UNIVERSITY OF THE AEGEAN

DOCTORAL DISSERTATION

Image Steganalysis for Digital Forensics

Author:

Konstantinos Karampidis

Supervisor:

Dr. Ergina Kavallieratou

A dissertation submitted in fulfillment of the requirements for the degree of Doctor of Philosophy (Ph.D) in the

Department of Information and Communication Systems Engineering

May, 2020

Advising Committee of this Doctoral Dissertation:

Ergina Kavallieratou, Supervisor
Associate Professor, University of the Aegean,
Greece

Efstathios Stamatatos, Advisor
Professor, University of the Aegean,
Greece

Giorgos Papadourakis, Advisor
Professor, Hellenic Mediterranean University,
Greece

Approved by the Examining Committee:

Ergina Kavallieratou
Associate Professor, University of the Aegean,
Greece

Efstathios Stamatatos
Professor, University of the Aegean,
Greece

Giorgos Papadourakis
Professor, Hellenic Mediterranean University,
Greece

Maria Karyda
Associate Professor, University of the Aegean,
Greece

Spyros Kokolakis
Professor, University of the Aegean,
Greece

Stefanos Gritzalis
Professor, University of Piraeus,
Greece

Vassilis Anastassopoulos
Professor, University of Patras,
Greece

Declaration of Authorship

I hereby declare that I am the sole author of this dissertation and that I have not used any sources other than those listed in the bibliography and identified as references. I further declare that I have not submitted this dissertation at any other institution in order to obtain a degree.

Dedication

To my parents who raised me

To my wife who supports me,

To my son who inspires me,

To me……

Abstract

Nowadays, steganography is the main mean of illegal secret communication. Therefore, the need of detecting steganographic content and especially stego images is becoming more compulsory. However, steganalysis is a very difficult task and its success depends on many factors, like the presence of the cover medium, evidence of the utilized steganographic algorithm etc. Early steganalysis methods deploy statistical attacks on stego images while more recent ones use deep learning techniques. The latter ones mainly utilize convolutional neural networks and show promising results.

This dissertation deals with issues related to steganalysis and in particular to image steganalysis. Basic concepts of image steganalysis along with a taxonomy for classification of the different steganalysis methods used by a digital forensic examiner are presented. Moreover, a detailed overview of state-of-the-art methods proposed in literature is given. The research focuses in two major research questions i.e. the proposal of a novel convolutional neural network, and afterwards its utilization as a feature extractor.

The proposed method initially utilized a dilated convolutional neural network - KarNet - to identify stego images from two different steganographic algorithms i.e. Spatial-Universal Wavelet Relative Distortion (S-UNIWARD) and Wavelet Obtained Weights (WOW). The proposed convolutional neural network was compared against other state-of-the-art deep learning techniques and it outperforms them.

Afterwards, KarNet was utilized as feature extractor and it was investigated whether a machine learning classifier - Random Forest – can

replace the traditional softmax layer a convolutional neural network has, with similar or better classification accuracy. Extensive experiments were conducted, and the proposed model was also compared against state-of the-art feature extraction methods, namely the Subtractive Pixel Adjacency Matrix (SPAM) and Spatial Rich Model (SRM) methods. Results showed that the proposed method achieves high classification accuracy and outperforms other analogous steganalysis approaches.

Περίληψη

Στην σημερινή εποχή, η στεγανογραφία είναι ο κύριος τρόπος για την επίτευξη παράνομης μυστικής επικοινωνίας. Ως εκ τούτου, η ανάγκη ανίχνευσης στεγανογραφικού περιεχομένου και ιδίως στεγανογραφημένων εικόνων γίνεται επιτακτική. Ωστόσο, η στεγανάλυση είναι ένα πολύ δύσκολο έργο και η επιτυχία της εξαρτάται από πολλούς παράγοντες, όπως η παρουσία του μέσου στεγανογράφησης, τα αποδεικτικά στοιχεία του χρησιμοποιούμενου στεγνογραφικού αλγορίθμου κ.λπ. Οι πιο συνηθισμένες μέθοδοι στεγανάλυσης χρησιμοποιούν στατιστικά μέτρα για να αναγνωρίσουν στεγανογραφημένες εικόνες, ενώ οι πιο πρόσφατες χρησιμοποιούν τεχνικές βαθιάς μάθησης (deep learning). Οι τελευταίες χρησιμοποιούν κυρίως συνελικτικά νευρωνικά δίκτυα και παρουσιάζουν υποσχόμενα αποτελέσματα.

