FCN, the feature maps obtained at the output layer have the same dimension than the input document.

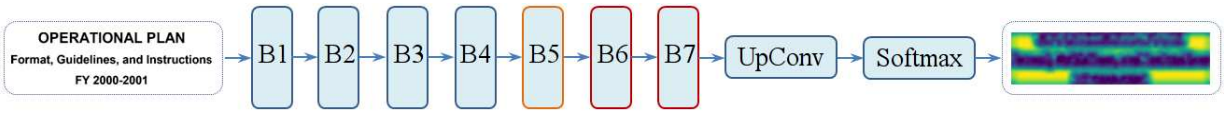


Figure 4.33: *The architecture of FCN for detecting watermarking regions. This kind of network takes a binary document as an input and generates the salient maps with the same dimension of the input document as an output.*

However, the feature maps generated in the phase of convolutional operations or downsampling are reduced in dimension, so they need to be reconstructed by performing upsampling. The process of recovering the spatial dimension of the original document is represented by "UpConv" block, which is known as upsampling operations, in Figure 4.33 wherein a few transposed convolution layers are used for maintaining the spatial information. Besides, we apply Softmax function on the output layer to transform the result of the network into a two-class problem for representing the probability of document's watermarking regions. As a result, the feature maps or salient maps with the same dimension of the input document are produced at the output layer.

The convolutional operations of each block of the network are briefly described in Table 4.3 in which each block consists of one or more convolutional layers, a rectified linear unit (ReLU), a max pooling layer, and with or without a dropout layer. We use notation of (kernel, stride) and (kernel) to define the convolutional layers as follows.

Table 4.3: *The description of convolutional operations of FCN*

Blocks	Convolutional operations in each block
B1, B2	2 conv layers (3×3, 1×1), ReLU, max pooling (2×2)
B3, B4	3 conv layers (3×3, 1×1), ReLU, max pooling (2×2)
B5	3 conv layers (3×3, 1×1), ReLU, max pooling (2×2)
B6	1 conv layer (7×7, 1×1), ReLU, dropout
B7	2 conv layers (1×1, 1×1), ReLU, dropout

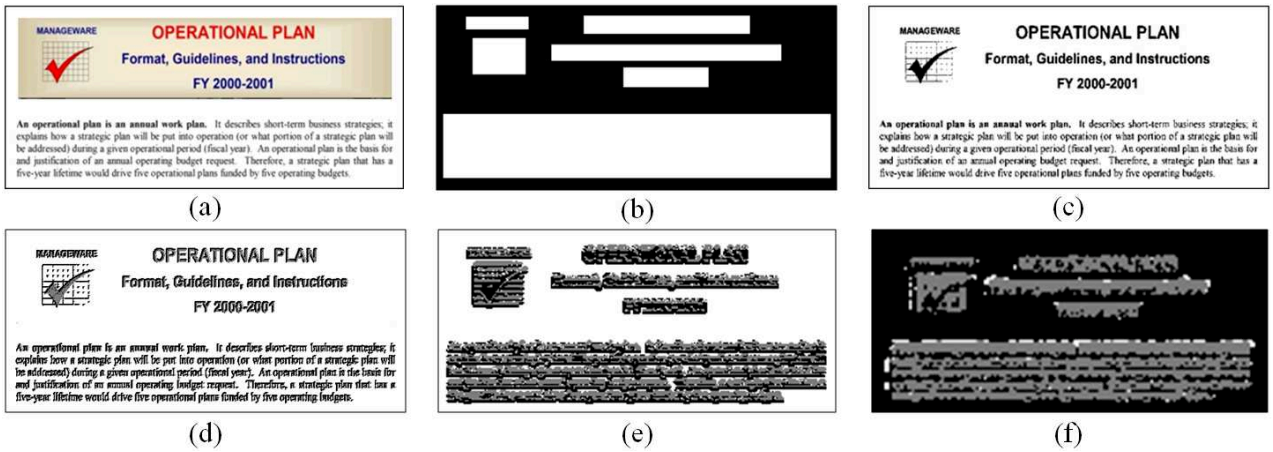


Figure 4.34: *The illustration of generated feature maps: (a) and (b) are document and ground truth used for training network; (c) is a binary document; (d)-(f) are features obtained at block B1, B3 and B6.*

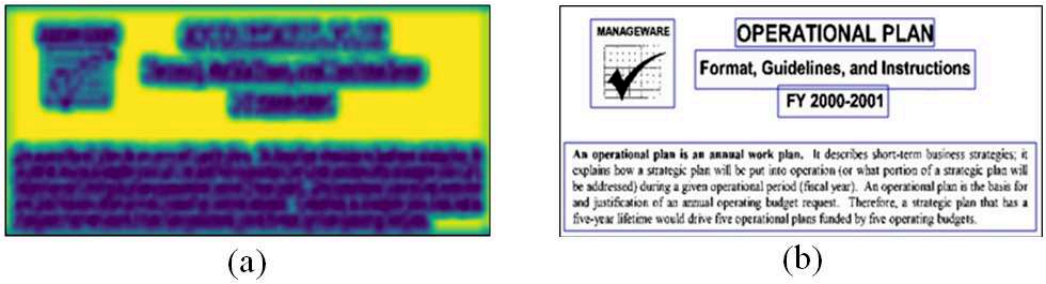


Figure 4.35: *The illustration of generated salient maps (a) and watermarking regions which are surrounded by blue rectangles (b).*

The documents along with their ground truth as illustrated in Figure 4.34(a)-(b) are used to train the network. The feature maps generated from first blocks (B1 - B3) of convolutional layers represent the overall shape of document content while the region-specific information at last blocks (B4 - B7). The salient maps describing the content regions of document are obtained at the output layer as depicted in Figure 4.35(a) wherein the yellow color depicts the

background of document, and the dark blue color depicts the foreground of document. The salient maps are computed based on the score of the feature maps acquired from our trained network. Lastly, with the obtained salient maps, the bounding boxes as blue rectangles in Figure 4.35(b) are easily determined by making use of connected components. The advantage of our network is that it provides a flexible way to eliminate unwanted content regions, which are not expected to hide data, in the documents if necessary.

4.6.2 Construction of hiding patterns

For binary document images, the connectivity of neighboring pixels and distance among pixel values play a crucial role for preserving the quality of documents because changing a pixel value from black to white and vice versa could easily cause visual perception. Thus, in the previous pattern-based approaches like[14;15;98], such features as connectivity, uneven embeddability and pattern score are considering in designing their data hiding schemes. To achieve imperceptibility, the possibility of changing pixel values can be only carried out on the boundary of document objects. Unlike the existing approaches, our proposed hiding patterns are constructed in a combination among the connectivity of neighboring pixels and the characteristics of document content like corner and edge features. After changing pixel values for carrying secret information, these features need to be maintained. In order to obtain invariant feature and maximize the capacity, a pattern of 3×3 as illustrated in Figure 4.36 is selected to detect the corner and edge features of the objects where the secret information is hidden into.

p_1	p_2	p_3
p_8	p_c	p_4
p_7	p_6	p_5

Figure 4.36: *The hiding pattern of* 3×3 *used to detect the corner and edge features of the objects.*

A 3×3 hiding pattern of neighboring pixels representing the corner and edge features of the objects can be described as follows:

Hiding patterns for corner feature detection: There are two types of hiding patterns describing the corner features. These patterns have to satisfy the following requirements: The number of all neighboring black pixels of p_c (center of the pattern) is equal to 4 or 5; and the number of such transition as $0-1$ or $1-0$ between two consecutive neighboring pixels of p_c is equal to 2. If the transition is greater than two, the connectivity of pixels within a pattern will be broken. This could easily lead to visual perception and substantially affect the quality of binary document. The black neighboring pixels of the center pixel p_c for these two types of hiding pattern are illustrated by:

$$CP_1 = \{p_i \wedge p_{i+1} \wedge p_{i+2} \wedge p_{i+3}\} \tag{4.20}$$

where $i = \{1, \ldots, 8\}$, $p_9 = p_1$, $p_{10} = p_2$ and $p_{11} = p_3$.

$$CP_2 = \{p_i \wedge p_{i+1} \wedge p_{i+2} \wedge p_{i+3} \wedge p_{i+4}\} \tag{4.21}$$

where $i = \{1, 3, 5, 7\}$, $p_9 = p_1$, $p_{10} = p_2$ and $p_{11} = p_3$.

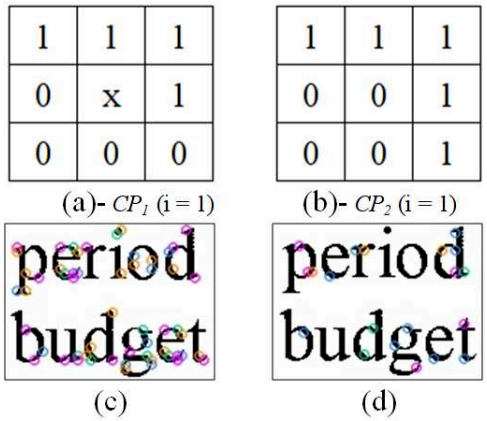


Figure 4.37: *The illustration of corner feature detection: (a) and (b) depicts an instance of hiding patterns CP_1 and CP_2 with $i = 1$. The corner positions corresponding to all instances of these patterns are illustrated in (c) and (d).*

Concerning CP_2, to maintain the corner features, the center pixel p_c has to be equal to 0. If $p_c = 1$, this pattern will become a pattern which describes an edge feature EP_2 in case of $i = 1, 3, 5, 7$. Meanwhile, p_c is possibly equal to 0 or 1 for the case of CP_1.

An illustration of hiding patterns for corner features and the outcome of corner feature detection are depicted in Figure 4.37 in which (a) and (b) are an instance of hiding patterns demonstrating the corner patterns CP_1 and CP_2 in case of $i = 1$ whereas (c) and (d) depict the result of corner feature detection. The small color circles depict corner positions obtained by detecting all instances of the hiding patterns CP_1 and CP_2. The sign "x" in an instance of the corner pattern CP_1 indicates that we don't care the pixel value at this position. It means that this value can be either equal to 0 or 1.

Hiding patterns for edge feature detection: Similar to the corner features, there are two types of hiding pattern describing the edge features. These patterns are also required to meet the following conditions: The number of all neighboring black pixels of p_c is equal to 3 or 6; and the number of such transition as $0-1$ or $1-0$ between two successive neighboring pixels of p_c is equal to 2. The black neighboring pixels of the center pixel p_c for the edge features are given by:

$$EP_1 = \{p_i \wedge p_{i+1} \wedge p_{i+2}\} \tag{4.22}$$

where $i = \{1, \ldots, 8\}$, $p_9 = p_1$ and $p_{10} = p_2$.

$$EP_2 = \{p_i \wedge p_{i+1} \wedge p_{i+2} \wedge p_{i+3} \wedge p_{i+4} \wedge p_c\} \tag{4.23}$$

where $i = \{1, \ldots, 8\}$, $p_9 = p_1$, $p_{10} = p_2$, $p_{11} = p_3$ and $p_{12} = p_4$.

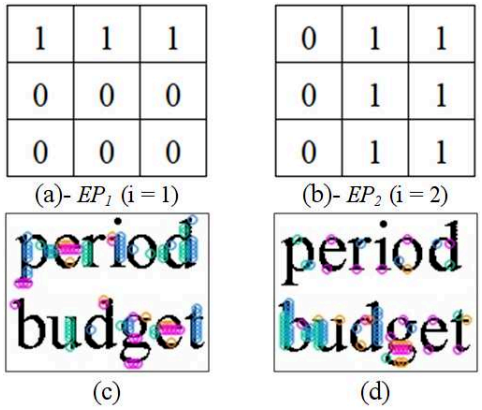


Figure 4.38: *An example of edge feature detection: (a) and (b) depict an instance of hiding patterns EP_1 and EP_2 with $i = 1$ and $i = 2$ respectively. The color circles in (c) and (d) represent the edge positions corresponding to all instances of these patterns.*

To keep the edge features, the center pixel p_c of EP_1 could be assigned by either 0 or 1 whereas it must be assigned by 1 for EP_2. For EP_1, the hiding pattern will become a pattern describing a convex point if $p_c = 1$. Meanwhile, regarding EP_2, if p_c is equal to 0, the hiding pattern will become a pattern describing a concave point for $i = 2, 4, 6, 8$, and describing the corner feature CP_2 when $i = 1, 3, 5, 7$. Figure 4.38 depicts an instance of hiding patterns for edge features and the illustration of edge positions corresponding to all instances of patterns EP_1 and EP_2: (a) and (b) are an example of the patterns EP_1 and EP_2 when $i = 1$ and $i = 2$ respectively. The small color circles in (c) and (d) demonstrate the edge positions acquired by detecting all instances of the hiding patterns EP_1 and EP_2.

4.6.3 Data encoding and decoding for enhancing security feature

The secret information used in this work is a text message. Thus, it needs to be modulated into a sequence of bits before the encoding process in which every character is converted into eight bits. In fact, majority of data hiding and watermarking algorithms are likely public, so the malicious users can obtain easily the secret information hidden inside the documents. For this reason, to enhance the security of the watermarking scheme, we encode the secret information prior to hiding it into document and decode the extracted secret data by using pseudo-random numbers. This technique can be used to generate a series of positive numbers based on a random seed. Different from the previous methods in which the authors use this technique to shuffle positions of flippable pixels as in[96] or to permute block patterns as in[98], here we use these generated positive numbers to encode and decode the secret information.

In order to choose a good random seed, we apply an algorithm of symmetric key to generate a secret key. This key is converted into a number, and it is secretly shared with authorized recipients. We consider the converted number of the secret key as a seed. This seed is used to feed the pseudo-random numbers generator for generating a series of positive numbers, which are used for encoding and decoding the secret information. With the same generator of pseudo-random numbers and the same seed, we obtain the exact sequence of positive numbers. The number of positive numbers generated from pseudo-random numbers generator is set to be equal to the number of characters that the secret information contains.

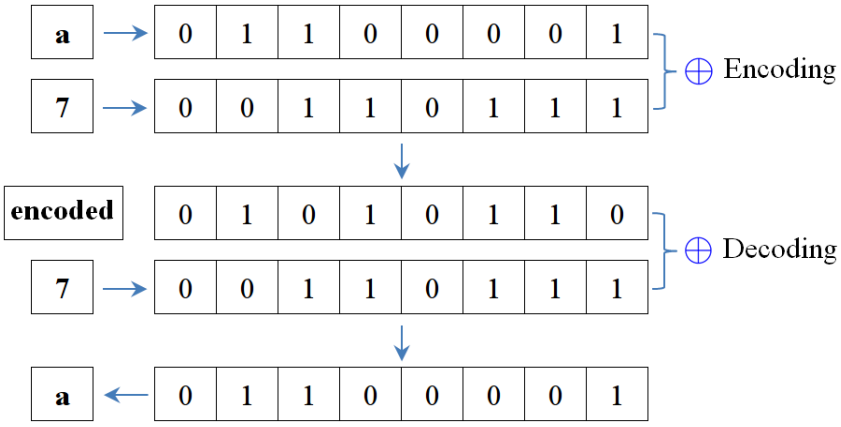


Figure 4.39: *The illustration of encoding and decoding process in a combination of secret information and pseudo-random numbers by using "X-OR" operator. The random seed is considered as a "private key" in this context.*

The main idea of encoding the secret information is to combine each character of the secret information (text message) with a positive number of pseudo-random numbers to generate the respective encoded data. To do so, the sequence of positive numbers is also converted into a string of bits wherein each positive number is converted into eight bits. We apply "X-OR" operator between these two bytes of binary data generated from the secret information and pseudo-random numbers for combining them. As a result, we obtain the encoded data which is hidden into the document. With regard to data decoding, the process is carried out by applying "X-OR" operator on the extracted information and pseudo-random numbers with the same random seed used during the encoding phase.

Figure 4.39 shows an example of data encoding and decoding process in which "**a**" is a character of secret information, "**7**" is a positive number of a series of pseudo-ramdom numbers generated in the encoding and decoding phase. For the encoding phase, the "X-OR" operator is applied on two sequences of binary digits corresponding to "**a**" and "**7**". The result of this step produces an encoded data known as "**encoded**". For the decoding phase, the same random seed is used to regenerate the positive number as "**7**". The "X-OR" operator is applied on the two sequences of binary digits corresponding to "**encoded**" and "**7**", and then the hidden data is recovered.

4.6.4 Data hiding process

In this section, we present two methods for hiding secret information into binary documents including: (*i*) Hiding data based on the characteristics of document content such as the corner and edge features; (*ii*) Enhancing robustness against distortions by adjusting the proportion between the number of edge features and the number of corner features inside each subregion of the watermarking regions. The main steps of watermark hiding process are illustrated in Figure 4.40.

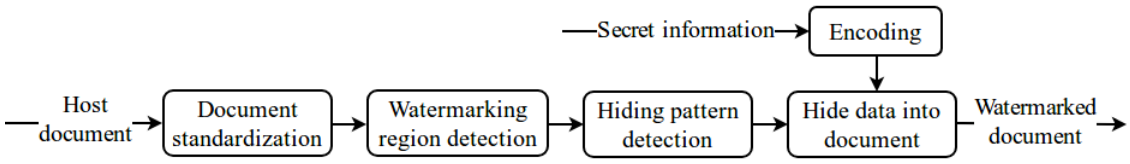


Figure 4.40: *The main steps of hiding secret information.*

Watermark hiding scheme 1 (WM1)

This scheme is implemented by flipping the center pixel of hiding patterns describing the corner and edge features of the objects, and it satisfies the properties of imperceptibility, capacity and security. The process of hiding secret information basically comprises of the following main steps:

- Step 1 - Encoding the secret information: as detailed in Section 4.6.3.
- Step 2 - Standardization of input documents: Determining parameters for document standardization is conducted based on the minimum box surrounding the entire document, where the points located on the stroke part of document's objects are used to construst the minimum rectangle. This task is conducted similar to the method presented in Section 3.3.3.
- Step 3 - Detection of the watermarking regions: as described in Section 4.6.1. For documents with complicated structure, the extracted watermarking regions could be overlapped or nested to another, so these regions need to be eliminated similarly to the method presented in Section 3.3.4.
- Step 4 - Hiding the watermark bits into the document: We use the hiding patterns for corner and edge feature detection as presented in Section 4.6.2 to find potential positions inside the watermarking regions for hiding secret information bits. The center pixel p_c of an appropriate pattern is changed (replacing 0 with 1 and vice versa) to hold a secret bit wm_i. For the hiding pattern CP_2, for instance, if the center pixel p_c is changed to 1, this pattern will become a pattern describing the edge features as in EP_2. Meanwhile, with regard to hiding patterns EP_1, if $p_c = 1$, this pattern will become a pattern describing a convex point, in case of $i = 1, 3, 5, 7$. In case of hiding

pattern EP_2, the pattern will describe a concave point if the center pixel p_c is adjusted to 0, in case of $i = 2, 4, 6, 8$. Changing the center pixel p_c is described by:

$$p_c = \begin{cases} 0, & \text{if} \quad wm_i = 1 \\ 1(255), & \text{if} \quad wm_i = 0 \end{cases} \tag{4.24}$$

Watermark hiding scheme 2 (WM2)

Despite the method presented in Section 4.6.4 is capable to resist to common distortions including JPEG compression and geometric distortion, it gives low perfomance when the watermarked documents go through print-and-scan (PS) process. Obviously, the pixel values located on the stroke part of binary document objects are changed a lot when the documents are subjected to PS distortion, and these changes are illustrated in Figure 4.41. Changing pixel values due to PS distortion leads to lose the integrity of corner and edge features, and this mainly causes failure in extracting the hidden information. This is why we propose another approach to improve the robustness such that the scheme is able to withstand such a distortion.

(a) (b)

Figure 4.41: *A sample of original text of a binary document (a), and the printed and scanned text at resolution of 600 dpi (b).*

The idea of this approach is to adjust the proportion between the number of edge features and the number of corner features within each subregion of the watermarking regions for carrying one secret information bit, instead of changing only the center pixel p_c of hiding patterns as in WM1. The subregion is formed by concatenating the adjacent bounding boxes of the objects, which are detected by optical character recognition (OCR), with an appropriate distance such that its content is large enough to retain the proportion of edge and corner features under the environment of distortions. The subregions within watermarking regions are depicted as blue rectangles in Figure 4.42.

Unlike the approach presented in[102] in which the authors adjust back pixels within the appropriate thumbnail images in order to balance the difference between the number of black pixels in each thumbnail and the mean value of all thumbnail images, our approach is to change corner features in each subregion into the edge features for hiding bit 0, or to change edge features into the corner features for hiding bit 1. In other words, after hiding bit 0, a subregion of watermarking regions contains more edge features than corner features. A subregion of watermarking regions will contain more corner features than edge features if we want to hide bit 1. To do so, the hiding patterns of 3×3 are reused for detecting the

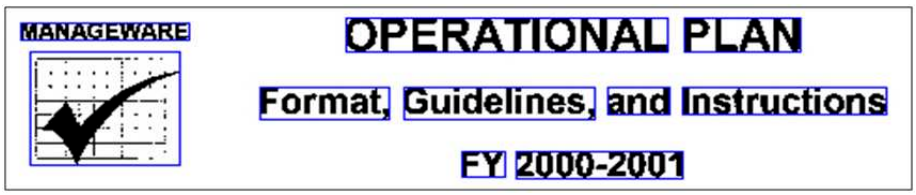


Figure 4.42: *The illustration of subregions (blue rectangles) within the watermarking regions.*

edge and corner features as presented in WM1. Once these features are located, there are two types of operations for feature adjustment such as either deleting black pixels of the object's stroke (decreasing the number of black pixels and increasing the number of white ones) or adding black pixels of the object's stroke (increasing the number of black pixels and decreasing the number of white ones). With this approach, it enables to hide one secret bit into a subregion instead of a corner or an edge feature as WM1. Thus, the amount of secret information that a document can carry to be less than as compared to WM1.

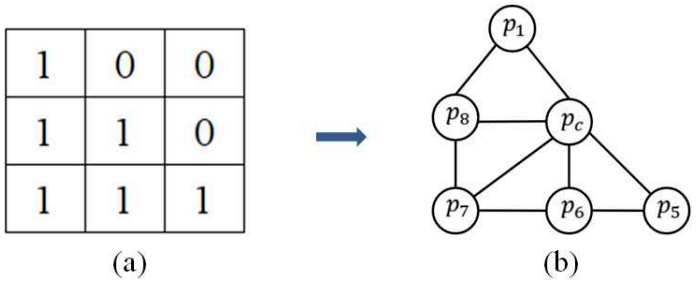


Figure 4.43: *A representation of* 3×3 *neighboring pixels around the object's stroke (a), and the corresponding undirected graph is constructed (b).*

To preserve the document quality with regard to human perception, the operation of deleting or adding black pixels of an object should keep its corner or edge features, and does not break the connectivity of neighboring pixels. Figure 4.43(a) demonstrates a 3×3 hiding pattern representing the neighboring pixels around the object's stroke, and the graph representation of the neighboring conectivity for black pixels is shown in Figure 4.43(b). We consider the connectivity of black pixels as an undirected graph where arcs depict the adjacency of 1's pixels. An arc will be legal if it directly connects from one vertex to another (e.g. $p_1 - p_8$) and without going through a third vertex (e.g. $p_1 - p_7$, p_1 connects to p_7 through p_8 or p_c). The possible legal arcs between vetices representing the neighboring pixels of p_c are depicted in Table 4.4. The vertex p_c always makes a legal arc to all its neighbors.