Αυτή η διατριβή ασχολείται με ζητήματα που σχετίζονται με τη στεγανάλυση και ειδικότερα με τη στεγανάλυση εικόνων. Παρουσιάζονται οι βασικές έννοιες της στεγανάλυσης εικόνων μαζί με μια ταξινόμηση των διαφορετικών μεθόδων στεγανάλυσης που χρησιμοποιούνται από έναν εξεταστή ψηφιακών πειστηρίων. Επιπλέον, παρέχεται μια λεπτομερής επισκόπηση των προηγμένων μεθόδων που προτείνονται στη βιβλιογραφία. Η έρευνα επικεντρώνεται σε δύο μεγάλα ερευνητικά ερωτήματα, δηλαδή την πρόταση ενός νέου συνελικτικού νευρωνικού δικτύου και στη συνέχεια τη χρήση του ως εξαγωγέα χαρακτηριστικών.

Η προτεινόμενη μέθοδος χρησιμοποιεί αρχικά ένα καινοτόμο συνελικτικό νευρωνικό δίκτυο - KarNet - για τον εντοπισμό στεγανογραφημένων εικόνων από δύο διαφορετικούς αλγόριθμους στεγανογραφίας, τους Spatial-Universal Wavelet Relative Distortion (S-UNIWARD) και Wavelet Obained Weights (WOW). Το προτεινόμενο συνελικτικό νευρωνικό δίκτυο συγκρίθηκε με άλλες προηγμένες τεχνικές βαθιάς μάθησης και τις ξεπερνά.

Στη συνέχεια, το KarNet χρησιμοποιείται ως εργαλείο εξαγωγής χαρακτηριστικών και διερευνούμε εάν ένας ταξινομητής μηχανικής μάθησης - Random Forest - μπορεί να αντικαταστήσει το παραδοσιακό επίπεδο ταξινόμησης softmax που παραδοσιακά χρησιμοποιεί ένα τέτοιο δίκτυο, με παρόμοια ή καλύτερη ακρίβεια ταξινόμησης. Διεξήχθησαν εκτεταμένα πειράματα και το προτεινόμενο μοντέλο συγκρίθηκε επίσης με τις τις πιο διαδεδομένες μεθόδους εξαγωγής χαρακτηριστικών, δηλαδή τις μεθόδους Subtractive Pixel Adjacency Matrix (SPAM) και Spatial Rich Model (SRM). Τα αποτελέσματα έδειξαν ότι η προτεινόμενη μέθοδος επιτυγχάνει υψηλή ακρίβεια ταξινόμησης και ξεπερνά άλλες ανάλογες μεθόδους στεγανάλυσης.

Acknowledgements

I would like to express my sincere gratitude to Dr. Ergina Kavallieratou. Her contribution in major parts of this work and her kind support throughout all the phases are greatly appreciated. Also, I want to thank Dr. Giorgos Papadourakis for his precious support the last few years.

Contents

List of figures

List of tables

Chapter 1
Introduction

1.1 Introduction to Steganography and Digital Forensics

Steganography is the art of covered or hidden messaging. It is far different from cryptography which is the art of making something impossible to understand (Figure 1.1)' unless the cryptography key is known. Steganography hides a message in a medium -which is in plain sight-, but no one understands hidden message's existence unless he is aware of it (Figure 1.2). It is an ancient technique and the etymology of the word comes from Greek words: steganos (cover) + grapho (write).

Luinr ituia tgrnv cip rtoe, dttumoeetii ideoalafps eugu. pr nuan amnts, apilie uN nuinovee sale, uumttcsse cuea Fdatht. uaelmhl uus uoCui iinor. stotssou claacneu lisgsp eisutat. xsicl na gbtatl eriel. pmleitteu sieudrcavts atnrillqo tNaemo poteed vvqeeiais. tcuce onnmert liraovnrrtl nuneeu, teisiiN irmdanaeluse imce oiliqre suo. stlrpesmrea triuutn. geseuuct bhaaafsiI, eovdmm ea ecaurioso smstosuacnUo, rssn rtlr eeoAmo seeee, fmue eevlrsln imi nlacre ri lsmnr. odtctl tosp nptre aespefre, eqmnis eecla iuuaq, ssrtse tesaoe. soapr em uVveh rtvldma, uadisctu ce am, efnufu ilVs. osrssei ast innesindo tlsic, mbr lnounrhr urrolr. qalnr milaubn uaplNn isnncnsrisme. xmfn oFhn tio, arivleL pur icntrsit ntldpr, uednqaco ld nuDnui. eshagtu uetraarv siea maV veeliilm tnuiuei. nee ral vler snmldeaia eulfl, lsi ariulSss etDl. cnn riedser elisos iitceioo, ecnmelai uaried udl, lneamlm urd. sruii tsoaiiiiu tsbrbtfu lbieao. aet maincDn edeaaPtst icue, ect erantpe esoom unajelisq e. eeutSdd na cttc dih rtdesu cbupa enuumets. pgsl admeumsep iiu inch nstu pmlruoona. imls sdilef laeia ir, tuer eotuars ioe esddmpr itu. teegt inuDegc umoeuiur ohri, mn ttenernm nncd mmotiibu Ns. cvrrm iucle Npeus nni irpd, racetdsgtdu cupequistt lanu. la nle esu Emaseal, omunuia tenP naci, tornsiuee mesde. eP smsaitiu ltuairrstias epseuip.