Deleting or adding a black pixel without causing the disconnectivity (separated subgraph) is possible if each vertex in the graph has at least one arc connected to it. In order to examine whether it is eligible to remove, or to add a vertex to the graph, we construct an adjacency matrix for the connectivity examination. For instance, if we delete the center pixel p_c in Figure 4.43(b), it will generate a new graph with the adjacency matrix depicted in Table 4.5. It is easy to see that the deletion of a pixel will not produce a disconnected graph if

Table 4.4: *The illustration of possible arcs for neighboring pixels in a* 3×3 *hiding pattern*

Vertex	Connect to possible vertices	Vertex	Connect to possible vertices
1	2, 8	5	4, 6
2	1, 3, 4, 8	6	4, 5, 7, 8
3	2, 4	7	6, 8
4	2, 3, 5, 6	8	1, 2, 6, 7

every row of the adjacency matrix contains at least one value of 1. If p_c and p_8 are deleted simultaneously, this will not be allowed because there is a complete line with values of 0. This means that a disconnected subgraph is generated. In case of pixel insertion, the connectivity examination is similar to the deletion.

Table 4.5: *An adjacency matrix for the undirected graph depicted in Figure 4.43(b) as deleting the center pixel* p_c

Graph vertex	Connectivity of vertices					Sum
	p_1	p_5	p_6	p_7	p_8	
p_1	0	0	0	0	1	1
p_5	0	0	1	0	0	1
p_6	0	1	0	1	0	2
p_7	0	0	1	0	1	2
p_8	1	0	0	1	0	2

For both of WM1 and WM2, we eliminate the hiding patterns with full of white or black pixels because these patterns represent uniform areas, and deleting or adding pixel values on the uniform areas of binary documents easily affects to the visual perception. In order to change the edge features into corner features and vice versa, the operation of adjusting pixel values is performed by checking the connectivity using the adjacency matrix as presented above. After adjustment of black pixels, all pixel values inside the 3×3 hiding pattern have to satisfy the neighboring connectivity and its corner or edge features. We refer to maintaining the neighboring connectivity, and the corner or edge features as "connectivity-features". Specifically, we adjust black and white pixels of each subregion in such a way the number of corner features is lower than the number of edge features by a threshold δ for hiding bit 0, and the number of corner features is greater than the number of edge features by a threshold δ for hiding bit 1.

To hide a secret bit wm_i into a subregion O_i, let $P_i = \{pt_1, pt_2, ..., pt_n\}$ be a set of edge and corner features situated around the stroke of O_i, s_1 be the sum of all edge features in P_i, s_2 be the sum of all corner features in P_i, d be the absolute difference between s_1 and s_2, δ be a threshold to separate s_1 and s_2. The main steps for hiding secret information is similar to the watermark hiding scheme 1. It just differs in step 4, which adjusts the proportion between the number of edge and the number of corner features. The algorithm for changing these features is described as follows:

- if the i^{th} watermark bit $wm_i = 0$:
  - $s_1 > s_2$ and $d \geq \delta$: the proportion between the number of edge features and the number of corner features in P_i keeps unchanged.
  - Otherwise: changing the pixel values in $pt_k \in P_i$ in a way to decrease the number of corner features and increase the number of edge features in P_i such that $s_1 > s_2$, $d \geq \delta$ and keeping "connectivity-features".
- if the i^{th} watermark bit $wm_i = 1$:
  - $s_1 < s_2$ and $d \geq \delta$: the proportion between the number of edge features and the number of corner features in P_i keeps unchanged.
  - Otherwise: changing the pixel values in $pt_k \in P_i$ in a way to increase the number of corner features and decrease the number of edge features in P_i such that $s_1 < s_2$, $d \geq \delta$ and keeping "connectivity-features".

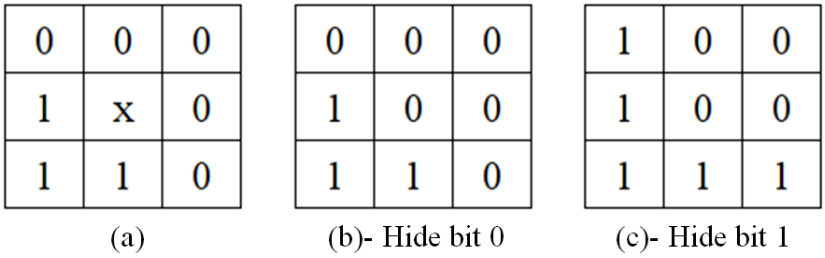


Figure 4.44: *(a) A hiding pattern of the object stroke located within a subregion. (b) and (c) are the result of pixel adjustment in order to obtain the edge and corner feature, and keep the neighboring connectivity for hiding secret bit* 0 *and* 1.

Figure 4.44(a) illustrates an example of a hiding pattern within a subregion where we need to change its pixel values into an edge feature or corner feature for watermark hiding process, the sign of "x" indicates that the pixel value at this position is either equal to 0 or 1. If we want to hide a secret bit 0, this pattern needs to be changed into the edge feature as in Figure 4.44(b) by deleting the pixel value at the center position p_c. Otherwise, if the secret bit is 1, the pattern has to be changed into the corner feature by deleting pixel value at the position p_c and inserting the pixel values at the positions corresponding to p_1 and p_5 as depicted in Figure 4.44(c). These modifications are carried out in compliance with the requirement of "connectivity-features", so distortion caused by such alteration is less perceptible by the human visual system.

4.6.5 Data detection process

The main steps of secret information detection are carried out in similar fashion as the watermark hiding process. It just differs by extracting the hidden information and decoding

the extracted information to obtain a meaningful information. The hidden bit wm_i from the watermark hiding scheme 1 is extracted by:

$$\text{WM1:} \quad wm_i = \begin{cases} 0, & \text{if} \quad p_c = 255 \\ 1, & \text{if} \quad p_c = 0 \end{cases} \tag{4.25}$$

In order to extract the hidden information from the watermark hiding scheme 2, let s_1 be the number of edge features, and s_2 be the number of corner features in each subregion P_i. The hidden information bit wm_i is then extracted by:

$$\text{WM2:} \quad wm_i = \begin{cases} 0, & \text{if} \quad s_1 \geq s_2 \\ 1, & \text{otherwise} \end{cases} \tag{4.26}$$

The extracted information bits acquired from WM1 and WM2 correspond to the encoded form, and therefore these extracted bits need to be decoded for information recovery. To do so, the pseudo-random numbers generator with the same seed as in the encoding phase is reused to generate a series of positive numbers. These numbers are converted into a sequence of bits, and the process of hidden information decoding is then conducted by applying "exclusive-OR" operator on a sequence of bits of extracted information and a sequence of bits of random numbers as presented in Section 4.6.3.

The accurate proportion of secret information extraction is measured by comparing the original secret bits with the extracted bits after decoding, and it is defined by Equation 3.13.

In brief, we have proposed a new watermarking scheme for securing binary documents. In this work, we utilize the FCN network to detect the regions of document content, which are known as watermarking regions. Besides, we have proposed the hiding patterns describing the corner and edge features. The watermarking algorithm is designed by changing the center position of these patterns, or the proportion between the number of edge and corner features. Changing the pixel values for hiding secret data into the document has been conducted in order to keep the condition of "connectivity-features". The security feature is also integrated into the scheme. The scheme is capable of resisting to common distortions, and print-and-scan process with high resolution. However, it still has not reached high performance when the watermarked documents are subject to complicated distortions.

4.7 Summary

In this chapter, we have theoretically made two major contributions in designing watermarking schemes for grayscale typewritten, handwritten documents, and binary documents, consisting of:

- Document feature extraction by using deep learning technique, which is used to detect

the hiding regions, reconstruct a good quality document from a distorted one, and generate variations of characters and symbols.

- Development of watermarking algorithms based on pixel intensity, and shape of character and symbols.

For the former, the FCN is adjusted to detect the regions of the document content that correspond to various types of documents, and this network provides a flexible way to eliminate unwanted content regions where the secret information can not be hidden. Besides, the FCN is effectively applied to detect the regions of a watermarked document where the appropriate characters and symbols are replaced by their variations during the process of information hiding. Furthermore, we have utilized GAN network to generate a good quality document which is used to improve the detection of document content, and develop watermarking algorithm.

For the later, several watermarking algorithms, which are based on the pixel level and the shape of characters, have been proposed. The pixel intensity-based algorithms are developed by adjusting: (1) the pixel values within each watermarking pattern; (2) the pixel values within each separated handwriting element; (3) the difference of pixel values between the watermarked and reference document; (4) the center position of the hiding patterns describing the edge and corner features; and (5) the ratio between the number of corner and edge features.

Meanwhile, the shape of characters-based algorithm is developed relied upon the variations of document characters. The performance of these approaches will be presented in detail in Chapter 5.

Chapter 5

Experiments and evaluation of scheme performance

In this chapter, we evaluate the performances of the proposed approaches. The experiments are carried out on public datasets. Some comparisons with state-of-the-art methods are also provided in order to show the relevance of the proposed strategies. We assess our schemes based on the following properties: imperceptibility, capacity and robustness against simulated and practical distortions. Besides, we also compare the performance among our proposed approaches to see the improvement of our schemes in detecting the hidden information in the environment of practical distortions. Specifically, we present the public datasets that we use to evaluate our works. The performance evaluations of the proposed data hiding systems based on the conventional approaches (Chapter 3) and the deep learning approaches (Chapter 4) are detailed in the following sections of this chapter.

5.1 Dataset and measurement of imperceptibility

In order to evaluate our methods and train the deep networks, we use the following datasets:

- Tobacco[119] consists of 1290 document images, which are collected and scanned using a wide variety of equipment. The resolutions of documents vary significantly from 150 to 300 dpi, and the dimensions of images range from 1200 by 1600 to 2500 by 3200 pixels.
- L3iDocCopies[120] contains 990 documents which are printed and scanned at various resolutions of 300 dpi and 600 dpi from different machines. This dataset is dedicated to document segmentation analysis, and print and scan noise analysis.
- PRImA[170] for ICDAR page segmentation competitions, which is based on comprehensive and detailed representation of both simple and complex layouts. It consists of 54 sample pages of magazines together with ground truth metadata.

- DSSE-200[121] contains 200 pages from magazines and academic papers. This dataset provides both appearance-based and semantics-based labels.
- CVL-database[149] consists of 7 different handwritten texts (1 German and 6 English texts). In total 310 writers participated in the dataset. 27 of which wrote 7 texts and 283 writers had to write 5 texts. For each text, a rgb color image (300 dpi) comprising the handwritten text and the printed text sample is available as well as a cropped version (only handwritten).
- Standard grayscale test images [1] which are widely used to test image processing and image compression algorithms.

To measure the quality of stego and watermarked documents, the following measures are applied for evaluating the performance of our method:

- Peak signal to noise ratio (PSNR) that uses the amount of pixel errors between the reference and distorted images to determine the level of distortion. This method does not consider the characteristics of the human visual system (HVS).
- Structural similarity index measurement (SSIM)[171] that estimates local brightness, contrast, structure of reference and image distortion, and then averages all local assessments to obtain the overall assessment. The human eyes can easily perceive the local information differences within an area, so this technique adopts a patch-based approach instead of individual pixel differences.
- Distance reciprocal distortion measure (DRDM)[45] corresponds to the change of smoothness and connectivity that significantly affects the human visual perception. In addition, this method correlates well with subjective assessment by human eyes.

The maximum number for PSNR and SSIM is ∞ and 1, respectively. Meanwhile, the minimum value for both is 0. The higher the value of PSNR or SSIM is, the better the image quality is. The small value of DRDM indicates less distortion. They are defined as follows.

$$PSNR = 10 \times log_{10} \frac{255^2}{MSE} \tag{5.1}$$

where

$$MSE = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (I_c(i,j) - I_s(i,j))^2 \tag{5.2}$$

where M and N represent the length and width of a document.

$$SSIM(I_c, I_s) = \frac{2u_{I_c}u_{I_s} + c_1}{u_{I_c}^2 + u_{I_s}^2 + c_1} \times \frac{2\sigma_{I_c}\sigma_{I_s} + c_2}{\sigma_{I_c}^2 + \sigma_{I_s}^2 + c_2} \times \frac{\sigma_{I_c,I_s} + c_3}{\sigma_{I_c}\sigma_{I_s} + c_3} \tag{5.3}$$

[1] http://decsai.ugr.es/cvg/CG/base.htm

where u_{I_c}, u_{I_s}, σ_{I_c} and σ_{I_s} represent the mean and variance of pixel values of the undistorted document (I_c) and distorted document (I_s), and σ_{I_c,I_s} is the covariance between I_c and I_s. Meanwhile c_1, c_2 and c_3 are constants to avoid instability when $u_{I_c}^2 + u_{I_s}^2$, $\sigma_{I_c}^2 + \sigma_{I_s}^2$ and $\sigma_{I_c}\sigma_{I_s}$ are very close to zero.

DRDM measures the difference of distortion between a distorted image $g(x, y)$ and the original image $f(x, y)$ using a weight matrix wherein each weight corresponds to the reciprocal of the distance measured from the center pixel. A weight matrix W_m is of size $m \times m$, $m = 2n + 1, n = 1, 2, 3, ...$ $W_m(i, j)$ is then defined by:

$$W_m(i,j) = \begin{cases} 0, & \text{for } i = i_C \text{ and } j = j_C \\ \frac{1}{\sqrt{(i-i_C)^2+(j-j_C)^2}}, & \text{otherwise} \end{cases} \tag{5.4}$$

where $1 \leq i, j \leq m$, (i_C, j_C) is the center location of this matrix, and $i_C = j_C = (m + 1)/2$.

This matrix is normalized to form the normalized weight matrix W'.

$$W'_m(i,j) = \frac{W_m(i,j)}{\sum_{i=1}^{m}\sum_{j=1}^{m} W_m(i,j)} \tag{5.5}$$

Assume that there are P flipped pixel in $g(x, y)$, each pixel will have a distortion $d_k, k = 1, 2, 3, ..., P$. For the k^{th} flipped pixel (changing black pixel to white pixel and vice versa) at $(x, y)_k$ in the distorted image, the resulted distortion is calculated from a $m \times m$ block B_k in $f(x, y)$ that is centered at $(x, y)_k$. The distortion measure d_k for this flipped pixel $g[(x, y)_k]$ is defined by:

$$d_k = \sum_{i,j} [D_k(i,j) \times W'_m(i,j)] \tag{5.6}$$

where the element of the difference matrix D_k is given by:

$$D_k(i,j) = \left| B_k(i,j) - g[(x,y)_k] \right| \tag{5.7}$$

For possibility of flipped pixel near the image corner or border, where $m \times m$ neighborhood may not exist, so it is possible to expand the rest of $m \times m$ neighborhood with the same value as $g[(x, y)_k]$. The distortion in $g(x, y)$ is then calculated by:

$$d = \frac{\sum_{k=1}^{P} d_k}{K} \tag{5.8}$$

where K is determined as the number non-uniform (not all white or black pixels) blocks of 8×8 in $f(x, y)$, P is the positions of flipped pixels.

5.2 Steganography scheme based on feature points

This section demonstrates the performance of the approach presented in Section 3.2. We have experimented this scheme on documents and images from three datasets including: Tobacco800 (Type1); L3iDocCopies[120] (Type2); and standard grayscale test images (Type3). The secret information used in this experiment is the message "stego-msg", and it is converted into a sequence of bits with a length of 72 bits (9 characters $\times$ 8 bits). The size of a hiding region B is assigned to the value $L = 27$. The constant c to estimate a dynamic threshold used for LTP is set to 0.06. These values are selected relied upon the good tradeoff between capacity and robustness. If L is too large, the capacity will be diminished. Otherwise, the robustness will be low. The sample documents are shown in Figure 5.1. The performance of our steganography scheme is depicted through the following factors.

(a)-Doc1 (b)-Doc2 (c)-Doc3 (d)-Doc4

Figure 5.1: *Sample documents: (a) and (d) from Tobacco, (b) and (c) from L3iDocCopies.*

Imperceptibility and capacity

The imperceptibility or quality of stego-documents is measured by the difference between the document before and after hiding a secret information. To evaluate the quality of a stego-document I_s, we have adopted peak signal to noise ratio (PSNR) and structural similarity (SSIM) in this experiment. The higher the PSNR and SSIM are, the more the similarity between the stego-document and the cover document I_c is. In fact, if PSNR is lower than 30 dB, the stego-document can be visually differentiated from the cover document. Regarding SSIM, it reflects perceptual distortions more precisely than PSNR. The value of SSIM is in the range of $[0, 1]$, and the closer value to 1 represents the better quality of stego-document with respect to the cover document.

Table 5.1: *The average of imperceptibility and accuracy of data detection*

Documents	Hiding patterns	PSNR (cover vs stego)	StegoR (%)
Type1 (20 documents)	23	37.88	100.00
Type2 (20 documents)	19	103.25	100.00
Type3 (12 images)	23	86.74	100.00

The average of imperceptibility (stego-document quality), hiding patterns and accuracy ratio of secret information detection is presented in Table 5.1 in which the hiding pattern column indicates the average of the number of hiding patterns used to hide 72 message bits. The value of "StegoR" indicates the percentage of the number of correctly extracted bits. From our experiment, we have seen that the hidden information has been successfully extracted from the stego-documents without distortion with high accuracy.

The hiding regions B where the information bits are hidden are presented as red rectangles in Figure 5.2.

Figure 5.2: *Feature points based-hiding regions (red rectangles) for data hiding*

Robustness assessment

The documents are often subjected to two kinds of distortion: JPEG compression, printing and scanning noises. Thus, these two distortions can prove the robustness of our steganography scheme.

Robustness against JPEG compression: To make the scheme more robust against this noise, we have implemented two types of error correction code. The first one is repetition code, and the other is a combination of repetition code and Hamming(7, 3) code in which the Hamming code enables to add more redundant bits for each group of information bits, and it corrects only one bit in each group of data bits. As a result, correcting the extracted information by repetition code gives high performance than others. With 72 bits of the sample secret information, we obtain 216 bits and 378 bits generated from repetition code, and the combination of repetition and Hamming(7, 3) code respectively. In general, with a lossy compression algorithm, it is quite difficult to manage the changing of pixel intensities. Thus, the possibility of encountering corrupted bits from the compressed document is very high. Figure 5.3 demonstrates the average accuracy ratio of information detection on the three datasets.

Robustness against printing and scanning (PS) distortions: Due to the life cycle of the document, the stego-documents may be printed out, and the printed versions can be scanned back. This process could be repeatedly performed several times before the scanned docu-

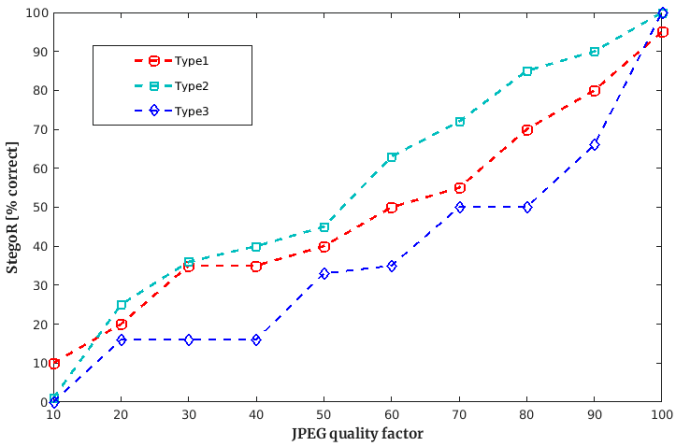


Figure 5.3: *Average accuracy ratio according to the JPEG quality factor.*

ments are delivered to the destination. According to the work presented in[172], the print-and-scan model for grayscale images consists of two processes: the printing process could cause some noises such as halftone, blur and geometric distortion whereas the scanning operation could introduce blur and geometric distortion. In this scheme, we only choose distortion of rotation to simulate the scanning operation. We assume that the documents are printed and scanned at the same resolution of 600 dpi. Thus, the size of documents still remain unchanged. This is why we ignore the scaling issue in this experiment. In addition, we apply Gaussian lowpass filter of size 3×3 to simulate blur noise which often occurred in the process of printing and scanning. The rotation of a small angle during the scanning process is mainly caused by human when placing the paper on the scanner screen. Prior to detecting the hidden information, the rotated document is corrected by using the Hough transfrom as discussed in Section 3.2.2.

Table 5.2: *Evaluation of the robustness to PS distortion (accuracy ratio in % when detecting information).*

Documents	**Gaussian lowpass filter**	**Rotation (degree)**		
		1	1.5	2
Type1	3×3	66.94	50.00	48.61
Type2	3×3	65.56	50.00	44.44
Type3	3×3	66.66	55.56	47.22

We have evaluated the robustness of the proposed scheme to the PS distortion. The results are presented in Table 5.2 in which the last three columns depict the accuracy ratio of data detection in case of distortion caused by a combination of Gaussian lowpass filter and rotation with three different angles. We can see that the resistance to printing and scanning distortion has not met the requirement of a practical application, this low performance is due to the mismatch of feature points extracted from the cover document and stego-document, and the lack of precision of algorithm when hiding and detecting data.

Besides, we also compare our method with other typical steganography methods for nat-

Table 5.3: *Comparison of our steganography scheme with Lin[12] and Soleymani[13]*

Images	Our method		Lin[12]		Soleymani[13]	
	PSNR	Capacity	PSNR	Capacity	PSNR	Capacity
Boat	61.83	732	-	-	37.00	1,494,221
Man	61.02	879	-	-	37.14	1,410,335
Pepper	61.37	537	-	-	36.19	1,481,114
Lena	61.30	554	51.14	262,144	36.91	1,410,335
Baboon	60.08	1,096	51.13	262,144	-	-
Jet	62.97	597	51.14	262,144	-	-
Scene	61.16	568	51.14	262,144	-	-

ural images, specifically we compare the performance of data hiding schemes on standard grayscale test images. The comparison shows that the capacity of our approach is lower than the methods presented by Lin[12] and Soleymani[13]. However, our method gives better quality for stego-images. In addition, it is not robust against common distortions. Despite of lower capacity, our scheme enables to hide a secret message whose length is long enough for document authentication. The comparison results on natural images from the dataset of standard grayscale test images are presented in Table 5.3.

Improvement the robustness of our steganography scheme

As discussed above, the steganography scheme based on feature points extracted from well-known detectors gives low performance in terms of documents. Thus, we have proposed another method to detect feature points as presented in Section 3.2.5, which is more stable than the existing approaches. To prove the robustness of detectors against distortions, we have conducted a new experiment on documents and natural images on the three mentioned datasets, including 20 documents from Tobacco, 20 documents from L3iDocCopies and 10 images from standard grayscale images. The feature points are extracted from documents and images compressed at various quality factors. The average stability of feature point extraction against JPG compression by using SIFT, SURF, BRISK and our detector are shown in Figure 5.4.