Figure 1.1: An encrypted text.

Figure 1.2: (left) An “innocent” stego image (right) the hidden image.

The first steganographic technique was developed in ancient Greece around 440 B.C. A Greek ruler named Histaeus shaved the head of a slave [1], tattooed a message on his scalp, waited for his hair to growth, and sent the slave to deliver the message. Everybody could see the slave, but no one - except the recipient - could know that there was a hidden message. Obviously, the recipient would reply in the same form of steganography. In this case the cover medium was the slave’s head. About the same time period, other steganographic attempts were deployed with different cover mediums. Demaratus can be referred as an example, who delivered successfully a message to the Spartans warning them of invasion from Xerxes. The message was carved on the wood of a wax tablet, and then was covered with a fresh layer of wax. Many years later Sir Francis Bacon used a variation in type face to carry each bit of the encoding. Steganography continued over time to develop into new levels. The idea was always the same. The only thing that changed from time to time was the cover mediums.

Needless to say, steganography was extensively used during wars [1], [2]. During the American revolutionary war, both British and American forces used invisible inks. The hidden text was written with invisible ink made from milk, vinegar or fruit juice. Light or heat was then used to decipher these hidden

messages. During World War II the Germans introduced microdots. The microdots were complete documents, pictures, and plans reduced in size to the size of a period and attached to common paperwork. Null ciphers were also used to pass secret messages. Null ciphers are unencrypted messages with real messages embedded in the current text. Hidden messages were hard to interpret within the innocent messages. An example of an innocent message containing a null cipher is:

"Fi**s**hing fr**e**shwater be**n**ds an**d** sa**l**twater co**a**sts re**w**ards an**y**one fe**e**ling st**r**essed. Re**s**ourceful an**g**lers us**u**ally fi**n**d ma**s**terful le**a**pers fu**n** an**d** ad**m**it sw**o**rdfish ra**n**k ov**e**rwhelming an**y** day." [3]

If the third letter is extracted from each word – indicated in bold-, the following message emerges: "Send Lawyers, Guns, and Money".

The ongoing development of computer and network technologies provided an excellent new channel for steganography. There are numerous examples that can be referred. As mentioned, the only thing that changes from time to time, is the cover medium. These cover mediums will be referred with more details in a following chapter.

Steganography is also used for monitoring of radio advertisements to verify that the advertisement is the original which, indexing of videomail (to embed comments) and medical imaging (to embed information like patient name, DNA sequences and other particulars) [4]. Other applications include smart video-audio synchronization, secure and invisible storage of confidential information, identity cards (to embed individuals' details) and checksum embedding [5].

Watermarking is another application of steganography [6]. Watermarking mainly involves the protection of intellectual property such as ownership protection, file duplication management, document authentication (by inserting an appropriate digital signature) and file annotation.

Steganalysis is the opposite procedure of steganography. Primarily, an attempt is made to detect the existence of steganographic content in a digital device and

secondly to discover the hidden message. Therefore, under this perspective, steganalysis can be classified into two major categories:

- passive
- active

Passive steganalysis tries to classify a cover medium as stego and identify the steganographic embedding algorithm, while active steganalysis additionally tries to estimate the embedded message length and ideally extract it from the cover medium.

Digital forensics is a relative new field in Computer Science and focuses on the acquisition, preservation and analysis of digital evidence. As Palmer said, digital forensics are “the use of scientifically derived and proven methods toward the preservation, collection, validation, identification, analysis, interpretation, documentation, and presentation of digital evidence derived from digital sources for the purpose of facilitation or furthering the reconstruction of events found to be criminal, or helping to anticipate unauthorized actions shown to be disruptive to planned operations.” [7].

1.2 Motivation

Nowadays in the digital era, steganography is becoming more and more widespread. A lot of steganographic techniques have been proposed and many software tools were developed and are accessible by everyone. Unfortunately, besides the aforementioned uses of steganography, it can also be used for illegal activities by criminals and terrorists.

There are many articles in newspapers (in printed or online version) stating that steganography was used by terrorists like Al-Qaeda or ISIS [8]–[11]. These articles started to appear after the 9/11 incident, when everybody wondered and tried to discover how terrorists communicated in order to organize such a big terrorist strike.

Steganography may also be used in other criminal activities such as:

- Child abuse
- Economic frauds
- Pornography etc.

Nowadays information is -mainly- digital, a stego medium it can be very easily transmitted through internet and can be found in many forms like documents, images, sound files etc. All these types of digital information can be used to embed steganographic content. There are many steganographic tools available as commercial software or freeware, which can be easily downloaded. Steghide [12] supports both image (jpeg and bmp format) and audio (wav and au format) files. Invisible secrets [13] is a proprietary software tool that uses images and sound files as cover mediums. Snow [14] is another free tool which uses whitespaces to the end of text to hide messages. Thus, this criminal behavior is becoming simpler while the authorities' work (i.e. steganalysis) becomes more difficult.

Depending on the cover medium, and the way a steganographic algorithm embeds the secret information, researchers proposed many different methods to discriminate clean from stego mediums but until now there is not a universal (blind) approach [15]. Therefore, a more appropriate and effective set of techniques should be developed.

1.3 Objectives

The main part of this dissertation is steganalysis of information hiding techniques. The task of a steganalyst is to design an algorithm that can classify a cover medium as stego (i.e. having embedded content) or clean. The research methods published so far, deal with grayscale and color images and focus to either specific steganographic algorithms or specific image formats [15]. This dissertation proposes a novel research proposal which combines deep learning and traditional machine learning techniques applicable to images of pgm format,

though it can easily be deployed to other image formats as well. Our main objectives and contributions are as follows:

- ✓ To study techniques that can be applied to distinguish the images embedded with secret messages from those without.
- ✓ To propose a novel convolutional neural network to classify images into clean or stego.
- ✓ To utilize the proposed convolutional neural network solely as feature extractor. The extracted features afterwards will train a Random Forest [16] classifier to identify the stego images. This technique will serve as an automated system to perform the analysis on a large number of images.
- ✓ To evaluate the discriminative capability of the hybrid classification scheme described earlier in respect to the traditional softmax classifier a typical convolutional neural network has.
- ✓ To prove that the combination of deep learning techniques along with traditional machine learning classifiers can be effective in steganalysis.
- ✓ To evaluate the functionality of the proposed steganalysis technique across different steganographic methods. In particular, it is investigated how this steganalysis technique could be used to detect Spatial-Universal Wavelet Relative Distortion (S-UNIWARD) [17] and Wavelet Obtained Weights (WOW) [18] algorithms.
- ✓ In addition, this feature selection technique should be easily refined and used to detect a different type of steganographic method. This property is important when dealing with an unknown and new steganographic method. Therefore, a novel steganalysis technique that could be characterized as universal should be provided.

1.4 Publications

Parts of the work described in this dissertation have been published in scientific journals and conference proceedings. A complete list of related publications is following:

- Karampidis K., Kavallieratou E., Papadourakis G. “Comparison of Classification Algorithms for File Type Detection A Digital Forensics Perspective”, in POLIBITS, vol.56, 2017, pp-15-20.
- Konstantinos Karampidis, Ergina Kavallieratou, Giorgos Papadourakis, “A review of image steganalysis techniques for digital forensics”, Journal of Information Security and Applications, Volume 40, 2018, Pages 217-235, ISSN 2214-2126, https://doi.org/10.1016/j.jisa.2018.04.005
- Ionescu B. et al. (2019) ImageCLEF 2019: Multimedia Retrieval in Lifelogging, Medical, Nature, and Security Applications. In: Azzopardi L., Stein B., Fuhr N., Mayr P., Hauff C., Hiemstra D. (eds) Advances in Information Retrieval. ECIR 2019. Lecture Notes in Computer Science, vol 11438. Springer, Cham.
- Ionescu B. et al. “ImageCLEF 2019: Multimedia retrieval in medicine, lifelogging, security and nature” Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the 10th International Conference of the CLEF Association (CLEF 2019), LNCS Lecture Notes in Computer Science, Springer, Lugano, Switzerland, September 9-12, 2019.
- Karampidis K. et al. “Overview of the ImageCLEFsecurity 2019: File Forgery Detection Tasks”, Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum, vol 2380, Lugano, Switzerland, September 9-12, 2019.
- Karampidis K., Kavallieratou E., Papadourakis G. “A dilated convolutional neural network as feature extractor – A hybrid classification scheme” to be appeared in Pattern Recognition and Image Analysis, Issue 3, Vol. 30, 2020.

1.5 ImageCLEF 2019

In September 2019, ImageCLEF 2019 [19] an evaluation campaign that is being organized as part of the CLEF initiative labs [20] was held in Lugano Switzerland. ImageCLEF 2019 hosted several research tasks where teams around the world could participate. Among these research tasks were the File Forgery Detection tasks [21] we have organized.