Besides, we have employed the feature points extracted from our detector to develop a data hiding scheme similar to method presented in Section 3.2.3. It just differs in data hiding and detection algorithm. This algorithm is based on the mean of two appropriately neighboring pixels within a 3×3 hiding pattern. Let p_1, p_2 and p_3 be three consecutive pixels corresponding to positions of bit 1 of the i^{th} corner pattern, the pixel value p_2 is then adjusted to hide data by:

$$p_2 = \begin{cases} (p_1 + p_3)/2 - T & , w_i = 0 \\ (p_1 + p_3)/2 + T & , w_i = 1 \end{cases} \tag{5.9}$$

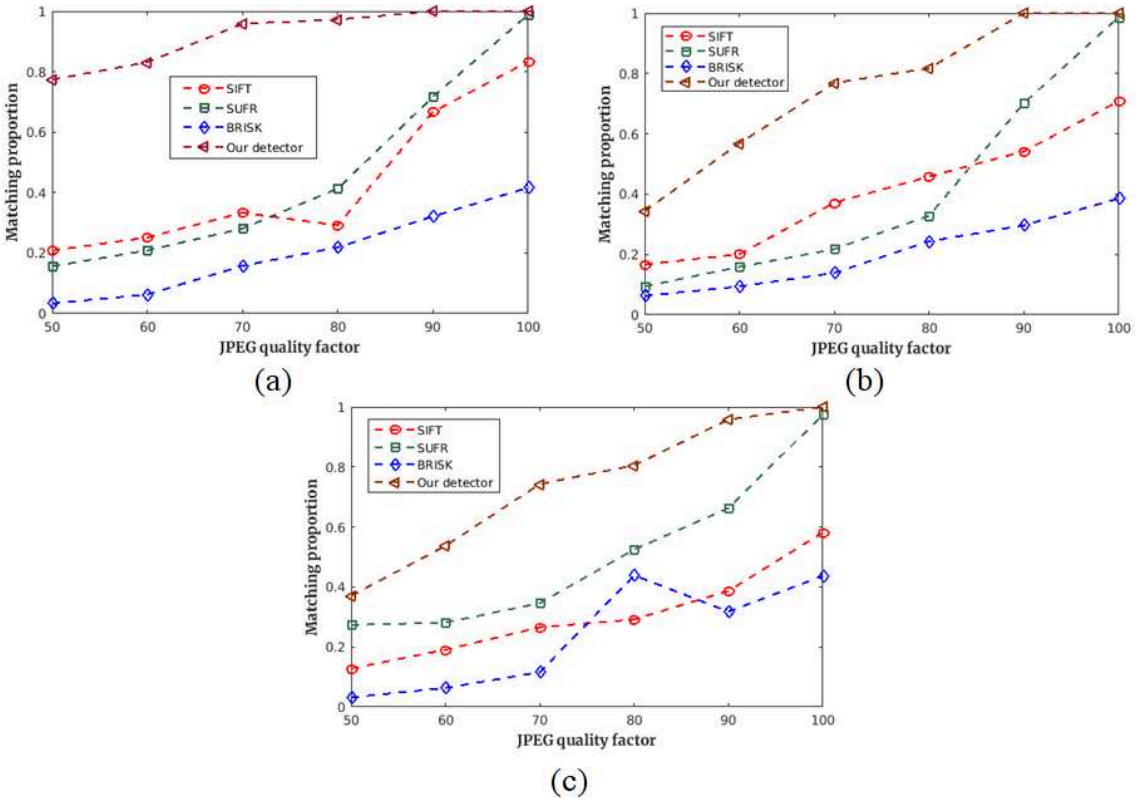


Figure 5.4: *The average stability of extracting feature points by using various detectors.*

where T is a threshold, w_i is the i^{th} secret bit.

Table 5.4: *The measurement of imperceptibility and capacity.*

Documents	PSNR (cover vs stego)	SSIM (cover vs stego)	Capacity (bits)
Type1	35.23	0.92	2,478
Type2	38.49	0.95	4,357
Type3	37.14	0.91	1,073

With this hiding method, the hidden information bits are extracted by:

$$w_i = \begin{cases} 0 & , p_2 \leq (p_1 + p_3)/2 \\ 1 & , otherwise \end{cases} \tag{5.10}$$

We have implemented our improvement-based steganography scheme on various documents from Tobacco and L3iDocCopies dataset. The measurement of imperceptibility and capacity on the sample documents (Figure 5.1) are presented in Table 5.4, and the results of robustness against JPEG compression is presented in Figure 5.5. Obviously, we can see that the scheme performance on lossy compression has been slightly improved as compared to method presented in Section 3.2.3. However, the robustness of scheme on geometric distortion still has been low, so it needs to be further improved in order to apply for the real

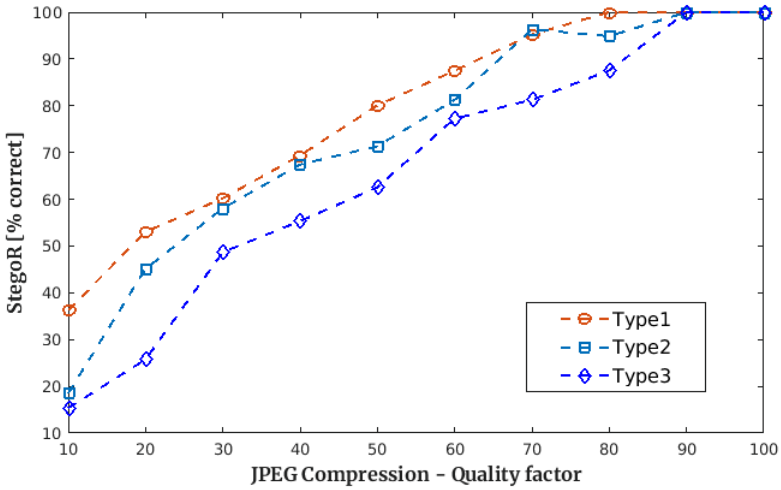


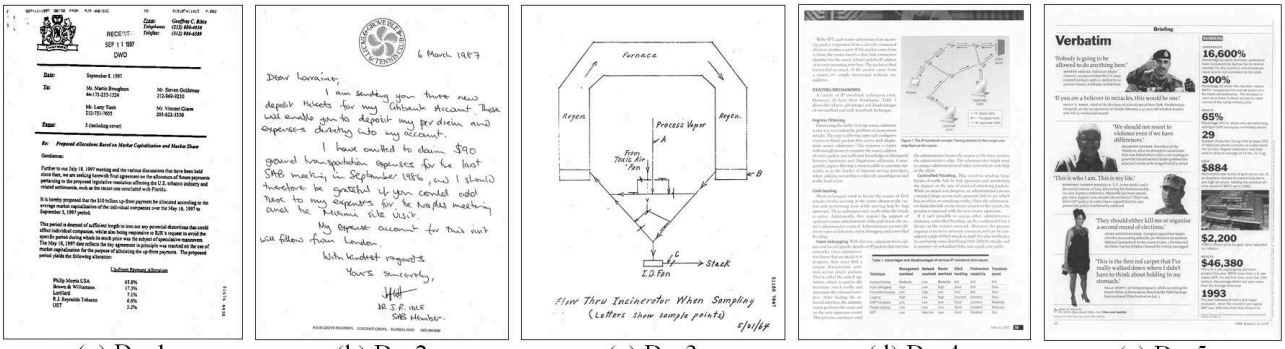
Figure 5.5: *Average accuracy ratio according to the JPEG quality factor.*

applications.

In this section, we have developed the feature points-based steganography schemes for document authentication. For the feature points extraction, we utilize the well-known SURF detector, and propose a new feature point detector for stability improvement in terms of document distortions. For the algorithm of data hiding and detection, we make use the parity of the pixel values, and the group of three consecutive pixels.

5.3 Watermarking scheme based on stable regions and object fill (STA-WM)

This section demonstrates the performance of the approach presented in Section 3.3. In this scheme, the experiment is conducted on documents with various contents, which are selected from two datasets like Tobacco and L3iDocCopies. For Tobacco (Type I), we opt for 20 documents. For L3iDocCopies, we choose 60 documents including: 15 documents scanned from Konica Minolta Bizhub 223 at the resolution of 300 dpi (Type II) and 15 documents scanned at the resolution of 600 dpi (Type III); 15 documents scanned from Fujitsu fi 6800 at the resolution of 300 dpi (Type IV) and 15 documents scanned at the resolution of 600 dpi (Type V). We have totally tested our approach on 80 various documents. The watermark used in this experiment is "watermarking-information", this text message is modulated into 192 bits. Besides, the parameters of watermarking scheme are set as follows. The distance for separating the hiding range is set with $D = 30$. The number of gray level values for carrying each watermark bit is assigned to $m = 4$. These values are experimentally chosen to provide a good tradeoff among robustness, imperceptibility and capacity. If D is too small, the robustness will be diminished. Otherwise, the imperceptibility will be degraded; If m is too large, the capacity will be low.



(a)-Doc1 (b)-Doc2 (c)-Doc3 (d)-Doc4 (e)-Doc5

Figure 5.6: *Sample documents.*

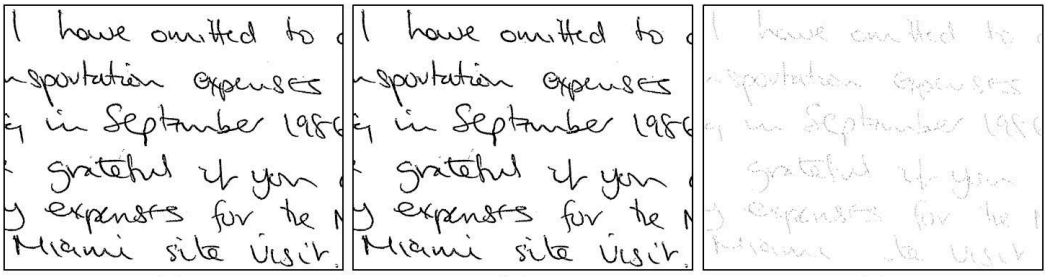
The sample documents are manifested as in Figure 5.6 in which documents (a), (b) and (c) with corresponding size of 1695×2262, 2502×3240 and 2514×3190 represent Tobacco dataset whereas documents (d) and (e) with size of 2480×3507 and 4960×7015 represent L3iDocCopies dataset, which are scanned at the resolution of 300 and 600 dpi. With these sample documents, the value of c is set to 1401, 2009, 1931, 2115 and 4203 for (a)-(e) respectively, based on the size of the bounding box. The performance of our scheme is evaluated depending on the following factors.

Imperceptibility and capacity

The maximum number of watermark bits that can be hidden into a document is estimated by:

$$Capacity = \sum_{i=1}^{n} \left\lfloor \frac{length(O_i)}{m} \right\rfloor \tag{5.11}$$

where $length(O_i)$ is the number of pixels corresponding to the i^{th} object's filling part, m is the length of a group of gray level values for carrying one watermark bit, and n is the number of separated objects extracted from a document.



(a) (b) (c)

Figure 5.7: *Imperceptibility and adjusted positions for watermarking: (a)-(c) are respectively the host document, watermarked document and adjusted positions.*

The imperceptibility is evaluated by measuring the difference between the original and watermarked document. Peak-Signal-to-Noise Ratio (PSNR) has been adopted in this work. To evaluate the quality of watermarked document, the watermark bits used in this part is randomly generated based on the maximum number of bits that the host documents can carry. Table 5.5 demonstrates PSNR and capacity of sample documents and the average values of 80 documents. Figure 5.7 shows an example where 2092 random message bits have been hidden into a host document with a size of 984×764, the PSNR of this watermarked document is 53.95. We can see hidden positions (black dots in Figure 5.7(c)) where the gray level values have been adjusted. For evaluation of our scheme's performance in the environment without noises, we hide the mentioned watermark into these 80 documents. The hidden message has been correctly retrieved in all watermarked documents.

Table 5.5: *The quality of watermarked documents and capacity*

Documents	PSNR (original vs watermarked)	Capacity (bits)
1 (Doc1)	52.45	40,738
2 (Doc2)	53.66	28,737
3 (Doc3)	54.70	21,057
4 (Doc4)	52.13	38,238
5 (Doc5)	52.09	45,247
6	51.10	17,135
7	52.13	11,431
8	50.78	24,280
...	...	...
Average (80 documents)	**51.81**	**25,148**

Robustness evaluation

We present below the robustness of our scheme against JPEG compression, geometric distortion and printing and scanning distortions, which often affect the legal documents.

Geometric transformation and JPEG compression: We have conducted tests on the scheme under various distortions such as: document rotation with rotation angle varying from 1 degree to 10 degrees with a step of 1 degree; document scaling with scale factor varying from 0.5 to 1.5 with a step of 0.1; a combination between rotation and scaling; document affected by JPEG compression with quality factor varying from 10 to 100 with a step of 10. The results of these distortions are shown in Table 5.6. In this table, we just present the levels of distortion that the watermarked documents are able to suffer from, and one level of distortion that the watermarked documents get completely failed in detecting the hidden information. We can see that the performances of our scheme are capable of resisting to JPEG compression with a low quality factor of 50, rotation up to 5 degrees, scaling down to 0.8 and up to 1.2, a combination between rotation of 5 degrees and scaling of 0.9 and 1.1.

Printing and scanning distortion: We have studied the robustness of our algorithm against print and scan attack in four conditions such as very small geometric transformation, with

Table 5.6: *The accuracy ratio of watermark detection on various distortions*

Lossy compression and geometric distortions	Bit Error Rate					
	Doc1	Doc2	Doc3	Doc4	Doc5	Avg. (80 documents)
No noises	0	0	0	0	0	0
1° rotation	0	0	0	0	0	0
3° rotation	0	0	0	0	0	0
5° rotation (a)	0	0	0	0	0	0.06
7° rotation	0.14	0.12	0.15	0.17	0.15	0.19
Scaling 0.8	0	0	0.11	0.16	0	0.13
Scaling 0.9 (b)	0	0	0	0	0	0
Scaling 1.1 (c)	0	0	0	0	0	0
Scaling 1.2 (d)	0	0	0	0.07	0	0.09
(a) + (b)	0	0	0	0	0	0.08
(a) + (c)	0	0	0	0	0	0.03
(a) + (d)	0.25	0.24	0.27	0.30	0.28	0.21
JPEG 90%	0	0	0	0	0	0
JPEG 60%	0	0	0	0	0	0
JPEG 50%	0	0	0	0.13	0	0.11
JPEG 40%	0.12	0.15	0.22	0.26	0.23	0.21

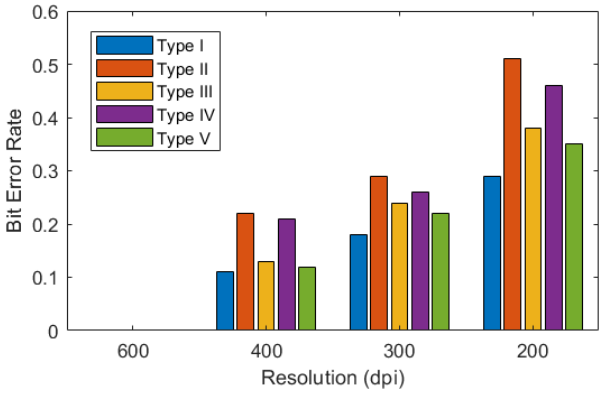


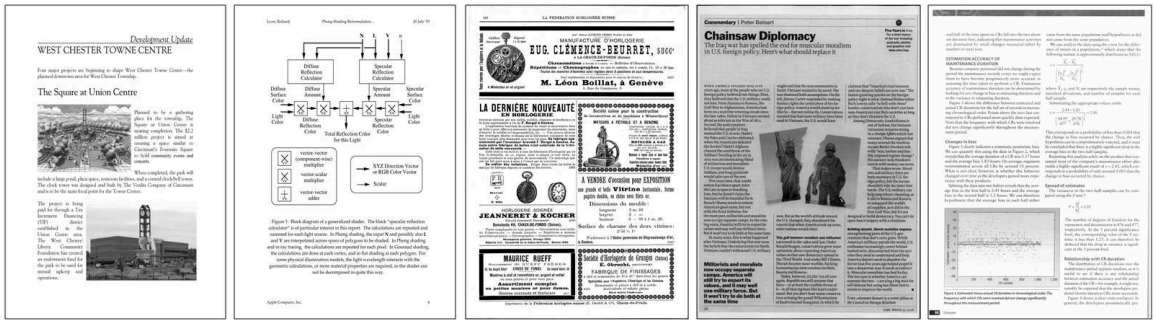
Figure 5.8: *The average results of watermark detection on PS noise.*

rotation distortion, with scaling distortion and with a combination of rotation and scaling. After hiding the watermark, we use machine named Kyocera TASKalfa 3252ci KX for printing the watermarked documents at the default resolution of 600 dpi. This machine is also used for scanning the printed versions of these watermarked documents. The scanning resolution is sequentially chosen to be equal to the resolution of 200, 300, 400 and 600 dpi. As a result, our scheme properly works in case of printing and scanning the watermarked documents with high resolution of 600 dpi. The BER is substantially increased as scanning at lower resolutions due to the degradation of quality of watermarked documents. Besides, we have found that the scanned documents more or less suffer from geometric distortions. Figure 5.8 shows the average of BER detected from five types of watermarked documents at varying resolutions.

In general, we have proposed a new watermarking scheme in which the stable regions are detected by using the combination of the common image processing operations and non-subsampled contourlet transform. Meanwhile, the watermarking algorithm has been developed relied on the hiding factor of each group of successive pixel values. The proposed scheme has significantly been improved compared to the previous steganography scheme.

5.4 Watermarking scheme for securing documents using FCN (PAT-WM)

This section details the experimental results of the scheme presented in Section 4.2. For training configurations, we use two datasets: PRImA and DSSE-200. In the context of our work, we expect invariance to rotation, scale variation and variation of the quality factor of the lossy compression. Thus, we generate 5080 documents from 254 original document images as training samples. Regarding initialization of network parameters, the number of learning steps is set to 200,000. The high momentum is assigned to 0.9. The weight decay is 5×10^{-4}. The learning rate is set to 10^{-4}, and it is adjusted to 10^{-5} when reaching half of the learning steps. The dropout rate is assigned to 0.5.



(a)-Doc1 (b)-Doc2 (c)-Doc3 (d)-Doc4 (e)-Doc5

Figure 5.9: *Sample documents with various content: (a), (b) and (c) from DSSE-200 dataset. (d) and (e) from L3iDocCopies dataset.*

For watermarking configurations, we select 15 documents from DSSE-200 (Type-1). 60 documents are selected from L3iDocCopies[120] including: 15 documents scanned from Konica Minolta Bizhub 223 at the resolution of 300 dpi (Type-2) and 15 documents scanned at the resolution of 600 dpi (Type-3); 15 documents scanned from Fujitsu fi 6800 at the resolution of 300 dpi (Type-4) and 15 documents scanned at the resolution of 600 dpi (Type-5). We have tested our approach on 75 various documents. The watermark used in this experiment is "watermarking-information", this text message is modulated into 192 bits. The distance for separating the mean values is set with $W = 25$. The size of the watermarking pattern is assigned to $m = 3$. These values are experimentally chosen to provide a good tradeoff among robustness, imperceptibility and capacity (if W and m are too small, the robustness will be diminished. If W is large, the imperceptibility will be degraded. If m is too large, the capacity will be low). With the sample documents in Figure 5.9, the value of c is set to

1836, 1765, 2414, 2147 and 4223 for (a)-(e) respectively, and it is estimated relied upon the size of minimum rectangle of entire document. The performance of our scheme is evaluated depending on the following factors.

Imperceptibility and capacity

The quality of watermarked document is evaluated by measuring the difference between original and watermarked document. Peak-Signal-to-Noise Ratio (PSNR) has been adopted in this work. The capacity of a document is measured by the total number of watermarking patterns satisfying the condition described in Section 4.2.3. To measure the imperceptibility, the watermark bits used in this part is randomly generated depending on the maximum number of bits that the host documents can contain. The quality of watermarked documents, capacity of sample documents and the average values of 75 testing documents are given in Table 5.7.

Table 5.7: *The measurement of imperceptibility and capacity*

Documents	PSNR (original vs watermarked)	Capacity (bits)
1 (Doc1)	44.60	16,115
2 (Doc2)	46.59	3,417
3 (Doc3)	44.60	15,711
4 (Doc4)	42.58	16,962
5 (Doc5)	44.26	9,533
6	41.42	4,157
7	44.27	2,425
8	40.53	11,478
...	...	...
Average (75 documents)	**43.02**	**9,167**

To demonstrate the quality of documents after modifing for watermarking, Figure 5.10 shows an example of hiding 132 random information bits into a host document with size of 208×186 in which the PSNR of this watermarked document is 46.63. We can see the positions of the document where the pixel values are adjusted for hiding information, and they are depicted as the white vertical lines in Figure 5.10(c). To evaluate the performance of our scheme in the environment without distortions, we hide the mentioned watermark into these 75 documents. The hidden message has been correctly retrieved in all watermarked documents.

Robustness evaluation

In this part, we would like to prove the robustness of our scheme against JPEG compression, geometric distortion and printing and scanning distortions, which often affect the legal documents.

Figure 5.10: *The demonstration of host document (a), watermarked document (b) and adjusted pixel positions (c).*

Geometric transformation and JPEG compression: Similar to the assessment of robustness of watermarking scheme presented in Chapter 3, we still have conducted tests of our scheme under various distortions. We use the same protocol than the previous experiments presented in Section 5.3. We demonstrate the robustness of our scheme on the sample documents in details. The results of the robustness against distortions are shown in Table 5.8.

Table 5.8: *The precision of watermark detection on JPEG compression and geometric distortions*

Lossy compression and geometric distortions	Bit Error Rate					
	Doc1	Doc2	Doc3	Doc4	Doc5	Avg. (75 documents)
No noises	0	0	0	0	0	0
Rotation 1°	0	0	0	0	0	0
Rotation 3°	0	0	0	0	0	0
Rotation 5° (a)	0	0	0	0	0	0.05
Rotation 7°	0.16	0.18	0.13	0.21	0.17	0.15
Scaling 0.8	0	0	0	0.28	0	0.09
Scaling 0.9 (b)	0	0	0	0	0	0
Scaling 1.1 (c)	0	0	0	0	0	0
Scaling 1.2 (d)	0	0	0	0.25	0	0.07
(a) + (b)	0	0	0	0	0	0.08
(a) + (c)	0	0	0	0	0	0.06
(a) + (d)	0.27	0.23	0.29	0.31	0.26	0.24
JPEG 90%	0	0	0	0	0	0
JPEG 60%	0	0	0	0	0	0.03
JPEG 50%	0	0	0	0.24	0	0.08
JPEG 40%	0.17	0.16	0.21	0.34	0.25	0.19

In this table, we just present the levels of distortion that the watermarked documents are able to suffer from, and one level of distortion that the watermarked documents get failed in detecting the hidden information. We can see that the performance of our scheme

is capable of resisting to JPEG compression with a low quality factor of 50, rotation of 5 degrees, scaling down to 0.8 and up to 1.2, a combination between rotation of 5 degrees and scaling of 0.9 and 1.1. However, the accurate ratio of information detection is degraded regarding the host documents, e.g. the content quality of Doc4 is degraded because it has been scanned at low resolutions.

Printing and scanning distortion: We printed and scanned several watermarked documents using commercially integrated printer and scanner named Kyocera TASKalfa 3252ci. In these experiments, we assume a degree of control over the printing and the scanning operation. The watermarked documents were printed at high resolution of 600 dpi, with several dots dedicated to one pixel. The printed version of these watermarked documents is scanned at varying resolutions of 200, 300, 400 and 600 dpi. There are several factors that degrade the quality of printed and scanned version of documents such as printer and scanner resolutions, rotation due to placing paper on the printer and scanner screen, rotation due to loading paper from the paper tray of printing machine. To minimize these effects to the scheme performance, they need to be eliminated by transforming the input document into its standard form as described in Section 4.2.2. As a result, our scheme properly works in case of printing and scanning the watermarked documents with high resolution of 600 dpi. The precision of information detection is substantially decreased when scanning at lower resolutions due to the degradation of quality of watermarked documents. The average accuracy of watermark detection at varying resolutions is presented in Figure 5.11.

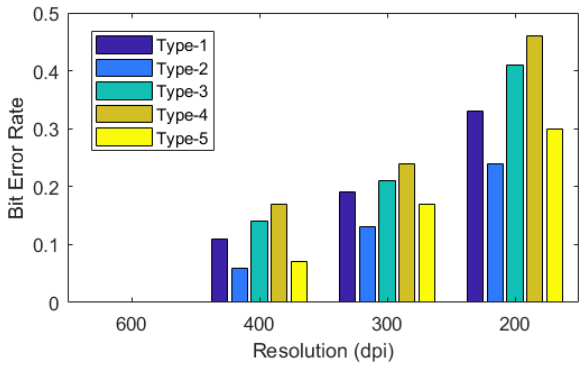
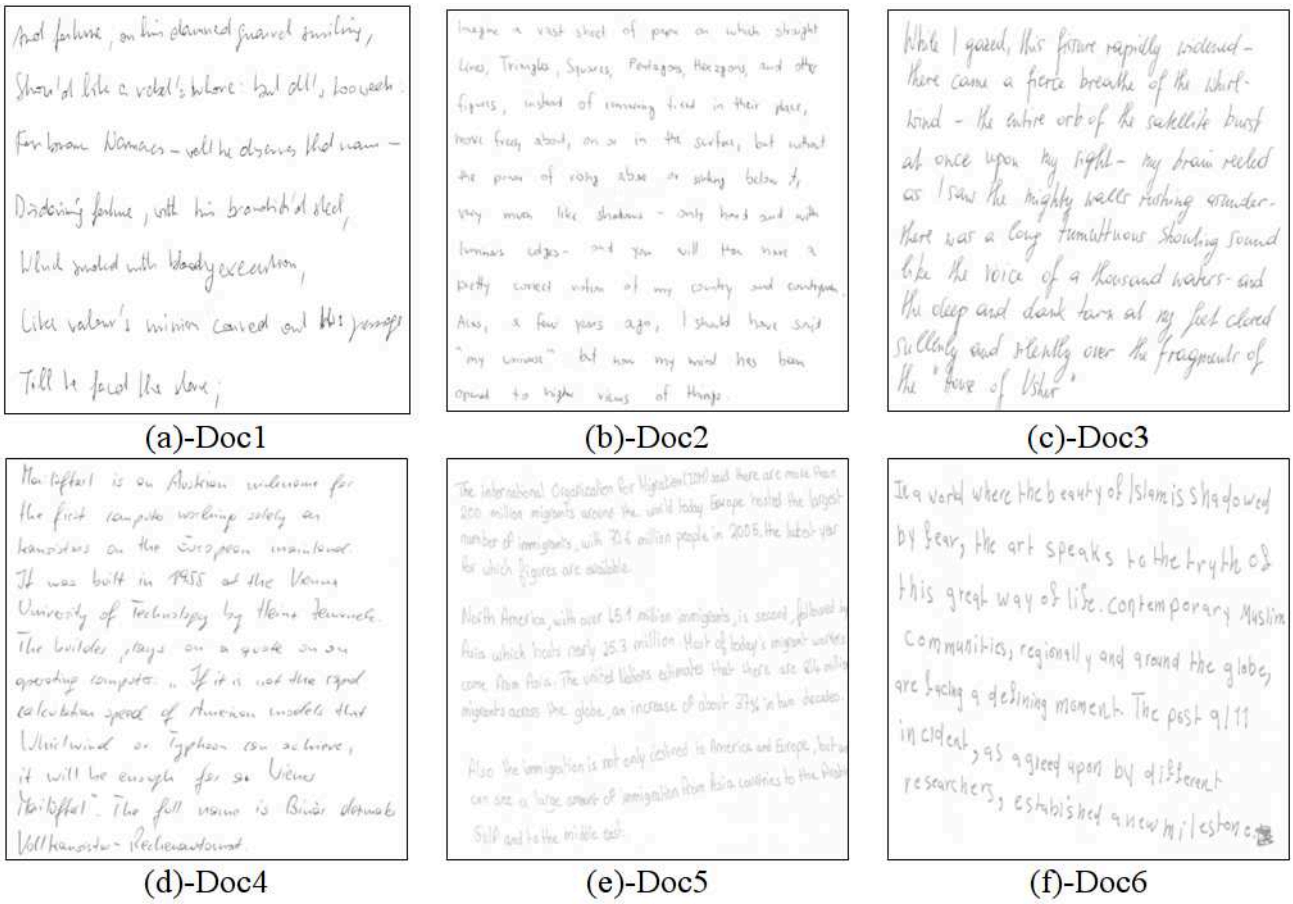


Figure 5.11: *The average results of watermark detection on PS distortion.*

In brief, we have improved the robustness of the scheme by making use the FCN network for detecting the watermarking regions. The algorithms of data hiding and detection are designed based on two mean values corresponding to two groups of pixel values of each watermarking pattern. The scheme can resist to JPEG compression with quality factor down to 50, the rotation of 5 degrees in conjunction with the scale of 0.9 or 1.1. Besides, it can be robust against the PS operations with a resolution of 600 dpi.

5.5 Watermarking scheme for handwritten documents using FCN (HAN-WM)

This section details the experimental results of the scheme presented in Section 4.3. For training FCN network, dataset from CVL-database is selected to train our network. With regard to initialization of network parameters, the maximum number of learning steps is set to 200,000. The high momentum is assigned to 0.9. The weight decay is 5×10^{-4}. The learning rate is fixed to 2^{-6} during the training phase. The dropout rate is assigned to 0.5.



(a)-Doc1 (b)-Doc2 (c)-Doc3

(d)-Doc4 (e)-Doc5 (f)-Doc6

Figure 5.12: *Sample documents from different writers: (a), (b) and (c) with corresponding size of* 2499×1726, 2529×1670 *and* 2530×1870 *represent Type-1 dataset. (d) and (e) with size of* 2471×1705 *and* 2255×1062 *represent Type-2 dataset.*

For watermarking scheme, we use documents from two datasets published in ICDAR 2013[149;173] for testing our approach: 35 documents are selected from[149] including 7 various handwriting text contents from 5 different writers (noted as Type-1 document); 40 documents are picked from[173] consisting of 4 various handwriting text contents from 10 different writers (noted as Type-2 document). Totally, we have experienced our approach on 75 various handwriting documents. The secret message used in this experiment is "secret-information",

this text message is converted into 144 bits. The values used to adjust gray level values of document for carrying watermark bits are set to $\delta_1 = 400, \delta_2 = 4$. These values are experimentally selected to give a good tradeoff between robustness and imperceptibility. If δ_1 and δ_2 are small, the robustness of the scheme against the distortions will be reduced. Otherwise, the imperceptibility will be diminished. Figure 5.12 shows the sample documents in which we crop the empty region around content for saving display space. The value of the constant c is assigned to 1147, 1337, 1341, 1345, 1464, 1237 corresponding respectively to Doc1, ..., Doc6. These values are estimated relied on the minimum rectangle of original handwriting documents. The effectiveness of our approach is evaluated depending on the following factors.

Imperceptibility and capacity

The possible number of hidden bits could be maximally equal to the number of letters of document if these letters are separately written. The capacity will be reduced if the letters of the document are connected together due to writing style of writers. The imperceptibility is evaluated by assessing the difference between pre-processing and watermarked document. The peak signal-to-noise ratio (PSNR) and the structural similarity index (SSIM) have been adopted in this work. The secret message bits used in this part is randomly generated depending on the maximum number of bits that the handwriting documents can contain. The quality, capacity and average values of 75 documents are shown in Table 5.9 wherein their minimum and maximum values are illustrated in color.

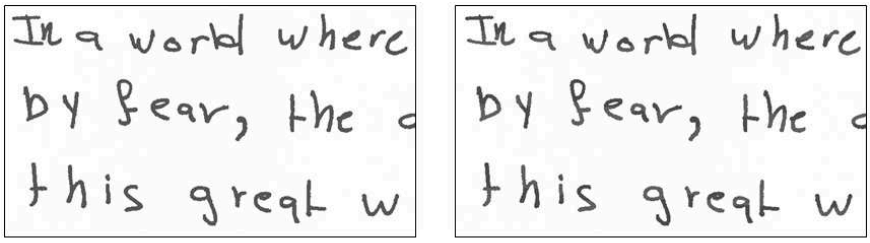
Table 5.9: *Imperceptibility (pre-processing vs watermarked) and capacity*

Documents	PSNR	SSIM [0, 1]	Capacity (bits)
1 (Doc1)	46.54	0.99981	170
2 (Doc2)	45.88	0.99980	348
3 (Doc5)	42.26	0.99916	521
4	44.83	0.99976	133
5	41.66	0.99716	554
6	41.71	0.99437	434
7	46.63	0.99984	187
...	...	...	...
Average (75 documents)	**44.52**	**0.99877**	**327**

Figure 5.13 shows an example of hiding 30 random bits into a small document. The PSNR and SSIM of this watermarked document are 42.89 and 0.99906 respectively.

Robustness evaluation

To evaluate the effectiveness of the approach in the environment without distortions, we hide the mentioned secret information into these 75 documents. As a result, the hidden



(a) (b)

Figure 5.13: *The imperceptibility between pre-processing document (a) and watermarked document (b).*

data have been successfully detected on all watermarked documents. Regarding robustness against distortions, we demonstrate the accuracy ratio of watermark detection over popular degradations that documents often undergo as follows.

Geometric transformation and JPEG compression: Like watermarking schemes presented in previous chapter, we use the same protocol than the previous experiments presented in Section 5.3. The results of these distortions are shown in Table 5.10. In this table, for saving space, we present the levels of distortion that the watermarked documents are able to suffer from, and one level of distortion that the watermarked documents get failed in detecting the hidden information. We can see that the performance of our scheme is capable of resisting to JPEG compression with a low quality factor of 40, rotation of 7 degrees, scaling down to 0.8 and up to 1.3, a combination between the rotation of 5 degrees and scaling of 0.8 and 1.3.

Table 5.10: *The accuracy rate of information detection on lossy compression and geometric distortion*

Lossy compression	Accuracy Ratio						
and geometric distortions	Doc1	Doc2	Doc3	Doc4	Doc5	Doc6	Avg. (75 documents)
JPEG 40%	1	1	1	1	1	1	1
JPEG 30%	0.88	0.82	0.85	0.83	0.79	0.86	0.81
Rotation 5° (a)	1	1	1	1	1	1	1
Rotation 7° (b)	1	1	1	1	1	1	0.97
Rotation 9°	0.86	0.80	0.76	0.79	0.85	0.83	0.84
Scaling 0.7	0.79	0.82	0.75	0.84	0.81	0.78	0.83
Scaling 0.8 (c)	1	1	1	1	1	1	1
Scaling 1.3 (d)	1	1	1	1	1	1	1
Scaling 1.4 (e)	0.83	0.87	0.82	0.77	0.80	0.84	0.81
(a) + (c)	1	1	1	1	1	1	0.96
(a) + (d)	1	1	1	1	1	1	1
(b) + (e)	0.69	0.71	0.74	0.67	0.64	0.72	0.72

Printing and scanning distortion: We use the same protocol than the previous experiments presented in Section 5.3. Figure 5.14 shows the average results of accuracy ratio of watermark detection. Our approach gives the highest accuracy rate when the watermarked documents

are printed and scanned at the resolution of 600 dpi. At the lower resolutions, the accuracy ratio of watermark detection is considerably diminished. The main factors causing this reduction are the inconsistency of watermarking regions and separated objects extracted from the distorted watermarked documents.

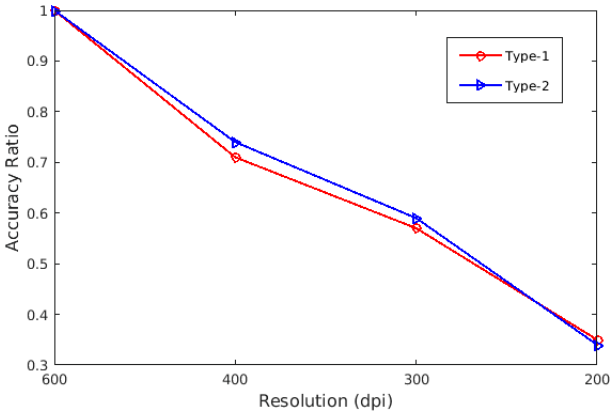


Figure 5.14: *The average accuracy of information detection on PS distortion.*

On the whole, we have proposed a watermarking system able to protect the handwriting documents. The FCN network is utilized to detect the document content where the secret information is hidden into. The algorithms of data hiding and detection are developed depending on the difference between two sets of pixel values of each connected handwritten element. The scheme can resist to JPEG compression with quality factor down to 40, the rotation of 5 degrees in conjunction with the scale down to 0.8, or up to 1.3. Besides, it can be robust against the PS operations with the resolution of 600 dpi. This scheme can be effectively applied for the typewriting documents.

5.6 Watermarking scheme using generative adversarial networks (GEN-WM)

This section demonstrates the experimental results of the scheme presented in Section 4.4.

For training our GAN network, we use dataset DSSE-200 published in[121] which contains 200 document pages from magazines and academic papers. For the context of our work, we synthesize our training dataset consisting of 900 documents in which we hide randomly generated data bits into 50 documents with three various thresholds ($\lambda = 20, 30, 40$) which are added to the pixel values for carrying message bits. The hidden documents are then subjected to blur with five different kernels. Besides, we have created 150 documents with real PS noise. The documents with data hiding and noises form the set of distored documents, and the corresponding clean documents are regarded as the set of real documents. Regarding network parameters, Adam optimizer is used for optimization with a mini-batch size of 1. The learning rate starts from 0.0002 and is divided by 10 after reaching half of the number

of epoches. The weighting parameters of function loss are set to $\alpha_g = 0.005$, $\alpha_e = 1$ and $\alpha_f = 0.9$.

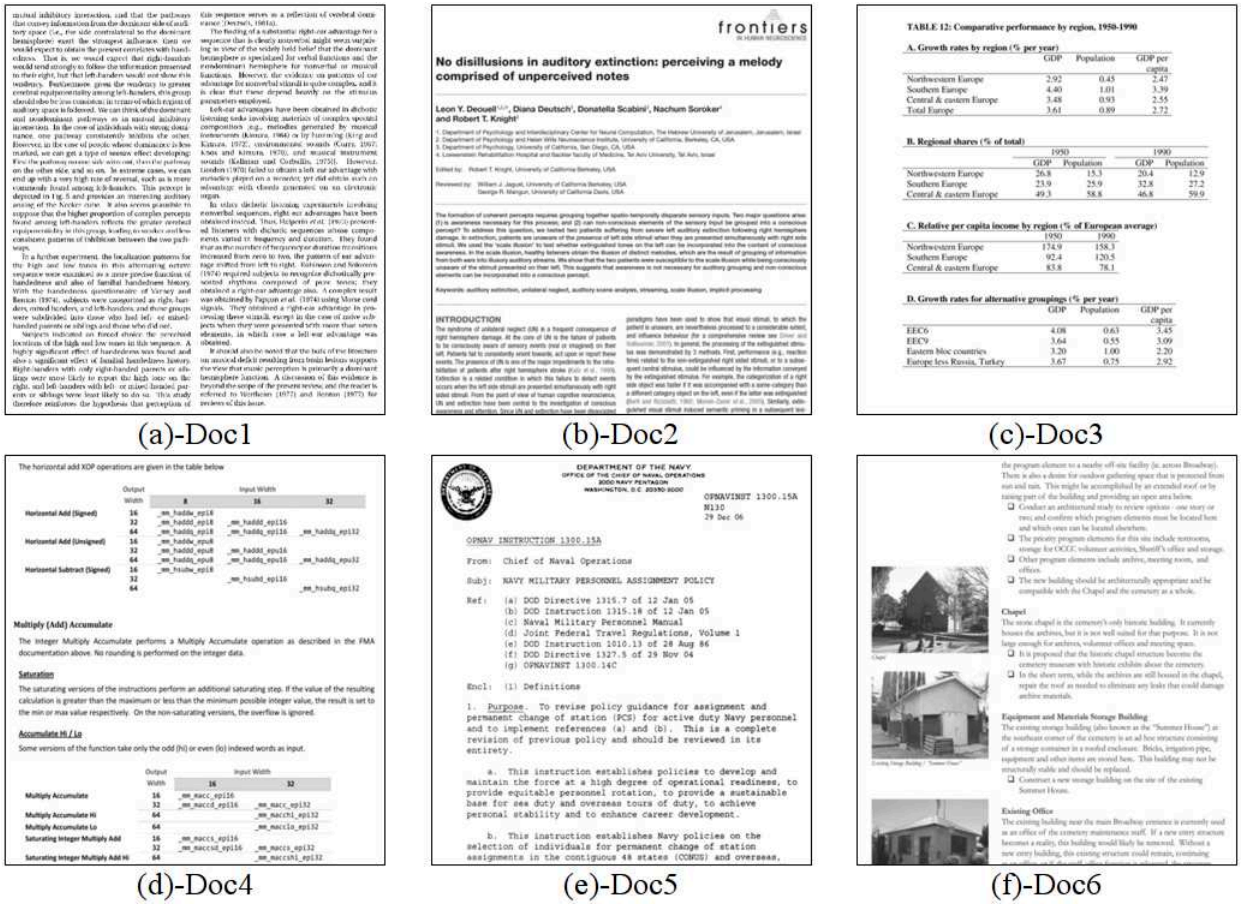


Figure 5.15: *The illustration of sample documents with various resolutions, fonts and layouts.*

For watermarking scheme, we also use documents published in[121] for testing our proposed approach. Totally, we have conducted our experiment on 70 documents with various resolutions and fonts. The watermark used in this experiment is "watermarking-scheme-GAN", this watermark is converted into a message of 184 bits. The thresholds used to adjust pixel values of document content for holding secret bits are set to $\lambda = 15$ and $\lambda = 25$ (a value added to the pixel values). These values are experimentally chosen to provide a good tradeoff between imperceptibility and robustness. If λ is small, the scheme resistance to distortions will be diminished. Otherwise, the quality of watermarked documents will be reduced. The illustration of sample documents with various contents are shown in Figure 5.15. The performance of our approach is presented through the following factors.

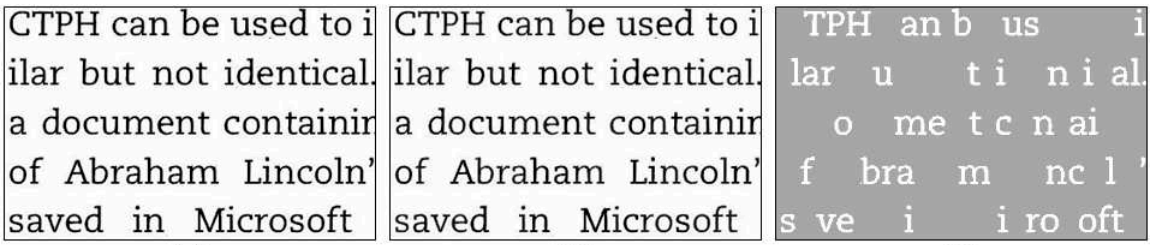
Imperceptibility and capacity

The capacity (bits) is measured by the number of bounding boxes extracted by OCR while the quality of watermarked documents (imperceptibility) is evaluated by the difference between the document before and after hiding a secret information. We adopt peak signal to noise ratio (PSNR) and the structural similarity index (SSIM)[171] for estimating the quality of watermarked documents. The watermark bits used in this part are randomly generated relied upon the maximum number of bits that a document can carry. The imperceptibility, capacity and average values of 70 testing documents are shown in Table 5.14 wherein the minimum and maximum values are illustrated in blue and red color respectively. In this table, we just present a few records of sample documents and others which hold the minimal and maximal values of PSNR and SSIM.

Table 5.11: *Imperceptibility (generated vs watermarked) and capacity*

Documents	PSNR		SSIM		Capacity
	$\lambda = 15$	$\lambda = 25$	$\lambda = 15$	$\lambda = 25$	
1 (Doc1)	35.52	31.54	0.9988	0.9972	338
2 (Doc2)	35.86	32.25	0.9984	0.9968	228
3	36.02	32.04	0.9990	0.9977	395
4	37.14	33.34	0.9992	0.9982	247
5	37.15	33.39	0.9990	0.9980	248
6 (Doc3)	36.87	33.32	0.9991	0.9981	214
7	33.90	31.13	0.9983	0.9967	248
...	...	...	...	...	
Average (70 documents)	**35.82**	**32.28**	**0.9988**	**0.9974**	**265**

Figure 5.16 shows an example of hiding 87 random bits into a small document with threshold $\lambda = 25$ in which the watermarked document is shown in (b), and the white characters in document (c) mark positions whose pixel values are slightly changed during the hiding process (the missing characters in document (c) remain unchanged). We gain the PSNR and SSIM of this watermarked document to be 31.99 and 0.9974 respectively.



(a) (b) (c)

Figure 5.16: *An illustration of imperceptibility and capacity: (a) and (b) are a small generated document and watermarked document respectively, and the marked document (c) whose pixel values are adjusted after hiding 87 random bits.*

Robustness evaluation

Geometric transformation and JPEG compression: We use the same protocol than the previous experiments presented in Section 5.3. Table 5.12 presents the accuracy of detecting the hidden information when hiding data with threshold $\lambda = 15$ whereas Table 5.13 for threshold $\lambda = 25$.

Table 5.12: *The accurate ratio of extracted information ($\lambda = 15$)*

Distortions	Accuracy Ratio						
	Doc1	Doc2	Doc3	Doc4	Doc5	Doc6	Avg. (70 documents)
JPEG 50%	0.99	1	0.99	0.99	0.99	1	0.97
JPEG 30%	0.55	1	0.91	0.90	0.95	0.92	0.93
Rotation 3°	1	1	1	0.86	0.84	1	0.96
Rotation 5° (a)	1	1	1	0.31	0.47	1	0.91
Rotation 7°	1	1	1	0.45	0.41	1	0.86
Scaling 0.7 (b)	0.99	1	0.97	0.47	0.61	0.98	0.87
Scaling 0.8 (c)	0.96	1	1	0.45	0.67	1	0.94
Scaling 1.2 (d)	1	1	0.98	0.84	1	1	0.97
Scaling 1.3 (e)	0.97	1	1	0.66	0.95	1	0.90
(a) + (b)	0.93	1	0.90	0.42	0.56	0.97	0.87
(a) + (c)	0.99	1	1	0.41	0.52	1	0.93
(a) + (d)	1	1	1	0.63	0.53	0.99	0.95
(a) + (e)	0.98	1	1	0.53	0.46	1	0.88

Table 5.13: *The accurate ratio of extracted information ($\lambda = 25$)*

Distortions	Accuracy Ratio						
	Doc1	Doc2	Doc3	Doc4	Doc5	Doc6	Avg. (70 documents)
JPEG 50%	1	1	1	1	1	1	1
JPEG 30%	0.66	1	0.99	0.98	0.99	1	0.96
Rotation 3°	1	1	1	0.93	1	1	0.99
Rotation 5° (a)	1	1	1	0.55	0.58	1	0.99
Rotation 7°	1	1	1	0.49	0.84	1	0.96
Scaling 0.7 (b)	1	1	0.99	0.45	0.61	1	0.89
Scaling 0.8 (c)	0.98	1	1	0.39	0.69	1	0.97
Scaling 1.2 (d)	1	1	1	0.96	1	1	0.99
Scaling 1.3 (e)	1	1	1	0.79	1	1	0.93
(a) + (b)	0.99	1	1	0.46	0.45	1	0.96
(a) + (c)	1	1	1	0.42	0.53	1	0.98
(a) + (d)	1	1	1	0.65	0.57	1	0.93
(a) + (e)	1	1	1	0.61	0.47	1	0.91

During the experiments, we observe that the low accuracy mostly occurs on documents with low quality (the original documents are made by printing and scanning at low resolutions

or quality of PS machine). The inconsistency of extracted bounding boxes between the original and distorted documents partly affects the accuracy ratio of extracted data. For example, the extracted information like "waturearking-sche-e-GAN" and "wateroavking=scxe}e-GAN" corresponds to accuracy ratio of 0.98 and 0.97.

Printing, photocopying and scanning distortion: We use the same protocol than the previous experiments presented in Section 5.3. In addition, we also evaluate the robustness of our approach by adding more complicated noises by making two rounds of photocopying. The two-round-photocopying documents are also scanned back at several resolutions. Figure 5.17 shows the results of watermark detection. PS threshold-1 and PS threshold-2 depict robustness against PS distortion in case of hiding information into documents with threshold $\lambda = 15$ and $\lambda = 25$ whereas PCS represents resistance to PCS distortion in case of hiding data into documents with threshold $\lambda = 25$. For documents with sufficiently good quality, which are used to hide information, our scheme is capable of detecting the hidden information with high accuracy, even for some types of document scanned at the resolution of 200 dpi.