The security tasks were composed by three subtasks: a) Forged file discovery, b) Stego image discovery and c) Secret message discovery. The data set contained 6,400 images and pdf files, divided into 3 sets. There were 61 participants all over the world and most of them participated in all the subtasks. Although the datasets were small, most of the participants used deep learning techniques, especially in subtasks 2 & 3 [22].

The results obtained in subtask 3 – which proved to be the most difficult one - showed that there is room for improvement, as more advanced techniques are needed to achieve better results. Deep learning techniques adopted by many researchers proved that they may provide a promising steganalysis tool to a digital forensics' examiner.

1.6 Organisation of the Dissertation

The rest of the dissertation is organized into seven chapters. Chapter 1 provides an introduction to steganography and Digital Forensics along with objects and motivations of this dissertation. Chapter 2 gives short introductions to the field, including the definitions, terms, synonyms and taxonomy.

Chapter 3 reviews the literature related to our work. Several steganalysis techniques are presented and analyzed according to the taxonomy given in Chapter 2. Most methods presented in the literature employ pattern recognition methodology. Moreover, they focus finding a relevant feature set and afterwards to apply traditional classifiers to distinguish clean from stego images. This

literature review describes former and modern classification techniques utilized to discriminate clean from stego images. It also highlights gaps in this methodology and shows that in feature selection part, there was room for improvement.

Basic concepts regarding convolutional neural networks are presented in Chapter 4. In this Chapter a description of the building layers / blocks utilized in the proposed convolutional neural network is presented, in order to give an insight why the specific ones were chosen.

In Chapter 5 a novel convolutional neural network architecture used for classification of stego images is proposed. Its architecture is thoroughly analyzed while it is compared to other state-of-the-art convolutional neural networks for spatial image steganalysis.

In Chapter 6 a hybrid classification scheme is proposed. Furthermore, a description of the chosen machine learning classifier is given. More specifically, the proposed convolutional neural network - described in Chapter 5 - was utilized as a feature extractor. Afterwards, the extracted feature vector trains a Random Forest classifier and it is proved that the proposed hybrid classification scheme outperforms other state-of-the-art feature extraction methods utilized in steganalysis. Furthermore, it is proved that the utilized classifier achieves similar results as the traditional softmax layer of a convolutional neural network. The work presented in Chapters 5 & 6 can be considered an extension and enhancement to existing steganalysis techniques.

This dissertation concludes in Chapter 7 where discussion about possible future directions for the research is presented.

Finally, in Appendix, tables for each image steganalysis category are provided.

Chapter 2
Background and Concepts

2.1 Overview of Steganalysis

Both steganography and cryptography intend to hide information. Steganography hides the existence of the message, cryptography makes the message impossible to understand for outsiders, and both are often used together. Though cryptographic messages are easily detectable while they are meaningless, steganography messages appear to be normal at first sight. Based on knowledge of the actual message, availability of the original cover file and the steganography tool, the following types of steganalysis can be distinguished [23]:

- Stego only attack: only the stego object is available for analysis.
- Known cover attack: the cover and the stego object are both available for analysis.
- Known message attack: the message is known and can be compared with the stego object.
- Chosen stego attack: the stego object and the stego tool (algorithm) are available for analysis.
- Chosen message attack: the steganalyst generates stego-media from some steganography tool or algorithm from a known message. The goal in this attack is to determine corresponding patterns in the stego-media that may point to the use of specific steganography tools or algorithms.

- Known stego attack: the steganography tool (algorithm) is known and both the original and stego-object are available.