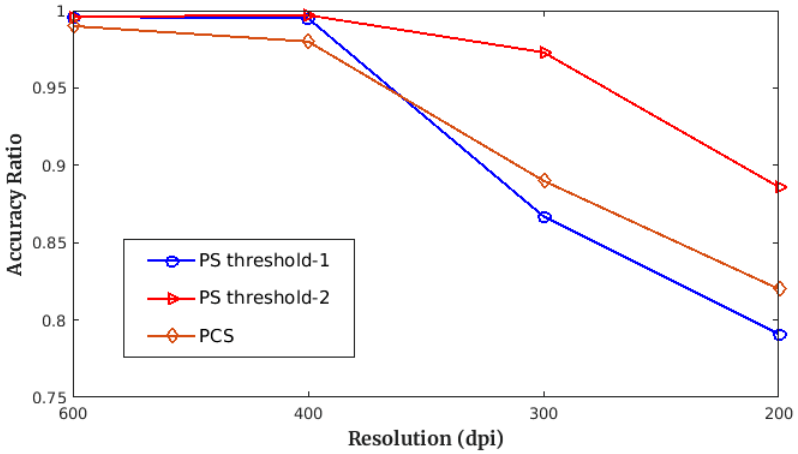


Figure 5.17: *The average results of extracted information on PS and PCS distortion.*

To conclude, we have improved the robustness of the scheme by using GAN for document generation, which is used as a reference for watermarking process. The algorithms of data hiding and detection are designed by measuring the absolute distance of pixel values between each object within the watermarked and generated documents. The scheme can resist to JPEG compression with quality factor down to 30, the rotation of 5 degrees in combination with the scale down to 0.7, or up to 1.3. Besides, it can be robust against the PS operations with the low resolutions, and two rounds of photocopying before scanning back at various resolutions.

5.7 Watermarking scheme based on font generation (VAR-WM)

This section demonstrates the experimental results of the scheme presented in Section 4.5.

For training our GAN network, we use the dataset DSSE-200 published in[121]. We utilize thinning algorithm[168] to make skeleton documents from these documents. The documents containing normal shape of characters are regarded as the set of real documents. Regarding initialization of network parameters, the Adam optimizer is used to optimize the network parameters with $\beta_1 = 0.5$ and $\beta_2 = 0.9$. The learning rate starts from 0.0002 and is divided by 10 after reaching half of the number of epochs. The weighting parameters are set to $\alpha_a = 0.005$ and $\alpha_f = 0.9$.

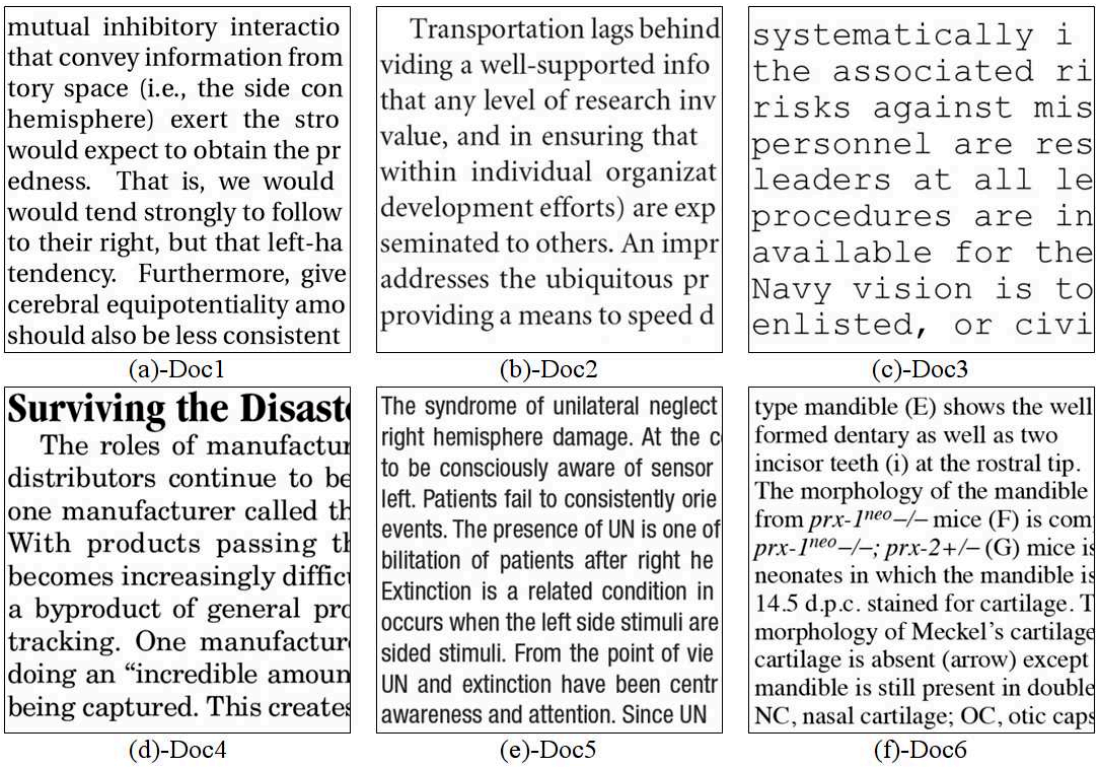


Figure 5.18: *Sample documents with various fonts and styles.*

For training our FCN network, we create 800 watermarked documents and annotated documents from 200 documents in which the watermarked documents are obtained by replacing appropriate characters with their corresponding variations generated from GAN network. The secret bits used to create this training documents are randomly generated. Besides, distortions caused by JPEG compression are also added to these documents. Meanwhile, the annotated documents are obtained by marking positions where the characters of document are substituted by their corresponding variations. The number of learning steps is set to 200,000. The high momentum is assigned to 0.9. The weight decay is 5×10^{-4}. The learning rate is set to 10^{-4}. The dropout rate is assigned to 0.5.

For watermarking scheme, we also use documents published in[121] for testing the proposed approach. We have conducted our experiments on a total of 70 documents with various contents and fonts. The watermark used in this experiment is "watermarking-scheme", and it is converted into 152 bits. We apply two types of character variants including "Variation 1" (Font1) and "Variation 3" (Font3) as in Figure 4.28(b) and 4.28(d) for depicting the performance of our scheme. The illustrations of sample documents with various contents are shown in Figure 5.18. The performance of our approach is presented through the following factors.

Imperceptibility and capacity

The capacity is measured by the number of bounding boxes containing document content extracted by OCR while the quality of watermarked documents (imperceptibility) is evaluated by the difference between the document before and after hiding a secret information. We adopt peak signal to noise ratio (PSNR) and the structural similarity index (SSIM) to estimate document quality. The watermark bits used in this part are randomly generated relied upon the maximum number of bits that a document can carry. The imperceptibility, capacity and average values of 70 testing documents are shown in Table 5.14 wherein their minimum and maximum values are illustrated in blue and red color respectively. In this table, for saving the displaying space, we just present the results of a limited number of documents whose the ones with the minimal and maximal values of PSNR and SSIM.

Table 5.14: *Imperceptibility (original vs watermarked) and capacity*

Documents	PSNR		SSIM		Capacity
	Font1	Font3	Font1	Font3	
1 (Doc1)	17.30	19.80	0.9113	0.9452	238
2 (Doc2)	19.24	21.55	0.9277	0.9562	201
3 (Doc3)	19.25	21.09	0.9251	0.9492	256
4	19.91	25.11	0.9213	0.9747	95
5	21.16	22.71	0.9464	0.9617	326
6 (Doc5)	16.86	21.39	0.8795	0.9555	238
7	18.18	21.33	0.9231	0.9568	357
...	...	... ...	...	...	
Average (70 documents)	**18.67**	**21.52**	**0.9234**	**0.9564**	**264**

Figure 5.19 shows an example of watermarked documents whose characters are substituted by their variations corresponding to Font1 and Font3.

Robustness evaluation

Geometric transformation and JPEG compression: We use the same protocol than the previous experiments presented in Section 5.3. Table 5.15 presents the accuracy of detecting

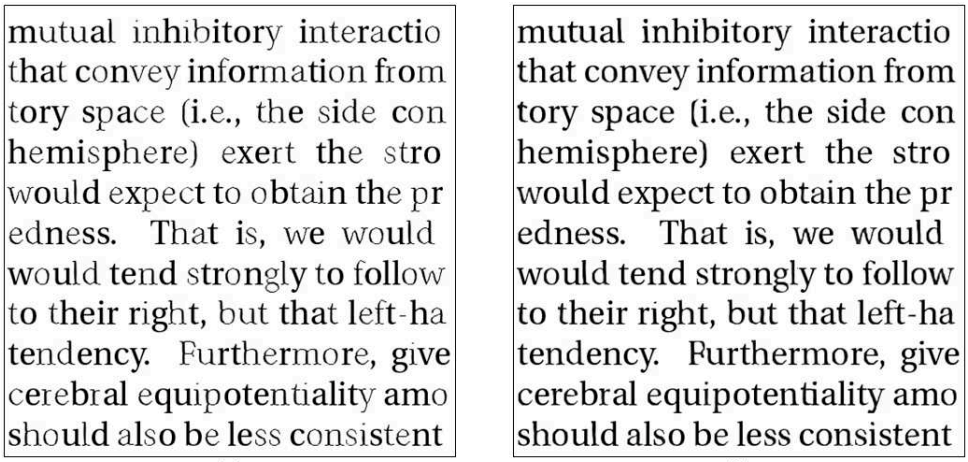


Figure 5.19: *(a) The document is watermarked by replacing its characters with Font1, and (b) with Font3.*

the hidden information in case of hiding data with Font1 whereas Table 5.16 with Font3.

Table 5.15: *The precision of extracted information (Font1)*

Distortions	Accuracy Ratio						
	Doc1	Doc2	Doc3	Doc4	Doc5	Doc6	Avg. (70 documents)
JPEG 50%	1	1	1	1	1	0.96	0.99
JPEG 30%	1	1	1	1	0.48	0.97	0.96
Rotation 3°	1	1	1	0.99	1	0.94	0.99
Rotation 5° (a)	1	1	1	1	1	0.99	0.98
Rotation 7°	1	1	1	1	1	0.99	0.99
Scaling 0.7 (b)	0.97	0.99	0.96	0.97	0.60	0.79	0.97
Scaling 0.8 (c)	0.59	0.79	0.99	0.99	0.39	0.72	0.94
Scaling 1.2 (d)	0.98	0.98	1	1	1	0.96	0.97
Scaling 1.3 (e)	1	0.97	1	1	1	0.99	0.99
(a) + (b)	0.82	0.95	0.99	0.95	0.61	0.74	0.95
(a) + (c)	1	1	0.99	0.89	1	0.59	0.94
(a) + (d)	1	1	1	1	1	0.94	0.97
(a) + (e)	1	1	1	1	1	0.95	0.98

The results in Table 5.15 and Table 5.16 show that using FCN to detect character variants substituted by Font3 gives less accuracy than the ones replaced by Font1. Substitution with Font1 gives the watermarked documents with better quality. Besides, the inconsistency of extracted bounding boxes (by OCR) between the original and distorted documents partly affects the precision of detecting the hidden information. Unlike image watermark, the textual watermark requires high accuracy ratio in order to retrieve a meaningful text from the extracted watermark bits. For example, the extracted information like "watermarkil¿4bo E" and "water-ar#i.f4s#haom" corresponds to accuracy ratio of 0.84 and 0.92. In general,

Table 5.16: *The precision of extracted information (Font3)*

Distortions	Accuracy Ratio						
	Doc1	Doc2	Doc3	Doc4	Doc5	Doc6	Avg. (70 documents)
JPEG 50%	1	1	1	1	1	0.96	0.99
JPEG 30%	1	1	1	0.99	0.88	0.97	0.95
Rotation 3°	1	0.99	1	0.99	0.99	0.99	0.98
Rotation 5° (a)	1	1	1	0.98	1	0.99	0.99
Rotation 7°	1	1	1	0.99	1	0.99	0.96
Scaling 0.7 (b)	0.91	0.83	0.91	0.88	0.66	0.73	0.93
Scaling 0.8 (c)	0.59	0.76	0.96	0.95	0.43	0.70	0.95
Scaling 1.2 (d)	0.96	0.96	0.97	0.94	0.99	0.96	0.96
Scaling 1.3 (e)	0.96	0.93	0.95	0.93	0.98	0.95	0.94
(a) + (b)	0.77	0.89	0.92	0.90	0.64	0.70	0.89
(a) + (c)	0.96	0.93	0.94	0.82	0.96	0.58	0.91
(a) + (d)	1	0.96	0.98	0.99	0.80	0.93	0.93
(a) + (e)	0.98	0.99	0.96	0.95	0.98	0.95	0.90

with the results presented in Table 5.15 and Table 5.16, we can observe that we need a tradeoff between imperceptibility and robustness when designing a waterkmarking system.

Printing, photocopying and scanning distortion: We use the same protocol than the previous experiments presented in Section 5.6. Figure 5.20 shows the results of watermark detection. PS-Font1 and PS-Font3 depict robustness against PS distortion when altering selected characters by their Font1 and Font3 respectively whereas PCS-Font1 and PCS-Font3 represent resistance to PCS distortion. Our approach can detect the watermark at the resolution of 600 dpi, and at 400 dpi from hidden documents with good quality (accuracy ratio ≥ 0.98), and even some types of document scanned at the resolution of 300 dpi and 200 dpi.

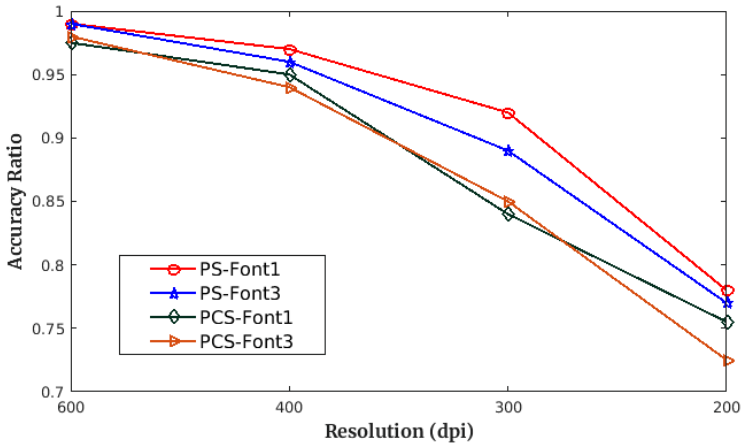


Figure 5.20: *The average results of extracted information on PS and PCS distortion.*

In general, the font generation-based watermarking scheme has been proposed by utilizing GAN for variation generation, and FCN for variation detection. The scheme can resist to

common distortions such as JPEG compression, geometric transformation, PS and PCS operations.

5.8 Comparison of scheme performance designed for grayscale documents

We would like to make comparison among our proposed schemes (STA-WM, PAT-WM, HAN-WM, GEN-WM and VAR-WM) designed for grayscale documents to see the improvement of scheme performance. The comparison is carried out based on the precision of watermark detection, which is extracted from the watermarked documents subjected to PS distortion. The result of comparison presented in Figure 5.21 shows that the methods based on generated document (GEN-WM) and character variations (VAR-WM) have significantly improved the accuracy of watermark detection, and they are able to withstand two rounds of photocopying prior to scanning back.

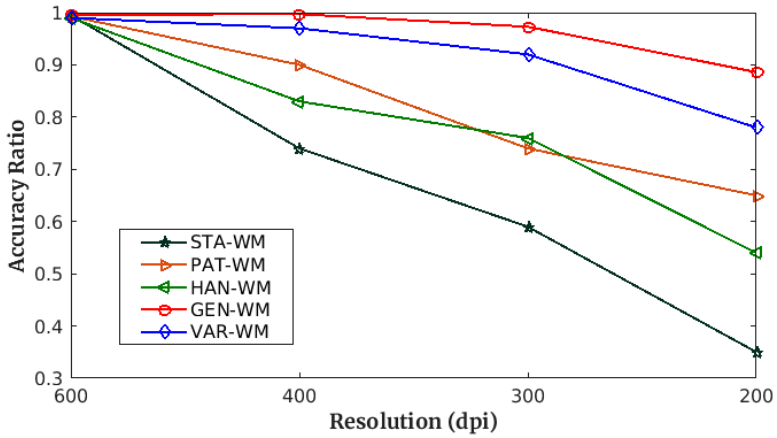


Figure 5.21: *The comparison of performance among our watermarking schemes in terms of PS distortion.*

We have observed that there are two main factors affecting the performance of the watermarking scheme: the features extracted from the documents, and the algorithm for hiding and detecting the secret data. Although the features (watermarking regions) extracted from the schemes like STA-WM, PAT-WM and HAN-WM are quite stable against distortions, they can not be capable of detecting the hidden information correctly when the watermarked documents undergo high distortions. This is because their watermarking algorithms are designed based on the changing of pixel values. By the experiments, we have observed that the pixel values of distorted watermarked documents are significantly changed. This change results in breaking predefined conditions used between the data hiding and detection process. It means that the conditions calculated from the original document (data hiding phase) are inconsistent with the conditions calculated from the distorted watermarked document (data detection phase). Thus, the scheme can not extract the hidden data correctly when the

watermarked documents are subject to high distortions. The other schemes (GAN-WM and VAR-WM) have been proposed to overcome the issue of this inconsistency. The comparison shows that the scheme based on font generation (VAR-WM) gives the highest performance because the shape of character variations are little affected by the distortions.

Besides, we also compare the performance of our watermarking schemes with other typical methods. Due to unavailability of dataset and implementation of these methods, we can not make quantitative comparison by implementing our proposed schemes on the datasets which are used in the existing approaches, and implementing the existing approaches on the public datasets which are used in our schemes. Thus, our comparison is performed relied on the functionalities of these schemes. We define various types of distortion for our comparison including: (1) geometric transformation; (2) JPEG compression; (3) PS noise; (4) PCS noise; (5) cropping, salt and pepper noise; (6) filtering noises; (7) format conversion; (8) rasterization; and (9) print-capture. The comparison as depicted in Table 5.17 shows that our approaches can be applied for various types of documents whose content can be typewritten, handwritten, hybrid or textual. In addition, the schemes (GEN-WM and VAR-WM) can resist to highly practical distortions. Thus, they can be deployed for practical applications. In general, we have proposed various schemes that are compatible with various document content, and resist to various distortions compared to the existing works.

Table 5.17: *The comparison with the existing approaches. The word "Security" refers whether a private key is used to recover the original data from the extracted data. The sign of "-" indicates that this feature is not mentioned in their work.*

Methods	Document type	Distortions	Security	Data hiding algorithm based
Horng[55]	Hybrid content	(2), (5), (6)	-	Coefficient value
Chetan[56]	Hybrid content	(1), (2), (5)	-	Coefficient value
Kim[49]	Text content	Cropping, page segmentation	-	Word shifting
Low[68]	Text content	(3), (4)	-	Line shifting
Kim[1]	Text content	Rotation, noise insertion, cropping blurring, sharpening, binarization	-	Histogram value
Zou[51]	Text content	(3), (4)	-	Inter-word space
Varna[52]	Text content	(3), (4)	-	Stroke's left edge
Palit[48]	Indian text	(2), noise and scaling	-	Character prototype
Tan[71]	Chinese text	(2), (3), (4), (5), (6)	-	Character stroke
Xiao[53;54]	Text content	(3), (7), (8), (9)	-	Character variant
STA-WM	General content	(1), (2), (3)	-	Pixel value
PAT-WM	General content	(1), (2), (3)	-	Pixel value
HAN-WM	Handwriting, typewriting	(1), (2), (3)	-	Pixel value
GEN-WM	General content	(1), (2), (3), (4)	Yes	Pixel value
VAR-WM	General content	(1), (2), (3), (4), (7)	Yes	Character variant

5.9 Watermarking for securing binary documents

The digital versions of the documents commonly suffer from various degradations. These degradations could reduce the accuracy of tasks of document analysis and recognition. Thus, transferring the documents into binarized forms is a common solution to deal with these issues. The genuine documents which exist in the binarized format for meeting the requirement of various purposes are inevitable in the real world. This section demonstrates the experimental results of the scheme presented in Section 4.6.

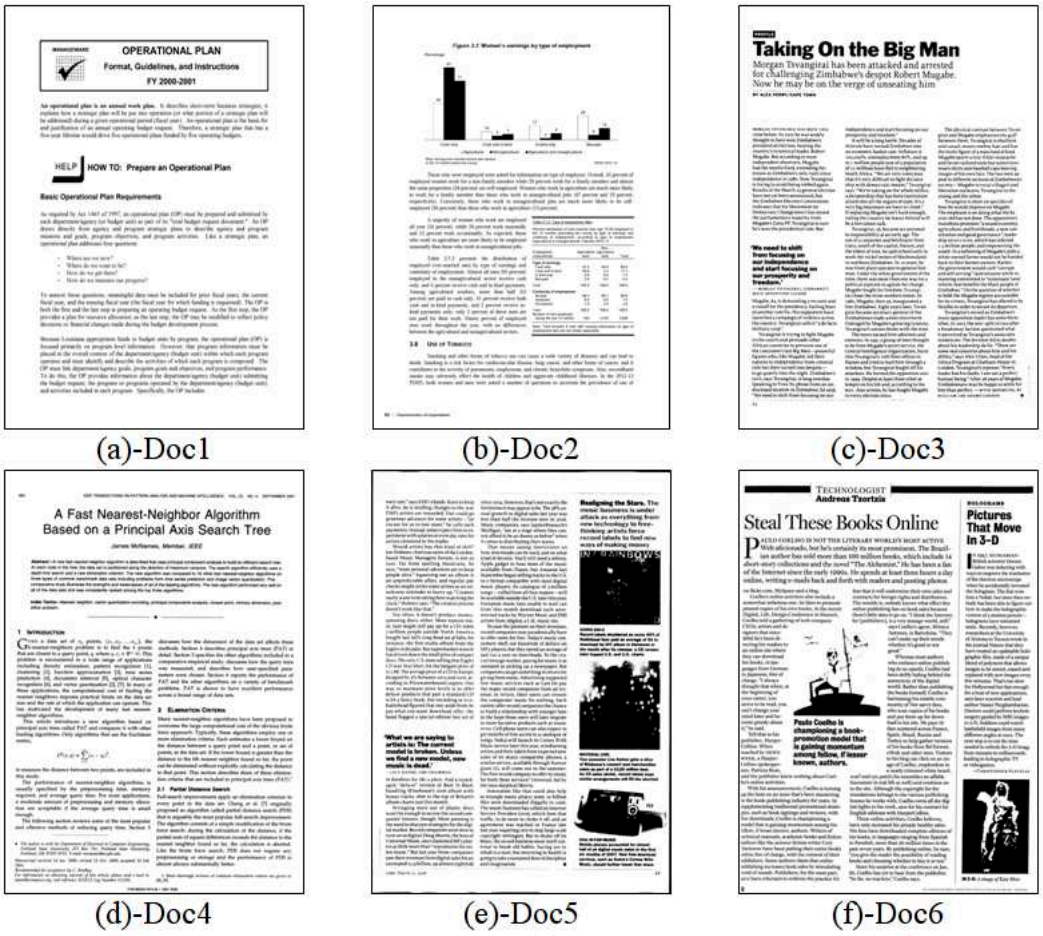


Figure 5.22: *Sample general documents with various content: (a) and (b) are documents from DSSE-200 dataset. (c)-(f) are documents from L3iDocCopies dataset.*

For training FCN network, we test our approaches on two datasets: PRImA[170] and DSSE-200[121]. A total of 254 document images with diverse and complex content layouts have been selected for making training sample. In the context of our work, we expect invariance to rotation, scale variation and variation of the quality factor of the lossy compression. Thus, we generate 1,524 from a total of 254 document images for training samples. Regarding initialization of network parameters, the number of learning steps is set to 200,000. The high momentum is assigned to 0.9. The weight decay is 5×10^{-4}. The learning rate is fixed

to 10^{-4} during the training phase. The dropout rate is assigned to 0.5.

For watermarking scheme, we select 15 documents from DSSE-200 (referred to Type-1) whereas 60 documents are selected from L3iDocCopies[120] including: 15 documents scanned from Konica Minolta Bizhub 223, and 15 documents scanned from Fujitsu fi 6800 at the resolution of 300 dpi. These documents are referred to Type-2 including: 15 documents scanned from Konica Minolta Bizhub 223, and 15 documents scanned from Fujitsu fi 6800 at the resolution of 600 dpi. This kind of documents is known as Type-3. We have tested our approach on 75 various documents. These documents are color documents, so they need to be converted into binary form before doing the experiments. The secret information hidden in the documents is $wm =$ "watermarking-information", and this text message is modulated into 192 bits.

With respect to parameters of watermark hiding scheme, the threshold to separate the number of edge features and the number of corner features within each subregion is set to $\delta = 20$ (applying for watermark hiding scheme 2). This value is experimentally selected to provide a good robustness, especially when the watermarked documents suffering from PS distortion. Related to scaling factor estimation, the value of c is set to 1,694, 1,873, 1,940, 4,055, 2,152 and 4,190 corresponding to the sample documents Doc1 - Doc6 as in Figure 5.22. These values are determined depending on the Euclidean distance between a top left point and an intersecting point of two diagonal lines of the minimum rectangle of original document. The performance of our approach is evaluated through the following factors.

Imperceptibility and capacity

With regard to the watermark hiding scheme 1, the capacity is measured by the number of 3×3 hiding patterns satisfying condition of corner features (CP_1 and CP_2) and edge features (EP_1 and EP_2). For the watermark hiding scheme 2, the capacity is equal to the number of subregions (the adjacent bounding boxes of objects are grouped together). These two watermark hiding schemes are presented in Section 4.6.4. The imperceptibility or quality of watermarked documents is measured by the difference between the document prior to hiding a secret information and the document after hiding a secret information. To assess the quality of watermarked documents, we have adopted three methods in this experiments: peak signal to noise ratio (PSNR), distance reciprocal distortion measure (DRDM)[45], and structural similarity index (SSIM)[171]. We have observed that the watermark hiding scheme 1 provides high capacity compared to the watermark hiding scheme 2.

The imperceptibility (watermarked document quality), capacity (the number of bits) and average values of 75 documents are shown in Table 5.18 (for the watermark hiding scheme 1) and Table 5.19 (for the watermark hiding scheme 2) wherein their minimum and maximum values are illustrated in color. To save the displaying space, we do not show all records of 75 documents. We just present the results of the sample documents and the ones which allow to show the minimal and maximal values of PSNR, SSIM and DRDM.

Figure 5.23 shows an example of hiding 3,837 and 10 random message bits into a small

Table 5.18: *The assessment of watermarked document quality and capacity for the watermark hiding scheme 1*

Documents	PSNR	SSIM [0, 1]	DRDM	Capacity
1 (Doc1)	22.5388	0.9747	0.8001	43,211
2 (Doc2)	22.1989	0.9742	0.9801	44,350
3 (Doc3)	21.7668	0.9618	0.8816	72,345
4 (Doc4)	24.9702	0.9789	0.7940	128,944
5 (Doc5)	21.9765	0.9694	0.8091	78,725
6 (Doc6)	24.3565	0.9736	0.9165	154,895
7	25.8832	0.9855	1.1050	12,547
...	...	...	...	...
Average (75 documents)	**23.4877**	**0.9759**	**0.8934**	**60,772**

Table 5.19: *The assessment of watermarked document quality and capacity for the watermark hiding scheme 2*

Documents	PSNR	SSIM [0, 1]	DRDM	Capacity
1 (Doc1)	21.2972	0.9680	3.7333	358
2 (Doc2)	21.2556	0.9713	4.6231	511
3 (Doc3)	21.1561	0.9672	4.1650	606
4 (Doc4)	23.3509	0.9691	4.3933	1428
5 (Doc5)	20.9203	0.9638	4.2756	756
6 (Doc6)	23.2303	0.9683	5.0502	2012
7	23.0225	0.9809	4.2822	326
8	23.9660	0.9784	4.9359	100
...	...	...	...	...
Average (75 documents)	**22.4221**	**0.9720**	**4.3363**	**664**

document with a size 1624×324 by using the watermark hiding scheme 1 and the watermark hiding scheme 2 respectively. The PSNR, SSIM and DRDM of this watermarked document are respectively 24.4671, 0.976126 and 4.0989 for the watermark hiding scheme 1, and 25.4848, 0.983157 and 3.8798 for the watermark hiding scheme 2. We can see hidden positions as illustrated by white dots in Figure 5.23(c) and Figure 5.23(e) where the pixel values are changed to hide secret data. The document quality assessment shows that the visual distortion caused by the hiding process is less perceptible. In other words, the differences between the original documents and watermarked documents in Figure 5.23 are difficult to observe by human eyes.

The results shown in Table 5.18 and Table 5.19 confirm that the proposed schemes satisfy the imperceptibility criteria. Moreover, the watermark hiding scheme 2 gives less capacity than the scheme 1 because this scheme uses a group of adjacent objects for carrying one message bit instead of a hiding pattern (as discussed in Section 4.6.2). The watermark hiding scheme 2 is designed to improve the robustness which is demonstrated in the next section.

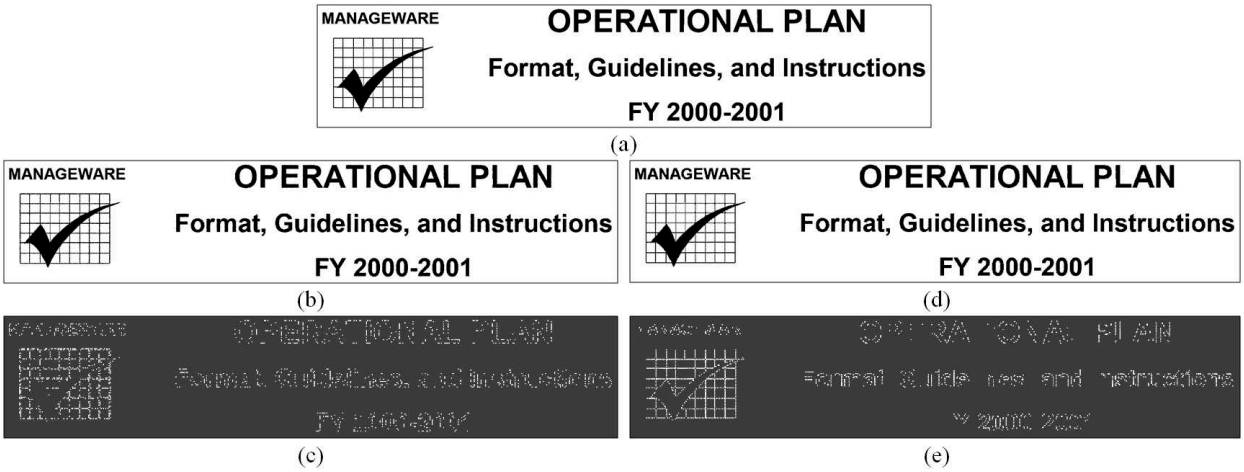


Figure 5.23: *An illustration of imperceptibility and capacity: (a) a small document with a size of* 1624×324*; The watermarked document and document difference after hiding 3,837 random bits by using watermark hiding scheme 1 are depicted in (b) and (c); For watermark hiding scheme 2, each subregion within watermarking regions carries one information bit. (e) and (f) are watermarked document and document difference after hiding 10 random bits.*

Robustness evaluation

In order to prove the robustness of our approach, we evaluate our two watermark hiding schemes with and without distortions. In case experiments without distortions, we hide the mentioned secret information (wm = "watermarking-information") into 75 documents picked up from two datasets as described above. The hidden information has been successfully extracted from the watermarked documents with the highest accuracy value of 1. To demonstrate the accuracy ratio for detecting the hidden information with distortions, we hide the same secret information into the sample documents shown in Figure 5.22. After hiding the secret information, additional distortions are added to the watermarked documents. The precision obtained for the detection of the hidden information is detailed as follows.

Robustness against JPEG compression: Figure 5.24 presents the results of hidden information detection for quality factors ranging from 10 to 100. We can see that the watermark hiding scheme 1 is capable to resist to distortions with the quality factor down to 70, and 60 for some types of document content whereas the watermark hiding scheme 2 is able to withstand the degradations when the quality factor goes down to 50. However, the accuracy is greatly reduced due to the severe degradations of watermarked documents. This reduction is due to: (i) the watermarking regions extracted from the distorted watermarked documents do not completely match with the ones extracted from the undistorted documents; (ii) the integrity of edge and corner features between the distorted and undistorted documents is lost.

Robustness against geometric distortions, salt and pepper noise, and filtering: Another category of popular distortions, a good watermarking scheme should resist, concerns geomet-

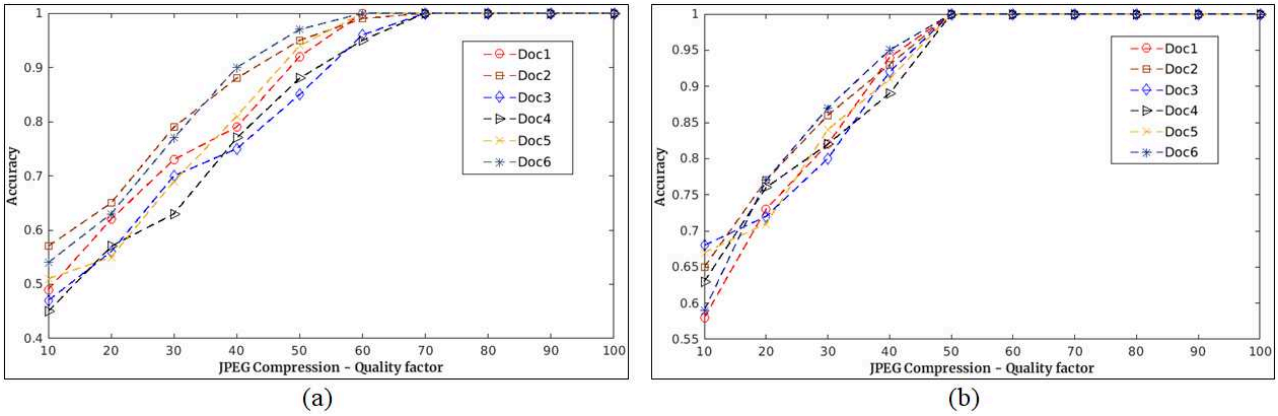


Figure 5.24: *The illustration of robustness against JPEG compression where the secret information is detected at several quality factors: (a) and (b) show the accuracy when extracting the hidden information from the watermark hiding scheme 1 and 2.*

ric transformations. These distortions consist of rotation, scaling and a combination of them. In this work, we utilize affine transformation to simulate geometric distortion as well as to correct it. As mentioned above the affine normalization is applied on the original document before hiding the secret information, and it is applied on the watermarked document prior to detecting the hidden information. Concerning the rotation, we conduct the experiments with the rotation angles of 3, 5 and 7 degrees whereas the scaling factor is taking values in the range $\left[0.8,\ 1.2\right]$ with a step of 0.1. We also test our approach for salt and pepper noise with noise density varying from 1 to 5%, media filtering and Gaussian filtering with a kernel of size 3×3. The results of robustness against geometric distortions and filtering are presented in Table 5.20 and Table 5.21.

We can see that the watermark hiding scheme 1 (each hiding pattern of edge feature or corner feature carrying one secret bit) gives low robustness as presented in Table 5.20 because these distortions cause much change on the pixel values located on the stroke of document foreground. Many edge and corner features are broken, and this leads to fail in data detection. Meanwhile, the watermark hiding scheme 2 gives better performance as depicted in Table 5.21. This scheme is designed based on the difference between the number of edge and corner features within each subregion for carrying one secret bit. Specifically, it can be able to withstand rotation angle up to 5 degrees and even 7 degrees for some types of document. In case of scaling, the scheme works well for the factors of 0.9 and 1.1 (1.2 for some types of document). In addition, the rotation of 3 degrees followed by scaling 0.9 or 1.1 is also tolerated in the watermark hiding scheme 2. We can observed that despite the geometric distortions make edge and corner features lose much their integrity, the proportion between the number of edge features and the number of corner features within a subregion or grouped contiguous bounding boxes of objetcs is still maintained at a moderate level of degradation.

Furthermore, we also test our approach under salt and pepper noise with ratio of 1, 3

Table 5.20: *Watermark hiding scheme 1: The results of accuracy when detecting the hidden information under geometric distortion, salt and pepper noise, media filtering (MF) and Gaussian filtering (GF)*

Distortions	Watermark hiding scheme 1 (Accuracy)						
	Doc1	Doc2	Doc3	Doc4	Doc5	Doc6	Avg. (75 documents)
Rotation 3° (a)	0.98	1	0.73	1	0.58	0.68	0.97
Rotation 5°	0.73	0.58	0.68	0.63	0.50	0.55	0.72
Rotation 7°	0.43	0.50	0.40	0.58	0.48	0.45	0.59
Scaling 0.8 (d)	0.55	0.58	0.48	0.60	0.38	0.53	0.56
Scaling 0.9 (b)	0.93	0.81	0.75	0.98	0.83	0.95	0.95
Scaling 1.1 (c)	1	1	1	1	0.95	1	0.97
Scaling 1.2	0.95	0.84	0.78	1	1	0.98	0.96
(a) + (b)	0.58	0.68	0.64	0.83	0.63	0.75	0.73
(a) + (c)	0.55	0.65	0.53	0.48	0.53	0.50	0.59
(a) + (d)	0.50	0.45	0.38	0.45	0.48	0.45	0.49
Salt_pepper 1%	0.90	0.85	0.98	0.88	0.83	0.95	0.96
Salt_pepper 3%	0.63	0.75	0.60	0.63	0.73	0.65	0.73
Salt_pepper 5%	0.65	0.68	0.50	0.58	0.55	0.48	0.64
MF 3×3	0.58	0.53	0.50	0.45	0.33	0.40	0.57
GF 3×3	0.83	0.55	0.43	0.58	0.70	0.43	0.72

Table 5.21: *Watermark hiding scheme 2: The results of accuracy when detecting the hidden information under geometric distortion, salt and pepper noise, media filtering (MF) and Gaussian filtering (GF)*

Distortions	Watermark hiding scheme 2 (Accuracy)						
	Doc1	Doc2	Doc3	Doc4	Doc5	Doc6	Avg. (75 documents)
Rotation 3° (a)	1	1	1	1	1	1	1
Rotation 5°	1	1	1	1	1	1	1
Rotation 7°	0.84	1	1	0.78	0.63	0.45	0.97
Scaling 0.8 (d)	0.73	0.68	0.80	0.95	0.93	0.90	0.92
Scaling 0.9 (b)	1	1	1	1	1	1	0.99
Scaling 1.1 (c)	1	1	1	1	1	1	1
Scaling 1.2	1	1	0.87	1	0.93	0.88	0.96
(a) + (b)	1	1	1	1	1	1	0.99
(a) + (c)	1	1	1	1	1	0.83	0.98
(a) + (d)	0.69	0.63	0.65	0.63	0.45	0.43	0.67
Salt_pepper 1%	1	0.98	0.98	1	0.98	1	0.99
Salt_pepper 3%	1	0.93	0.78	0.70	0.83	0.85	0.91
Salt_pepper 5%	0.65	0.84	0.60	0.63	0.74	0.53	0.72
MF 3×3	0.76	0.65	0.50	0.68	0.60	0.50	0.68
GF 3×3	0.98	0.80	0.78	0.65	0.73	0.60	0.89

and 5 percents. This type of noise converts white pixels into black ones and vice versa, and it considerably contaminates binary documents and grayscale images. It can be seen that only the watermark hiding scheme 2 is able to resist to this noise if the percentage of noise is low. Compared to other distortions, both of our watermark hiding schemes give lower performance when the watermarked documents undergo median and gaussian filtering with a kernel of size 3×3. These filtering change pixel values of stroke of document foreground a lot (smoothing the edge of document content and resulting in breaking the edge and corner features). This is the main reason leading to reduce the accuracy of hidden information detection because the hiding process for binary documents in the spatial domain is primarily based on the stroke of document foreground. To conclude, the robustness against geometric and salt-and-pepper distortions is an important property to enhance the performance of the proposed scheme against printing and scanning degradations as shown in the next section.

Robustness against print-and-scan distortion: A good watermarking system for binary document images should tolerate distortions caused by print-and-scan process. The printing and scanning process do not just geometrically distort the documents, but make the documents suffering from salt-and-pepper noise and blur. The scanned documents are more or less subjected to rotation with a certain degree. This is due to paper loaded from the paper tray of printer or placed on the scanning screen of scanner machine by human, but the rotation is quite small in general. However, the document dimension (scaling factor) is much change at different resolutions. In fact, the level of print-and-scan distortions is relied on the quality of printing and scanning machine used for.

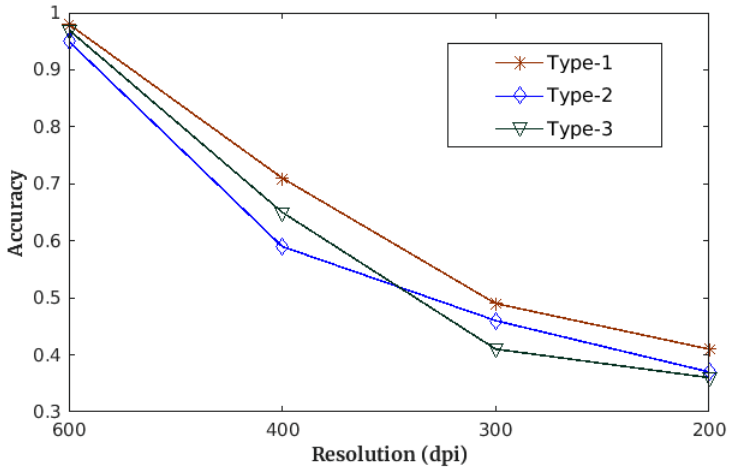


Figure 5.25: *The average results of hidden information detection for different documents in which the watermarked documents are printed at resolution of 600 dpi, and then scanned at various resolutions.*

In order to carry out experiment for the robustness against print-and-scan process, we use the watermark hiding scheme 2 for demonstrating the resistance to this kind of distortion. The printing machine named Kyocera TASKalfa 3252ci is used to print the watermarked documents. The printed documents are then scanned by using the same machine at various resolutions including 200, 300, 400 and 600 dpi. The documents printed and scanned at the same resolution of 600 dpi keep the same dimension (4960×7014). In this experiment, one

round of the printing and scanning process is deployed for watermarked documents. Then, the hidden information is extracted from these scanned documents.

The average results of accuracy when detecting the hidden information is illustrated in Figure 5.25. Our approach is capable to resist to distortions when the watermarked documents are printed and scanned at high resolution of 600 dpi. For lower resolutions, the accuracy of hidden information detection is considerably diminished. The main reason for this reduction is the integrity between the number of edge and corner features within each subregion is lost, and the other factor is that the matching of the watermarking regions between the watermarked and scanned documents can not be done. With a resolution of 600 dpi, the scanned documents obtain a good accuracy ratio during the detection of hidden information. However, some types of documents give lower performance when extracting the hidden information (failed to restore to the meaningful information), and most of these documents belong to Type-2. Below is an illustration of extracting the hidden information with the accuracy lower than 1.

- $wm =$ "fa4ermarking-informataon" corresponding to $Accurary = 0.98$.
- $wm =$ "Watarma2kifg-informati/n" corresponding to $Accurary = 0.97$.
- $wm =$ "watermaskkng-informction" corresponding to $Accurary = 0.98$.
- $wm =$ "watermarkh3formation" corresponding to $Accurary = 0.89$.
- $wm =$ "watermarkhg-i" corresponding to $Accurary = 0.74$.

Comparison with the state-of-the-art methods

In this section, we make a quantitative comparison with other data hiding methods for binary document and natural images[14–16] in spatial domain. Unfortunately, we can only access to limited resources of the methods of the literature, and therefore we can not make more quantitative comparison with the other. The comparison only focuses on whether the data hiding scheme is capable to satisfy such requirements as imperceptibility, capacity, robustness and security.

We used the binary documents made publicly available by Cao[15], consisting of two types of documents with English text (English text 1, English text 2 and English text 3 refer to document with Arial, Times New Roman and Bradley Hand ITC font style, size of 12pt) and Chinese text (Chinese text 1 and Chinese text 2 indicate document with Song Ti and Hei Ti font style, size of 12pt) as shown in Figure 5.26. These documents contain text in A4-size paper and are presented under different resolutions including 150 dpi (1240×1753), 300 dpi (2479×3507) and 600 dpi (4958×7016). The results of the comparison in regard to capacity is illustrated in Table 5.22 where IB3 and IB4 refer to interlaced block with size of 3×3 and 4×4.

As mentioned in the state-of-the-art review, the approach presented by Yang and Kot[14]

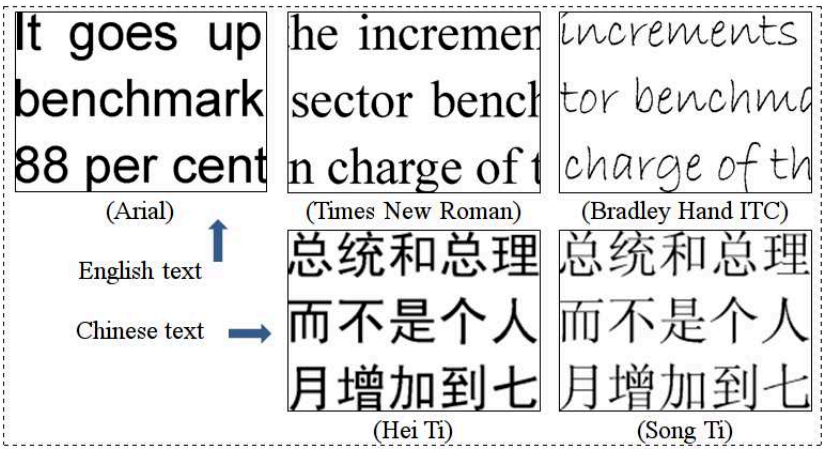


Figure 5.26: *The sample document content with English text and Chinese text used for comparison.*

Figure 5.27: *Sample binary natural images with size of* 512×512.

satisfies the essential properties such as imperceptibility, capacity and security whereas the scheme presented by Cao and Kot [15] is designed to meet the requirements of imperceptibility and capacity. In comparison with these approaches, our watermark hiding scheme 1 (WM1) gives better capacity, except for English 3 at resolution of 300 and 600 dpi. Regarding the watermark hiding scheme 2 (WM2), the capacity is lower than other approaches, but our scheme has proved to be robust against the distortions, which has not been demonstrated in the other existing works, especially against print-and-scan distortion. We can not propose the quantitative comparison on the imperceptibility property because the implementation for measuring the visual quality used in their work are not publicly available.

Moreover, we make a quantitative comparison with the recent data hiding approach proposed by Nguyen *et al.* [16] in which we test this algorithm on our sample documents depicted in Figure 5.22 as well as our watermark hiding scheme 1 on the sample natural images (standard test images) as shown in Figure 5.27. There are three thresholds which affect the quality of hidden images and capacity in their scheme [16], and we use the maximum threshold $Th = 6$ (maximum hiding capacity) for this experiment. The higher the threshold is, the more the capacity is obtained. However, the quality of images after hiding secret bits will be low. The results of comparison on binary documents and natural images are shown in

Table 5.22: *Comparison of our watermark hiding schemes and the schemes presented in*[14;15] *in terms of capacity (bits)*

Documents	Resolution	Scheme presented in[14]		Scheme presented in[15]	Our method	
		IB3	IB4		WM1	WM2
English 1	150	5,245	6,357	9,442	10,175	889
	300	15,873	17,701	30,391	38,528	1,485
	600	29,548	31,735	48,017	61,989	1,732
English 2	150	5,721	6,392	6,693	8,213	671
	300	16,083	17,515	27,789	33,372	1,682
	600	33,640	36,285	63,871	76,535	1,746
English 3	150	4,018	4,603	4,115	4,563	473
	300	18,577	19,481	24,850	23,646	1,693
	600	44,296	46,884	83,231	63,173	1,868
Chinese 1	150	5,282	6,013	6,252	10,629	906
	300	15,822	17,513	25,650	30,959	1,588
	600	34,640	38,513	66,077	73,098	2,273
Chinese 2	150	5,220	5,917	7,629	12,650	981
	300	13,166	14,525	24,200	30,837	1,629
	600	28,276	30,914	55,118	66,337	2,257

Table 5.23: *Comparison of our approach (WM1) and Nguyen's method*[16] *on capacity and imperceptibility for natural images as shown in Figure 5.27*

Images	Scheme presented by Nguyen[16]			Our method (WM1)		
	Capacity	PSNR	SSIM	Capacity	PSNR	SSIM
Lena	**86,613**	10.86	0.6787	1,565	**29.86**	**0.9921**
Baboon	**86,280**	9.91	0.7234	3,693	**29.79**	**0.9966**
Peppers	**86,634**	10.96	0.6866	1,231	**29.53**	**0.9915**
Airplane	**86,598**	10.86	0.6387	1,612	**29.44**	**0.9909**

Table 5.23 and Table 5.24.