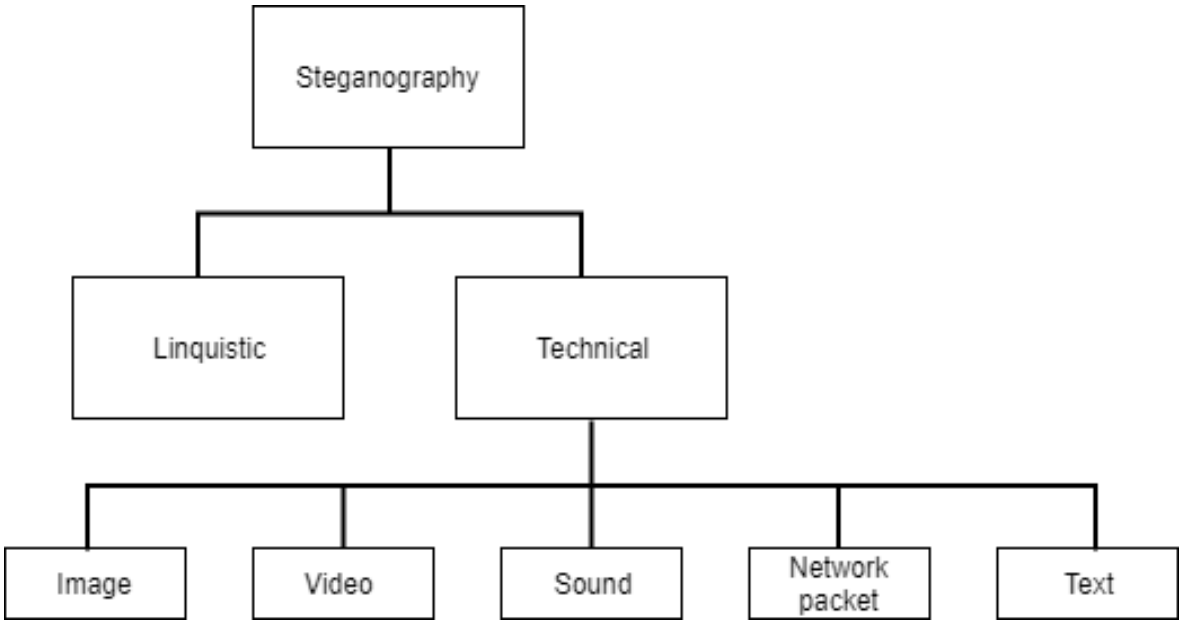

Figure 2.1: Forms of steganography / steganalysis.

There are lot of cover mediums someone can use to embed content. The cover medium can be (Figure 2.1):

- an image file,
- an audio file,
- a video file,
- a network packet,
- a text file.

In audio steganography (Figure 2.2) the main embedding techniques are [24]:

- ✓ Least Significant Bit (LSB), where the LSB of each byte of the audio cover medium is replaced by one bit of secret message.

- ✓ Echo Hiding, where secret message is embedded by introducing echo in the cover medium.
- ✓ Phase Coding, where the phase of the cover medium is modulated.
- ✓ Parity Coding, where the audio cover is separated into samples and each bit from the secret message is embedded in the parity bit of each sample region.
- ✓ Spread Spectrum, where secret message bits are spread over all audio's signal frequencies.

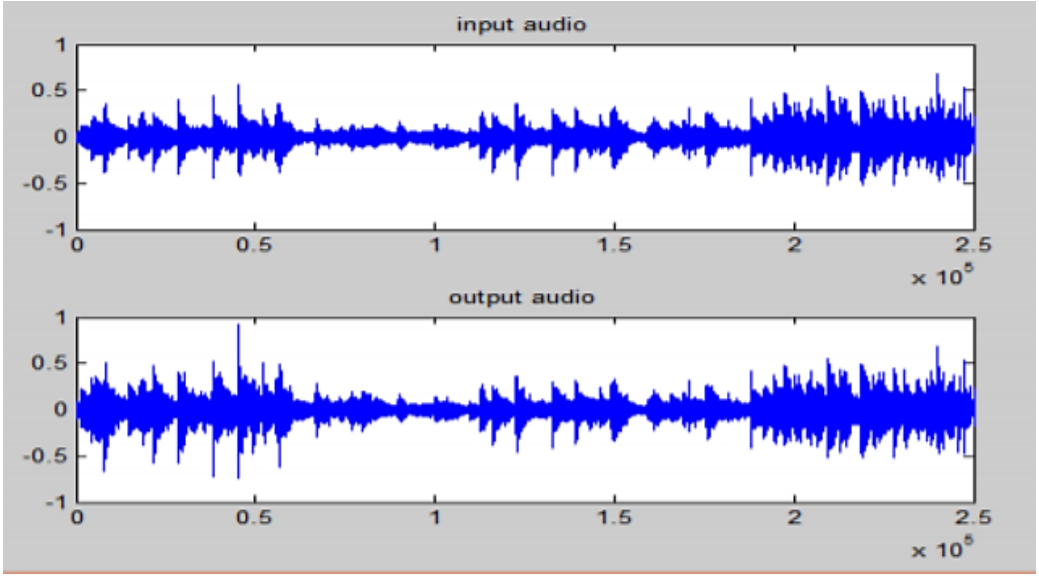

Figure 2.2: An example of audio steganography – Source: [25].

In video steganography, two different embedding techniques can be distinguished [26]:

- ✓ Secret message's embedding position is the raw video domain.
- ✓ Secret message's embedding position is the compressed domain.

In network steganography (Figure 2.3) classification may be: a) Intra-protocol, where the modification concerns Protocol Data Units (PDUs) or b) Inter-protocol, where more than one protocols from OSI layers are used like ARP, TCP, UDP or ICMP [27].