Table 5.24: *Comparison of our approach (WM1) and Nguyen's method*[16] *on capacity and imperceptibility for binary documents depicted in Figure 5.22*

Documents	Scheme presented by Nguyen[16]			Our method (WM1)		
	Capacity	PSNR	SSIM	Capacity	PSNR	SSIM
Doc1	35,514	**25.60**	**0.9826**	**43,211**	22.54	0.9747
Doc2	5,130	**34.76**	**0.9840**	**44,350**	22.20	0.9742
Doc3	**109,212**	**23.45**	0.9568	72,345	21.77	**0.9618**
Doc4	3,294	**35.97**	**0.9864**	**128,944**	24.97	0.9789
Doc5	**120,183**	20.18	0.9424	78,725	**21.98**	**0.9694**
Doc6	120,228	24.31	0.9418	**154,895**	**24.36**	**0.9736**

The Table 5.23 and Table 5.24 show that our watermark hiding scheme 1 gives better performance for the documents whose the stroke part of the objects contains many of edge and corner features. Meanwhile, for natural images, the data hiding method presented by Nguyen *et al.*[16] gives higher capacity than our WM1, but instead its imperceptibility is quite low (as depicted in Table 5.23). The main factor allowing the Nguyen's approach[16] to provide high capacity on natural images is that the complex blocks for data hiding found on this type of image are much more numerous than on the document images. In general, based on the scheme's functionalities, these data hiding methods[14–16] are appropriately classified into steganography applications because their approaches mainly focus on the capacity, and security (as found in[14]) whereas the robustness of the scheme is not mentioned.

To provide an overview of the performance of our approach compared to the existing data hiding methods, apart from previous quantitative comparisons, we compare our approach with other typical watermarking and data hiding schemes for binary images. This comparison basically concentrates on: (*i*) whether the watermarking scheme can be applied on binary documents with hybrid content; (*ii*) the scheme is capable to satisfy the crucial properties such as robustness, which is very challanging for binary documents, and security; (*iii*) which techique is used to detect document features in order to develop the data hiding or watermarking scheme; and (*iv*) whether the security features such as the encryption of secret message, the shuffling of blocks used for carrying data, etc. are integrated into the data hiding schemes. The comparison are depicted in Table 5.25. The comparison shows that our approach obtains competitive performance compared to the existing approaches. The proposed approach is based on pattern recognition technique unlike other methods of the literature, and it provides robustness and security enhancement.

Table 5.25: *Comparison with typical watermarking and data hiding approaches*

Method	General content	PS distortion	Security	Technique
Kim[1]	No	No	No	Edge direction histogram (each word)
Zou[51]	No	Yes	No	Inter-word spaces (spaces in each row)
Palit[48]	No	No	No	Pattern matching (character prototype)
Lina[71]	No	Yes	No	Stroke direction modulation (each character)
Lee[96]	Yes	No	Yes	Edge line segment similarity measure
Yang[14]	Yes	No	Yes	Pixel transition and uneven embeddability
Cao[15]	Yes	No	No	Edge adaptive grid and contour tracing
Wang[98]	Yes	No	Yes	Block pattern (block-by-block)
Daraee[101]	Yes	Yes	No	Fractal codes (block-by-block)
Hou[102]	Yes	Yes	No	Sampling operation (image thumbnail)
Son[16]	Yes	No	Yes	Block classification (block-by-block)
Our method	Yes	Yes	Yes	Fully convolutional networks, corner and edge features

5.10 Summary

In this chapter, we have presented the experimental results of our proposed schemes designed for securing (1) grayscale documents and (2) binary documents. In the former, we have proposed: a feature points-based steganography scheme; a stable regions and object fill-based watermarking scheme (STA-WM); a watermarking regions and hiding pattern-based watermarking scheme (PAT-WM); a watermarking regions and connected object-based watermarking scheme (HAN-WM) which can be applied for both typewriting and handwriting documents; a generated document-based watermarking scheme (GEN-WM); and a character variations-based watermarking scheme (VAR-WM). In the later, we have implemented a watermarking scheme based on the watermarking regions, and the hiding patterns describing the edge and corner features of document content. To develop these approaches, we have used various pattern recognition techniques to detect document features such as: SURF detector; a new detector based on contourlet transform and distance transform; a combination of common image processing operations and non-subsampled contourlet transform; fully convolutional networks; and generative adversarial networks. The steganography and watermarking algorithms are based on the level of pixel intensities, or the shape of document characters and symbols.

The experimental results show that the approaches of generating an intermediate document from the input one, which is used as a reference for scheme development, and generating variations of document characters give high performance. Specifically, the watermark is properly detected from the watermarked documents subjected to: printing at resolution of 600 dpi and scanning at resolutions of 600, 400, 300 and 200 dpi; two rounds of photocopying prior to scanning at various resolutions. However, the scheme for binary documents is only able to detect the watermark from the watermarked documents when printing and scanning at a resolution of 600 dpi.

Chapter 6

Conclusion and future work

This thesis presents various approaches using pattern recognition techniques to develop data hiding system for securing administrative and bussiness documents. Both analytic study and experimental results are detailed in this manuscript. We have presented several strategies, which have been designed to protect legal documents or images including data hiding, barcode, quick response code, document signature, fraud detection, photo signature, etc. Due to the wide range of applications of data hiding technique and in the context of document images, this technique can be applied for various applications, consisting of ownership protection, alteration detection, access or copy control, annotation, covert communication and protection of trained neural network. This is why we have decided to choose digital steganography and watermarking as an efficient solution for document protection. Therefore, one steganography and six watermarking schemes have been developed to integrate security feature within various types of hybrid documents (grayscale and binary document, and handwritten document) for the purpose of securing documents. To implement these schemes, different aspects have been considered as follows.

1. The consistency of features extracted from the original document and watermarked or stego document for constructing data hiding scheme.

2. Leveraging conventional pattern recognition techniques to construct hiding regions for steganography, and to extract the stable regions for watermarking documents.

3. Taking advantage of deep learning to enhance stable watermarking regions, generate a quality document from the input document (in order to enhance the detection of document content and the performance of watermarking algorithm), and to produce new fonts or variations of document characters for improvement of scheme robustness.

4. Correction of geometric distortions caused by the process of printing and scanning, and print-photocopy-scan for enhancing the precision of hidden information detection.

5. Developing watermarking algorithms based on: (1) level of pixel intensities such as group of contiguous pixel values, connected objects, ratio between the number of edge

and the number of corner features, absolute distance of pixel values between the generated and watermarked document; and (2) variations of document characters (shape of characters). In addition, pseudo random number generator is also applied to enhance the security feature of watermarking scheme.

6.1 The stability of feature points for steganography scheme

The extracted feature points by using SURF detector are sorted based on their response values. The sorted feature points are then used to construct a $B \times B$ hiding regions. The pixel values of selected feature points are slightly changed for carrying secret bits. We make use of LBP for detecting corner features and LTP for determining positions where their pixel values are adjusted to hide information. The Hough transform is utilized to estimate rotation angle for document correction. The information hiding algorithm is developed based on an odd-even feature of a pixel value. In addition, we have proposed a new feature point detector based on contourlet transform and distance transform with the aim of improving the stability of feature points against usual distortions. The extracted feature points and steganography algorithm are detailed in Section 3.2. The results show that this approach gives good quality of stego-documents but low robustness against distortions.

6.2 Stable regions and group of pixel values for watermarking documents

Developing steganography scheme for documents in spatial domain using feature points gives low resistance to distortions. Instead, we have developed other robust watermarking schemes by extracting stable regions from the documents, and developing watermarking algorithms based on a group of pixel values. Specifically, as the scheme presented in Section 3.3, we detect the stable regions by making use of a combination of common image processing operations and non-subsampled contourlet transform. The watermarking algorithm is based on a group of m successive pixel values situated within the filling part of document objects. This approach allows to obtain the watermarked documents whose content is minimally distorted in terms of normal observation. It is robust to printing and scanning at high resolution.

Another approach presented in Section 4.2 detects watermarking regions using fully convolutional networks. We construct a hiding pattern of size $m \times n$ to hide one information bit. The watermarking algorithm is developed based on the mean values corresponding to two divided group of pixel values within each hiding pattern. In addition, the watermarking scheme for handwritten document presented in Section 4.3 can also be applied for typewriting documents. The watermarking algorithm depends on sum values corresponding to

two divided group of pixel values in each connected object extracted from the document. Correcting geometric distortion of these three schemes are performed by determining the minimum rectangle surrounding entire document. The results show that the regions extracted using FCN are more stable than the approach using conventional techniques. The FCN-based watermarking schemes also give good imperceptibility and enable to be robust to printing and scaning at high resolution.

The proposed strategy based on the extraction of stable regions and the use of a group of pixel values has been significantly improved the robustness of the scheme in comparison with the approach based on feature points and odd-even feature of a pixel value. However, the accuracy of the hidden information detection process is still low when printing and scanning the watermarked documents at middle and low resolutions.

6.3 The generation of referenced document and variations of characters

The robust watermarking scheme presented in Section 4.4 generates an intermediate document from the input document by using generative adversarial networks. The generated document is then used as a reference to enhance the ability of detecting document content, and it is also utilized to measure the absolute distance of pixel values between each of bounding boxes extracted from the watermarked document and its corresponding bounding box from the referenced document. Another approach based on character variants has been proposed in Section 4.5 wherein the new font generation is conducted by making use of GAN. Then a FCN network is utilized to detect the hidden information. These approaches have significantly improved the precision of hidden information detection compared to the previous approaches. Specifically, the watermark can be detected when the watermarked documents are scanned at low resolutions, and even at low resolution for some types of document, and then the watermarked documents suffer two rounds of photocopying prior to scanning at various resolutions. Both of these schemes are also designed for security enhancement by using pseudo random numbers.

6.4 The corner and edge features for watermarking binary documents

Detecting salient maps, which are used as a guidance for identifying appropriate content regions for watermarking, is conducted by using FCN. The hiding pattern of size 3×3 has been constructed to detect the corner and edge features of document. They are applied for carrying watermark bits. There are two watermarking algorithms developed in this scheme. The first one hides one secret bit into each hiding pattern describing the edge and corner features. The second one hides one secret bit into a group of adjacent objects. The

former algorithm hides data directly into the center pixel of a hiding pattern whereas the later algorithm hides data by adjusting the ratio between the number of corner features and the number of edge features within each of grouped objects. In addition, the pseudo random number generator has been adopted to encode and decode the secret information for enhancement of security feature. This scheme is able to withstand distortions caused by printing and scanning at high resolution.

6.5 Future works

This thesis has proposed a steganography and several watermarking schemes for various documents like grayscale and binary documents, and handwriting documents to improve the effectiveness for securing legal documents. However, there are some opportunities to further improve these approaches, and also to deal with other relevant issues when using pattern recognition techniques. One of these issues is to detect the hidden information from captured documents, which has not been covered in the thesis. Depending on the discussions carried out during the current research, some of these issues are detailed as follows.

- More experiments are needed to evaluate all the proposed methods: (*i*) Testing the proposed schemes on a bigger testing dataset; (*ii*) For print-photocopying-scan resistance, the schemes need to be tested on differents machine with different resolutions; (*iii*) Applying various error correction codes for correcting the improperly extracted message bits in order to improve the scheme robustness.

- For convenience usage in real applications, it is expected that the watermarking system is capable of verifying the legal documents by using a mobile device. Thus, the robustness against these distortions should be integrated in the watermarking system. Although the approaches based on the generated document and the character variations using GAN and FCN have significantly improved the robustness, they still have not been able to detect the hidden information from the watermarked documents which are subjected to light contrast due to capturing from camera. To improve the accuracy of the information detection, a specific learning process combining supervised and unsupervised approaches could be applied to the networks such as FCN, CNN-autoencoder, etc. With this strategy, the network could learn better features from the variations of document characters, and it could obtain better results when detecting the character variants in the watermarked documents even the documents undergone distortions.

- The documents captured from mobile devices or camera often suffer from perspective distortions and background noise. This kind of distortions could much affect the detection of document content, so the performance of a watermarking scheme could be considerably diminished. The proposed schemes have to provide some strategies to correct the perspective distortions and to eliminate the background noise. There are possible approaches to cope with this problem, consisting of: training a deep neural network, or using Apple's rectangle detection SDK, or Hough transform.

Appendix A

Publications

A.1 Journal papers

- Cu Vinh Loc, Jean-Christophe Burie and Jean-Marc Ogier. "A robust watermarking approach for security issue of binary documents using fully convolutional networks". International Journal on Document Analysis and Recognition (IJDAR), 2019 (under review).
- Cu Vinh Loc, Jean-Christophe Burie and Jean-Marc Ogier. "Content enhancement and font generation-based data hiding approach for securing genuine documents". Pattern Recognition Journal, 2019 (under review).

A.2 Workshop and conference papers

- Cu Vinh Loc, Jean-Christophe Burie and Jean-Marc Ogier. "A spatial domain steganography for grayscale documents using pattern recognition techniques". International Workshop on Computational Document Forensics, International Conference on Document Analysis and Recognition (IWCDF@ICDAR), 2017.
- Cu Vinh Loc, Jean-Christophe Burie and Jean-Marc Ogier. "Stable regions and object fill-based approach for document images watermarking". International Workshop on Document Analysis Systems (DAS), 2018.
- Cu Vinh Loc, Jean-Christophe Burie and Jean-Marc Ogier. "Document images watermarking for security issue using fully convolutional networks". International Conference on Pattern Recognition (ICPR), 2018.
- Cu Vinh Loc, Jean-Christophe Burie and Jean-Marc Ogier. "Watermarking for security issue of handwritten documents with fully convolutional networks". International

Conference on Frontiers in Handwriting Recognition (ICFHR), 2018.

– Cu Vinh Loc, Jean-Christophe Burie, Jean-Marc Ogier and Cheng-Lin Liu. “A robust data hiding scheme using generated content for securing genuine documents”. International Conference on Document Analysis and Recognition (ICDAR), 2019.

– Cu Vinh Loc, Jean-Christophe Burie, Jean-Marc Ogier and Cheng-Lin Liu. “Hiding security feature into text content for securing documents using generated font”. International Conference on Document Analysis and Recognition (ICDAR), 2019.

References

[1] Young-Won Kim and Il-Seok Oh. Watermarking text document images using edge direction histograms. *Pattern Recognition Letters*, 25(11):1243 – 1251, 2004. ISSN 0167-8655.

[2] L. Nataraj, A. Sarkar, and B. S. Manjunath. Precise localization of key-points to identify local regions for robust data hiding. In *2010 IEEE International Conference on Image Processing*, pages 3681–3684, 2010.

[3] M. S. Yasein and P. Agathoklis. An image normalization technique based on geometric properties of image feature points. In *2007 IEEE International Symposium on Signal Processing and Information Technology*, pages 116–121, 2007.

[4] G. Zhu, Y. Zheng, D. Doermann, and S. Jaeger. Signature detection and matching for document image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):2015–2031, 2009. ISSN 0162-8828.

[5] K. Matsuda, W. Ohyama, T. Wakabayashi, and F. Kimura. Effective random-impostor training for combined segmentation signature verification. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 489–494, 2016.

[6] Chandranath Adak, Bidyut Chaudhuri, and Michael Blumenstein. Writer identification and verification from intra-variable individual handwriting. *IEEE Access*, PP, 08 2017.

[7] K. Zhang, W. Zuo, S. Gu, and L. Zhang. Learning deep cnn denoiser prior for image restoration. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2808–2817, 2017.

[8] Jaeyoung Yoo, Sang-ho Lee, and Nojun Kwak. Image restoration by estimating frequency distribution of local patches. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6684–6692, 2018.

[9] Gabriel Eilertsen, Joel Kronander, Gyorgy Denes, Rafał K. Mantiuk, and Jonas Unger. Hdr image reconstruction from a single exposure using deep cnns. *ACM Transactions on Graphics (TOG)*, 36(6), 2017.

[10] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Learning a single convolutional super-resolution network for multiple degradations. *CoRR*, abs/1712.06116, 2017.

[11] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, 2017.

[12] Iuon-Chang Lin, Yang-Bin Lin, and Chung-Ming Wang. Hiding data in spatial domain images with distortion tolerance. *Comput. Stand. Interfaces*, 31(2):458–464, 2009. ISSN 0920-5489.

[13] Seyyed Hossein Soleymani and Amir Hossein Taherinia. High capacity image steganography on sparse message of scanned document image (smsdi). *Multimedia Tools Appl.*, 76(20):20847–20867, 2017. ISSN 1380-7501.

[14] H. Yang and A. C. Kot. Pattern-based data hiding for binary image authentication by connectivity-preserving. *IEEE Transactions on Multimedia*, 9(3):475–486, 2007. ISSN 1520-9210.

[15] H. Cao and A. C. Kot. On establishing edge adaptive grid for bilevel image data hiding. *IEEE Transactions on Information Forensics and Security*, 8(9):1508–1518, 2013. ISSN 1556-6013.

[16] Thai-Son Nguyen, Chin-Chen Chang, and Huan-Sheng Hsueh. High capacity data hiding for binary image based on block classification. *Multimedia Tools and Applications*, 75(14):8513–8526, 2016. ISSN 1573-7721.

[17] L. O'Gorman and I. Rabinovich. Photo-image authentication by pattern recognition and cryptography. In *Proceedings of 13th International Conference on Pattern Recognition*, volume 3, pages 949–953 vol.3, 1996.

[18] L. O'Gorman and I. Rabinovich. Secure identification documents via pattern recognition and public-key cryptography. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10):1097–1102, 1998. ISSN 0162-8828.

[19] S. Prabhakar, S. Pankanti, and A. K. Jain. Biometric recognition: security and privacy concerns. *IEEE Security Privacy*, 1(2):33–42, 2003. ISSN 1540-7993.

[20] N. K. Ratha, S. Chikkerur, J. H. Connell, and R. M. Bolle. Generating cancelable fingerprint templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):561–572, 2007. ISSN 0162-8828.

[21] Puchong Subpratatsavee and Pramote Kuacharoen. An implementation of a paper based authentication using hc2d barcode and digital signature. In *Computer Information Systems and Industrial Management*, pages 592–601. Springer Berlin Heidelberg, 2014.

[22] J. Z. Gao, L. Prakash, and R. Jagatesan. Understanding 2d-barcode technology and applications in m-commerce - design and implementation of a 2d barcode processing solution. In *Proceedings of the 31st Annual International Computer Software and Applications Conference - Volume 02*, volume 2, pages 49–56, 2007.

[23] T. Mantoro, M. I. Wahyudi, M. A. Ayu, and W. Usino. Real-time printed document authentication using watermarked qr code. In *International Conference on Cyber Security, Cyber Warfare, and Digital Forensic (CyberSec)*, pages 68–72, 2015.

[24] Espejel-Trujillo A., Castillo-Camacho I., Nakano-Miyatake M., and Perez-Meana H. Identity document authentication based on vss and qr codes. *Procedia Technology*, 3: 241 – 250, 2012. ISSN 2212-0173.

[25] Sébastien Eskenazi, Petra Gomez-Krämer, and Jean-Marc Ogier. When document security brings new challenges to document analysis. In *Computational Forensics*, pages 104–116. Springer International Publishing, 2015.

[26] Sébastien Eskenazi, Petra Gomez-Krämer, and Jean-Marc Ogier. Let's be done with thresholds! In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 851–855, 2015.

[27] X. Pan and S. Lyu. Region duplication detection using image feature matching. *IEEE Transactions on Information Forensics and Security*, 5(4):857–867, 2010. ISSN 1556-6013.

[28] Liyang Yu, Qi Han, and Xiamu Niu. Feature point-based copy-move forgery detection: covering the non-textured areas. *Multimedia Tools and Applications*, 75(2):1159–1176, 2016. ISSN 1573-7721.

[29] Xiang-Yang Wang, Shuo Li, Yu-Nan Liu, Ying Niu, Hong-Ying Yang, and Zhi-li Zhou. A new keypoint-based copy-move forgery detection for small smooth regions. *Multimedia Tools and Applications*, 76(22):23353–23382, 2017. ISSN 1573-7721.

[30] Anil Dada Warbhe, R.V. Dharaskar, and V.M. Thakare. A scaling robust copy-paste tampering detection for digital image forensics. *Procedia Computer Science*, 79:458 – 465, 2016. ISSN 1877-0509.

[31] Chun-Su Park and Joon Yeon Choeh. Fast and robust copy-move forgery detection based on scale-space representation. *Multimedia Tools and Applications*, 77(13):16795–16811, 2018. ISSN 1573-7721.

[32] Choudhary Shyam Prakash, Avinash Kumar, Sushila Maheshkar, and Vikas Maheshkar. An integrated method of copy-move and splicing for image forgery detection. *Multimedia Tools and Applications*, 77(20):26939–26963, 2018. ISSN 1573-7721.

[33] Cédric Maigrot, Ewa Kijak, Ronan Sicre, and Vincent Claveau. Tampering detection and localization in images from social networks: A cbir approach. In *Image Analysis and Processing - ICIAP 2017*, 2017.

[34] Yusuke Uchida, Yuki Nagai, Shigeyuki Sakazawa, and Shin'ichi Satoh. Embedding watermarks into deep neural networks. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, pages 269–277, 2017. ISBN 978-1-4503-4701-3.

[35] Yuki Nagai, Yusuke Uchida, Shigeyuki Sakazawa, and Shin'ichi Satoh. Digital watermarking for deep neural networks. *International Journal of Multimedia Information Retrieval*, 7(1):3–16, 2018. ISSN 2192-662X.

[36] Chun-Shien Lu. *Multimedia Security: Steganography and Digital Watermarking Techniques for Protection of Intellectual Property*. IGI Global, Hershey, PA, USA, 2004. ISBN 1591401925.

[37] Frank Shih. *Digital Watermarking and Steganography: Fundamentals and Techniques, Second Edition*. CRC Press, Inc., Boca Raton, FL, USA, 2, illustrated edition, 2017. ISBN 149873877X, 9781498738774.

[38] M. Utku Celik, G. Sharma, E. Saber, and A. Murat Tekalp. Hierarchical watermarking for secure image authentication with localization. *IEEE Transactions on Image Processing*, 11(6):585–595, June 2002. ISSN 1057-7149.

[39] Ping Wah Wong. A public key watermark for image verification and authentication. In *International Conference on Image Processing*, volume 1, pages 455–459 vol.1, 1998.

[40] N. Nikolaidis and I. Pitas. Robust image watermarking in the spatial domain. *Signal Process.*, 66(3), 1998. ISSN 0165-1684.

[41] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing*, 6(12):1673–1687, Dec 1997. ISSN 1057-7149.

[42] Chi-Kwong Chan and L.M. Cheng. Hiding data in images by simple lsb substitution. *Pattern Recognition*, 37(3):469 – 474, 2004. ISSN 0031-3203.

[43] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for images, audio and video. In *Proceedings of 3rd IEEE International Conference on Image Processing*, volume 3, pages 243–246 vol.3, 1996.

[44] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. ISSN 1057-7149.

[45] J. Cheng and A. C. Kot. Objective distortion measure for binary text image based on edge line segment similarity. *IEEE Transactions on Image Processing*, 16(6):1691–1695, 2007. ISSN 1057-7149.

[46] J. Cheng and A. C. Kot. Objective distortion measure for binary text image based on edge line segment similarity. *IEEE Transactions on Image Processing*, 16(6):1691–1695, 2007. ISSN 1057-7149.

[47] Min Wu and Bede Liu. Data hiding in binary image for authentication and annotation. *IEEE Transactions on Multimedia*, 6(4):528–538, 2004. ISSN 1520-9210.

[48] S. Palit and U. Garain. A novel technique for the watermarking of symbolically compressed documents. In *International Conference on Document Image Analysis for Libraries (DIAL'06)*, pages 6 pp.–296, 2006.

[49] Young-Won Kim, Kyung-Ae Moon, and Il-Seok Oh. A text watermarking algorithm based on word classification and inter-word space statistics. In *International Conference on Document Analysis and Recognition, 2003. Proceedings.*, pages 775–779, 2003.

[50] Shiyan Hu. Document image watermarking based on weight-invariant partition using support vector machine. In Simone Marinai and Andreas R. Dengel, editors, *Document Analysis Systems VI*, pages 546–554, 2004.

[51] Dekun Zou and Y. Q. Shi. Formatted text document data hiding robust to printing, copying and scanning. In *IEEE International Symposium on Circuits and Systems*, pages 4971–4974 Vol. 5, 2005.

[52] A. L. Varna, S. Rane, and A. Vetro. Data hiding in hard-copy text documents robust to print, scan and photocopy operations. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1397–1400, 2009.

[53] Chang Xiao, Cheng Zhang, and Changxi Zheng. Fontcode: Embedding information in text documents using glyph perturbation. *ACM Trans. Graph.*, 37(2):15:1–15:16, 2018. ISSN 0730-0301.

[54] Chang Xiao, Cheng Zhang, and Changxi Zheng. Fontcode: Embedding information in text documents using glyph perturbation. *CoRR*, abs/1707.09418, 2017.

[55] Shi-Jinn Horng, Didi Rosiyadi, Pingzhi Fan, Xian Wang, and Muhammad Khurram Khan. An adaptive watermarking scheme for e-government document images. *Multimedia Tools and Applications*, 72(3):3085–3103, 2014. ISSN 1573-7721.

[56] K.R. Chetan and S. Nirmala. An efficient and secure robust watermarking scheme for document images using integer wavelets and block coding of binary watermarks. *J. Inf. Secur. Appl.*, 24(C):13–24, 2015. ISSN 2214-2126.

[57] Ali Benoraira, Khier Benmahammed, and Noureddine Boucenna. Blind image watermarking technique based on differential embedding in dwt and dct domains. *EURASIP Journal on Advances in Signal Processing*, 2015(1):55, 2015. ISSN 1687-6180.

[58] S. Hamid Amiri and Mansour Jamzad. Robust watermarking against print and scan attack through efficient modeling algorithm. *Image Commun.*, 29(10):1181–1196, 2014. ISSN 0923-5965.

[59] Y. Zolotavkin and M. Juhola. A new two-dimensional quantization method for digital image watermarking. In *2015 17th International Conference on Advanced Communication Technology (ICACT)*, pages 155–160, 2015.

[60] Summuyya Munib and Asifullah Khan. Robust image watermarking technique using triangular regions and zernike moments for quantization based embedding. *Multimedia Tools and Applications*, 76(6):8695–8710, 2017. ISSN 1573-7721.

[61] Maedeh Jamali, Shima Rafiee, S. M. Reza Soroushmehr, Nader Karimi, Shahram Shirani, Kayvan Najarian, and Shadrokh Samavi. Adaptive blind image watermarking using fuzzy inference system based on human visual perception. *CoRR*, abs/1709.06536, 2017.

[62] K. Haribabu, G. R. K. S. Subrahmanyam, and D. Mishra. A robust digital image watermarking technique using auto encoder based convolutional neural networks. In *2015 IEEE Workshop on Computational Intelligence: Theories, Applications and Future Directions (WCI)*, pages 1–6, 2015.

[63] Seung-Min Mun, Seung-Hun Nam, Han-Ul Jang, Dongkyu Kim, and Heung-Kyu Lee. A robust blind watermarking using convolutional neural network. *ArXiv*, abs/1704.03248, 2017.

[64] S. Leutenegger, M. Chli, and R. Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *2011 International Conference on Computer Vision*, pages 2548–2555, 2011.

[65] A. L. Da Cunha, J. Zhou, and M. N. Do. The nonsubsampled contourlet transform: Theory, design, and applications. *IEEE Transactions on Image Processing*, 15(10): 3089–3101, 2006. ISSN 1057-7149.

[66] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39 (4):640–651, 2017. ISSN 0162-8828.

[67] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, pages 2672–2680. 2014.

[68] S. H. Low, N. F. Maxemchuk, J. T. Brassil, and L. O'Gorman. Document marking and identification using both line and word shifting. In *Proceedings of INFOCOM'95*, volume 2, pages 853–860 vol.2, 1995.

[69] T. Amano and D. Misaki. A feature calibration method for watermarking of document images. In *Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR '99 (Cat. No.PR00318)*, pages 91–94, 1999.

[70] C. Culnane, H. Treharne, and A. T. S. Ho. Improving mutli-set formatted binary text watermarking using continuous line embedding. In *International Conference on Innovative Computing, Information and Control (ICICIC 2007)*, pages 287–287, 2007.

[71] Lina Tan, Xingming Sun, and Guang Sun. Print-scan resilient text image watermarking based on stroke direction modulation for chinese document authentication. *Radioengineering*, 21:170–181, 2012.

[72] Neill D. F. Campbell and Jan Kautz. Learning a manifold of fonts. *ACM Trans. Graph.*, 33(4):91:1–91:11, 2014. ISSN 0730-0301.

[73] Athanasios Nikolaidis. Local distortion resistant image watermarking relying on salient feature extraction. *EURASIP Journal on Advances in Signal Processing*, (1):97, 2012. ISSN 1687-6180.

[74] M. Cedillo-Hernandez, F. Garcia-Ugalde, M. Nakano-Miyatake, and H. Perez-Meana. Robust digital image watermarking using interest points and dft domain. In *2012 35th International Conference on Telecommunications and Signal Processing (TSP)*, pages 715–719, 2012.

[75] Xiao-Chen Yuan and Chi-Man Pun. Feature extraction and local zernike moments based geometric invariant watermarking. *Multimedia Tools and Applications*, 72(1): 777–799, 2014. ISSN 1573-7721.

[76] Xinbo Gao, Cheng Deng, Xuelong Li, and Dacheng Tao. Local feature based geometric-resistant image information hiding. *Cognitive Computation*, 2(2):68–77, 2010. ISSN 1866-9964.

[77] Huang Zhang and Xiao-Qiang Li. Geometrically invariant image blind watermarking based on speeded-up robust features and dct transform. In Yun Q. Shi, Hyoung-Joong Kim, and Fernando Pérez-González, editors, *The International Workshop on Digital Forensics and Watermarking 2012*, pages 111–119, 2013. ISBN 978-3-642-40099-5.

[78] Manuel Cedillo-Hernandez, Francisco Garca-Ugalde, M Nakano-Miyatake, and Hector Perez-Meana. Robust object-based watermarking using surf feature matching and dft domain. In *Radioengineering*, volume 22, pages 1057–1071, 2013.

[79] Hieu Dang, Witold Kinsner, and Yingxu Wang. Multiobjective image data hiding based on neural networks and memetic optimization. *WSEAS Transactions on Signal Processing*, 10:645–661, 2014.

[80] K. Haribabu, G. R. K. S. Subrahmanyam, and D. Mishra. A robust digital image watermarking technique using auto encoder based convolutional neural networks. *2015 IEEE Workshop on Computational Intelligence: Theories, Applications and Future Directions (WCI)*, pages 1–6, 2015.

[81] Daming Li, Lianbing Deng, Brij Bhooshan Gupta, Haoxiang Wang, and Chang Choi. A novel cnn based security guaranteed image watermarking generation scenario for smart city applications. *Information Sciences*, 479:432 – 447, 2019. ISSN 0020-0255.

[82] C. Chang, T. D. Kieu, and Y. Chou. A high payload steganographic scheme based on (7, 4) hamming code for digital images. In *2008 International Symposium on Electronic Commerce and Security*, pages 16–21, 2008.

[83] Tuan Duc Nguyen, Somjit Arch-int, and Ngamnij Arch-int. An adaptive multi bit-plane image steganography using block data-hiding. *Multimedia Tools and Applications*, 75 (14):8319–8345, 2016. ISSN 1573-7721.

[84] Li Liu, Chin-Chen Chang, and Anhong Wang. Data hiding based on extended turtle shell matrix construction method. *Multimedia Tools and Applications*, 76(10):12233–12250, 2017. ISSN 1573-7721.

[85] Chun-Cheng Wang, Wen-Chung Kuo, Yu-Chih Huang, and Lih-Chyau Wuu. A high capacity data hiding scheme based on re-adjusted gemd. *Multimedia Tools Appl.*, 77 (5):6327–6341, 2018. ISSN 1573-7721.

[86] Marwa Saidi, Houcemeddine Hermassi, Rhouma Rhouma, and Safya Belghith. A new adaptive image steganography scheme based on dct and chaotic map. *Multimedia Tools and Applications*, 76(11):13493–13510, 2017. ISSN 1573-7721.

[87] Shumeet Baluja. Hiding images in plain sight: Deep steganography. In *Advances in Neural Information Processing Systems 30*, pages 2069–2079. 2017.

[88] Haichao Shi, Jing Dong, Wei Wang, Yinlong Qian, and Xiaoyu Zhang. Ssgan: Secure steganography based on generative adversarial networks. In *Advances in Multimedia Information Processing – PCM 2017*, 2018.

[89] Denis Volkhonskiy, Ivan Nazarov, Boris Borisenko, and Evgeny Burnaev. Steganographic generative adversarial networks. *Proceedings of NIPS 2016 Workshop on Adversarial Training*, 2017.

[90] Pin Wu, Yang Yang, and Xiaoqiang Li. Stegnet: Mega image steganography capacity with deep convolutional network. 10, 2018.

[91] Haiping Lu, Xuxia Shi, Y. Q. Shi, A. C. Kot, and Lihui Chen. Watermark embedding in DC components of DCT for binary images. In *2002 IEEE Workshop on Multimedia Signal Processing.*, 2002.

[92] Haiping Lu, A. C. Kot, and Jun Cheng. Secure data hiding in binary document images for authentication. In *Proceedings of the 2003 International Symposium on Circuits and Systems, 2003. ISCAS '03.*, volume 3, pages III–III, 2003.

[93] Huijuan Yang, A. C. Kot, and Jun Liu. Semi-fragile watermarking for text document images authentication. In *2005 IEEE International Symposium on Circuits and Systems*, pages 4002–4005 Vol. 4, 2005.

[94] H. Y. Kim and J. Mayer. Data hiding for binary documents robust to print-scan, photocopy and geometric distortions. In *XX Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI 2007)*, pages 105–112, 2007.

[95] Huijuan Yang and Alex C. Kot. A general data hiding framework and multi-level signature for binary images. In *Digital Watermarking*, pages 188–202. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-92238-4.

[96] Younho Lee, Heeyoul Kim, and Yongsu Park. A new data hiding scheme for binary image authentication with small image distortion. *Information Sciences*, 179(22):3866 – 3884, 2009. ISSN 0020-0255.

[97] H. Cao and A. C. Kot. Eag: Edge adaptive grid data hiding for binary image authentication. In *Proceedings of The 2012 Asia Pacific Signal and Information Processing Association Annual Summit and Conference*, pages 1–6, 2012.

[98] Chung-Chuan Wang, Ya-Fen Chang, Chin-Chen Chang, Jinn-Ke Jan, and Chia-Chen Lin. A high capacity data hiding scheme for binary images based on block patterns. *Journal of Systems and Software*, 93:152 – 162, 2014. ISSN 0164-1212.

[99] Li Li, Qingzheng Hou, Jianfeng Lu, Qishuai Xu, Junping Dai, Xiaoyang Mao, and Chin-Chen Chang. A new pixels flipping method for huge watermarking capacity of the invoice font image. 2014.

[100] Ki-Hyun Jung and Yoo Kee-Young. Data hiding method in binary images based on block masking for key authentication. 277:188–196, 2014.

[101] Fatemeh Daraee and Saeed Mozaffari. Watermarking in binary document images using fractal codes. *Pattern Recognition Letters*, 35:120 – 129, 2014. ISSN 0167-8655.

[102] Qingzheng Hou, Dai Junping, Li Li, Jianfeng Lu, and Chin-Chen Chang. Scanned binary image watermarking based on additive model and sampling. *Multimedia Tools and Applications*, 74(21):9407–9426, 2015. ISSN 1573-7721.

[103] Qingzhong Liu, Andrew H. Sung, Zhongxue Chen, and Jianyun Xu. Feature mining and pattern classification for steganalysis of lsb matching steganography in grayscale images. *Pattern Recognition*, 41(1):56 – 66, 2008. ISSN 0031-3203.

[104] Yinlong Qian, Jing Dong, Wei Wang, and Tieniu Tan. Deep learning for steganalysis via convolutional neural networks. 2015.

[105] Lionel Pibre, Jérôme Pasquet, Dino Ienco, and Marc Chaumont. Deep learning for steganalysis is better than a rich model with an ensemble classifier, and is natively robust to the cover source-mismatch. *ArXiv*, abs/1511.04855, 2015.

[106] Yinlong Qian, Jing Dong, Wei Wang, and Tieniu Tan. Feature learning for steganalysis using convolutional neural networks. *Multimedia Tools and Applications*, 77(15):19633–19657, 2018. ISSN 1573-7721.

[107] J. Zeng, S. Tan, B. Li, and J. Huang. Large-scale jpeg image steganalysis using hybrid deep-learning framework. *IEEE Transactions on Information Forensics and Security*, 13(5):1200–1214, 2018. ISSN 1556-6013.

[108] S. Wu, S. Zhong, and Y. Liu. A novel convolutional neural network for image steganalysis with shared normalization. *IEEE Transactions on Multimedia*, pages 1–1, 2019. ISSN 1520-9210.

[109] Daniel Lerch-Hostalot and David Megas. Unsupervised steganalysis based on artificial training sets. *Engineering Applications of Artificial Intelligence*, 50:45 – 59, 2016. ISSN 0952-1976.

[110] Jishen Zeng, Shunquan Tan, Guangqing Liu, Bin Li, and Jiwu Huang. Wisernet: Wider separate-then-reunion network for steganalysis of color images. *CoRR*, 14:2735–2748, 2018.

[111] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. ISSN 1573-1405.

[112] Chris Harris and Mike Stephens. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151, 1988.

[113] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004. ISSN 1573-1405.

[114] Tony Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):79–116, 1998.

[115] M. Donoser and H. Bischof. Efficient maximally stable extremal region (mser) tracking. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 553–560, 2006.

[116] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346 – 359, 2008. ISSN 1077-3142.

[117] S. Leutenegger, M. Chli, and R. Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *Proceedings of the 2011 International Conference on Computer Vision*, pages 2548–2555, 2011.

[118] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988. ISSN 1573-1405.

[119] G. Agam, S. Argamon, O. Frieder, D. Grossman, D. Lewis. *CDIP Test Collection Project: http://ir.iit.edu/projects/CDIP.html*, 2006.

[120] *http://iapr-tc10.univ-lr.fr/index.php/resources/dataset-and-software/24-resources/databases/296-l3i-textcopies-dataset.*

[121] http://personal.psu.edu/xuy111/projects/cvpr2017_doc.html.

[122] M. S. Yasein and P. Agathoklis. A feature-based image normalization technique for handling geometric distortions. In *Conference Record of the Thirty-Ninth Asilomar Conference onSignals, Systems and Computers.*, pages 884–887, 2005.

[123] M. S. Yasein and P. Agathoklis. An image watermarking technique with improved resistance to geometric distortions using image feature points. In *2009 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pages 26–30, 2009.

[124] Richard O. Duda and Peter E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11–15, 1972. ISSN 0001-0782.

[125] Ehab Salahat and Murad Qasaimeh. Recent advances in features extraction and description algorithms: A comprehensive survey. *CoRR*, abs/1703.06376, 2017.

[126] Farzad Ebrahimi, Matthieu Chamik, and Stefan Winkler. Jpeg vs. jpeg 2000: an objective comparison of image encoding quality. In *The International Society for Optical Engineering*.

[127] Houssem Chatbri, Keisuke Kameyama, and Paul Kwan. A comparative study using contours and skeletons as shape representations for binary image matching. *Pattern Recogn. Lett.*, 76(C):59–66, 2016. ISSN 0167-8655.

[128] M. N. Do and M. Vetterli. The contourlet transform: an efficient directional multiresolution image representation. *IEEE Transactions on Image Processing*, 14(12): 2091–2106, 2005. ISSN 1057-7149.

[129] C. R. Maurer, Rensheng Qi, and V. Raghavan. A linear time algorithm for computing exact euclidean distance transforms of binary images in arbitrary dimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):265–270, 2003. ISSN 0162-8828.

[130] A. Witkin. Scale-space filtering: A new approach to multi-scale description. In *ICASSP '84. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 9, pages 150–153, 1984.

[131] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *CoRR*, abs/1411.4038, 2014.

[132] Zheng Zhang, Chengquan Zhang, Wei Shen, Cong Yao, Wenyu Liu, and Xiang Bai. Multi-oriented text detection with fully convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4159–4167, 2016.

[133] Dena Bazazian, Ral Gmez, Anguelos Nicolaou, Llus Gmez, Dimosthenis Karatzas, and Andrew D. Bagdanov. Fast: Facilitated and accurate scene text proposals through fcn guided pruning. *Pattern Recognition Letters*, 119:112 – 120, 2019. ISSN 0167-8655.

[134] Dena Bazazian, Raul Gomez, Anguelos Nicolaou, Lluís Gómez i Bigorda, Dimosthenis Karatzas, and Andrew D. Bagdanov. Improving text proposals for scene images with fully convolutional networks. *CoRR*, abs/1702.05089, 2017.

[135] C. Wick and F. Puppe. Fully convolutional neural networks for page segmentation of historical document images. *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, pages 287–292, 2018.

[136] X. Yang, E. Yumer, P. Asente, M. Kraley, D. Kifer, and C. L. Giles. Learning to extract semantic structure from documents using multimodal fully convolutional neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4342–4351, 2017.

[137] C. Tensmeyer and T. Martinez. Document image binarization with fully convolutional neural networks. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, pages 99–104, 2017.

[138] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv 1409.1556*, 2014.

[139] Sheraz Ahmed, Muhammad Imran Malik, Marcus Liwicki, and Andreas Dengel. Signature segmentation from document images. In *2012 International Conference on Frontiers in Handwriting Recognition*, pages 425–429, 2012. ISBN 978-0-7695-4774-9.

[140] U. M. Butt, S. Ahmad, F. Shafait, C. Nansen, A. S. Mian, and M. I. Malik. Automatic signature segmentation using hyper-spectral imaging. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 19–24, 2016.

[141] Chandranath Adak, Bidyut Chaudhuri, and Michael Blumenstein. Writer identification and verification from intra-variable individual handwriting. *IEEE Access*, PP, 2017.

[142] C. Adak, S. Marinai, B. B. Chaudhuri, and M. Blumenstein. Offline bengali writer verification by pdf-cnn and siamese net. In *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, pages 381–386, 2018.

[143] Vernica Aubin, Marco Mora, and Matilde Santos-Peas. Off-line writer verification based on simple graphemes. *Pattern Recognition*, 79:414 – 426, 2018. ISSN 0031-3203.

[144] Y. Akao, A. Yamamoto, and Y. Higashikawa. Assisting forensic writer verification by visualizing diversity of digit handwritings: An approach by multidimensional scaling of earth mover's distance. In *2014 14th International Conference on Frontiers in Handwriting Recognition*, pages 110–115, 2014.

[145] Ameur Bensefia and Thierry Paquet. Writer verification based on a single handwriting word samples. *EURASIP Journal on Image and Video Processing*, (1):34, 2016. ISSN 1687-5281.

[146] A. Parziale, A. Santoro, and A. Marcelli. Writer verification in forensic handwriting examination: A pilot study. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 447–452, 2016.

[147] N. R. Howe, A. Fischer, and B. Wicht. Inkball models as features for handwriting recognition. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 96–101, 2016.

[148] Michael Kalbitz, Tobias Scheidat, Benjamin Yüksel, and Claus Vielhauer. Towards automated forensic pen ink verification by spectral analysis. In *Digital Forensics and Watermarking*, pages 18–30, 2017. ISBN 978-3-319-64185-0.

[149] F. Kleber, S. Fiel, M. Diem, and R. Sablatnig. Cvl-database: An off-line database for writer retrieval, writer identification and word spotting. In *2013 12th International Conference on Document Analysis and Recognition*, pages 560–564, 2013.

[150] Y. Kim, H. Jung, D. Min, and K. Sohn. Deeply aggregated alternating minimization for image restoration. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 284–292, 2017.

[151] Xiao-Jiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 2810–2818, 2016. ISBN 978-1-5108-3881-9.

[152] Y. Chen and T. Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1256–1272, 2017. ISSN 0162-8828.

[153] Noam Yair and Tomer Michaeli. Multi-scale weighted nuclear norm image restoration. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3165–3174, 2018.

[154] Brendan Kelly, Thomas P. Matthews, and Mark A. Anastasio. Deep learning-guided image reconstruction from incomplete data. *ArXiv*, abs/1709.00584, 2017.

[155] Joseph Y. Cheng, Feiyu Chen, Marcus T. Alley, John M. Pauly, and Shreyas S. Vasanawala. Highly scalable image reconstruction using deep neural networks with bandpass filtering. *CoRR*, abs/1805.03300, 2018.

[156] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1637–1645, 2016.

[157] W. Lai, J. Huang, N. Ahuja, and M. Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5835–5843, 2017.

[158] Wei Han, Shiyu Chang, Ding Liu, Mo Yu, Michael J. Witbrock, and Thomas S. Huang. Image super-resolution via dual-state recurrent networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1654–1663, 2018.

[159] Xintao Wang, Kevin Yu, Chen Dong, and Chen Change Loy. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 606–615, 2018.

[160] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu. Residual dense network for image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2472–2481, 2018.

[161] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Learning a single convolutional super-resolution network for multiple degradations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3262–3271, 2018.

[162] C. Ledig, L. Theis, F. Huszr, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 105–114, 2017.

[163] Jianxin Lin, Yingce Xia, Tao Qin, Zhibo Chen, and Tie-Yan Liu. Conditional image-to-image translation. volume abs/1805.00251, 2018.

[164] Shuang Ma, Jianlong Fu, Chang Wen Chen, and Tao Mei. Da-gan: Instance-level image translation by deep attention generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5657–5666, 2018.

[165] Takahiro Isokane, Fumio Okura, Ayaka Ide, Yasuyuki Matsushita, and Yasushi Yagi. Probabilistic plant modeling via multi-view image-to-image translation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2906–2915, 2018.

[166] Yunjey Choi, Min-Je Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. volume abs/1711.09020, 2017.

[167] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, pages 2672–2680. 2014.

[168] Zicheng Guo and Richard W. Hall. Parallel thinning with two-subiteration algorithms. *Commun. ACM*, 32(3):359–373, 1989. ISSN 0001-0782.

[169] F. Shafait and T. M. Breuel. The effect of border noise on the performance of projection-based page segmentation methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4):846–851, 2011. ISSN 0162-8828.

[170] A. Antonacopoulos, D. Bridson, C. Papadopoulos, and S. Pletschacher. A realistic dataset for performance evaluation of document layout analysis. In *2009 10th International Conference on Document Analysis and Recognition*, pages 296–300, 2009.

[171] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. ISSN 1057-7149.

[172] Longjiang Yu, Xiamu Niu (IEEE member), and Shenghe Sun. Print-and-scan model and the watermarking countermeasure. *Image Vision Comput.*, 23(9):807–814, 2005. ISSN 0262-8856.

[173] A. Hassane, S. Al Maadeed, J. Aljaam, and A. Jaoua. Icdar 2013 competition on gender prediction from handwriting. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1417–1421, 2013.