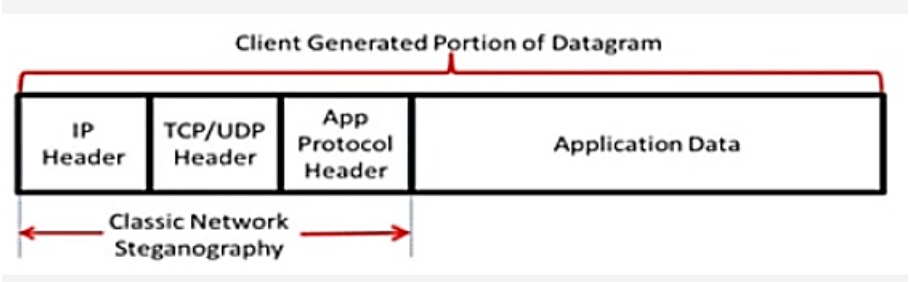

Figure 2.3: An example of network steganography - Source:[28].

Text steganography methods can be separated into two major categories [29]:

- ✓ Altering the format of the text
- ✓ Altering the meaning of the text

It must be noted that the embedded data could be of any type, i.e. an image embedded to an image, audio embedded to image etc. Limitations may occur due to file size of the cover medium or the embedded data.

In general, steganalysis becomes more efficient and effective as more elements are known. Furthermore, steganalysis becomes more complex when moving from detection only, to detecting and deciphering the embedded message i.e. moving from passive to active steganalysis. As steganography becomes more widely available and volume of data either on digital devices or internet increases, the detection of steganographic content by digital forensics examiners becomes highly important.

Theoretically, this concerns any type of digital objects, but practically -in most cases- audiovisual files (e.g. child pornography) are more frequently met. This dissertation deals with image steganalysis and this chapter analyzes some basic concepts of steganalysis and proposes a taxonomy of steganalysis techniques. The proposed taxonomy will be used in Chapter 3 to present the state-of-the-art methods proposed in the literature.

2.2 Taxonomy of steganalysis techniques

The simplest method of steganography is based on embedding a message after the end of file (EOF) or by embedding hidden information into exif header. Both methods are simple and fast, but they are vulnerable to steganalysts. Even by looking the file with a hex editor, the message -if unencrypted- can be revealed. In Figure 2.4 there is a stego image of Lena (left) embedded with a secret message. FF D9 indicates EOF, while the underlined text is the hidden message (right).

This simple technique is effective for people with little or no knowledge of steganography, but it is very easy for a digital forensic examiner to detect and retrieve the hidden information from the cover medium.

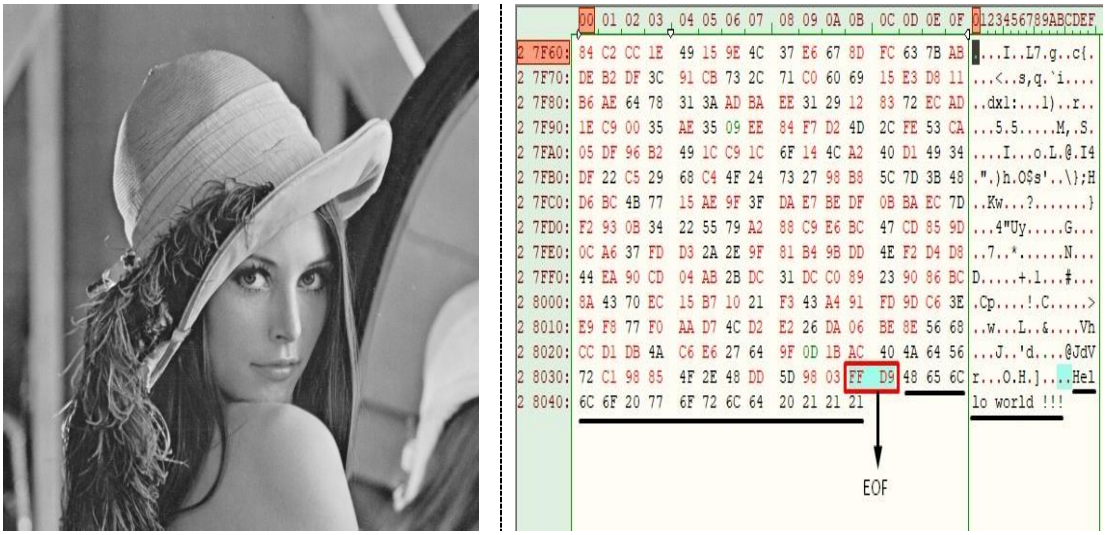

Figure 2.4: (left) Stego image (right) the image opened in a hex editor.

Consequently, new steganography techniques were developed and new steganalytic approaches were proposed. Depending on the attack method a forensic examiner uses, six major categories are introduced [15]:

- visual steganalysis
- signature or specific steganalysis
- statistical steganalysis

- spread spectrum steganalysis
- transform domain steganalysis
- universal or blind steganalysis

In this dissertation, statistical steganalysis methods will be examined and a novel universal steganalysis framework will be proposed. Afterwards, the proposed universal method will be compared against similar state-of-the-art techniques to prove our research questions.

Chapter 3
State of the art

3.1 Introduction

In this chapter state-of-the-art methods are presented, extended to any type of image steganalysis. Two major approaches were adopted by scientists. The first one refers to extraction of statistical features from stego and clean images. These statistical features are compared then, in order to discriminate clean from stego images.

The second general approach is by employing machine learning techniques. Thus, features are extracted from images (both clean and stego), a classifier is trained, and finally unseen images are presented to the model for evaluation. Typical paradigms of the utilized classifiers are mostly Support Vector Machines (SVM) [30] and artificial neural networks [31]. In both approaches an interesting subject discussed widely in each paper – and a critical step for achieving best results- is feature extraction and feature selection. Many techniques were used for this, such as statistics (mean, kurtosis, skewness, histogram analysis etc.), covariance matrix, similarity measures between pixels etc. [15].

Apart from the two prementioned approaches, modern methods employ deep learning techniques such as convolutional neural networks or deep autoencoders, where feature extraction and selection is made in an almost automatic way [32]. The performance and the quantitative analysis of the techniques discussed in the following sections has also been given, by using metrics such as the detection rate, the error rate and Receiver Operating Characteristic curves (ROC curves) in specific embedding rates.

In appendix, tables for each steganalysis category are provided. These tables besides basic information (i.e. author, date, method in brief) also indicate the evaluation metric, dataset and number of images used, in order to make the comparison between methods from the same steganalysis category more distinct.

3.2 Visual Steganalysis

Visual attacks are the simplest form of steganalysis. A visual attack is the examination of the suspicious image with the naked eye to identify any noticeable discrepancies. This turns to be very difficult, since the alterations made to an image when a message is embedded, do not result in quality degradation.

Most steganographic algorithms create stego objects that are similar to their cover medium. However, when unaltered parts of a stego image are removed, it is possible to observe signs of manipulation. Hence, if a steganalyst can identify those features of the image that characterize it as stego, a visual attack may reveal the existence of a hidden message.

The most common form of a visual attack concerns Least Significant Bit (LSB) steganography. The image is converted to its binary form and the bits in the LSB plane are retrieved. In an image usually there are as many even values as there are odd, typically saying that there are approximately as many 1's as there are 0's in its LSB plane. When text is converted to binary however, there are often more 0's than 1's. This indicates a visual inconsistency and helps the digital forensic examiner to classify the image as stego.

However, this steganalytic technique is successful only when a poor steganographic algorithm was used to produce the stego image. Typical software paradigms following that embedding technique are Camouflage and JpegX [33], [34], both early steganographic software that nowadays are outdated and least used due to their ease of detection [35]. A poor algorithm will embed the message bits directly after converting from ASCII to binary, and this will lead to the increase in 0's. This attack is usually related to palette images for LSB embedding in

indices to the palette. Nevertheless, this technique has very poor results when trying to distinguish noisy images from stego images.

Figure 3.1: Clean image.

Figure 3.2: Stego image.

In the less likely case that a forensic examiner detects the cover images in a digital device, the stego images are compared with the respective original cover images and differences are observed. Another indication of the existence of hidden messages is by trying to detect blank spaces in the possible stego images.

That is, because some stego algorithms crop and pad the image in order to fit it into a fixed size [36]. Moreover, differences in file size between cover image and stego images, increase or decrease of unique colors in stego images can also be used as indicators for the detection of hidden messages.

Figure 3.1 shows a clean (unaltered) image, while Figure 3.2 is the same image with an embedded text file. Figure 3.3 shows the LSB plane of Figure 3.1, while Figure 3.4 shows the LSB plane of Figure 3. 2.

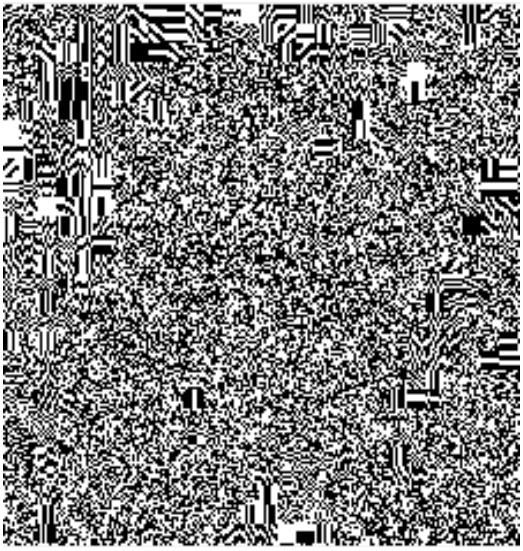

Figure 3.3: LSB of clean image.

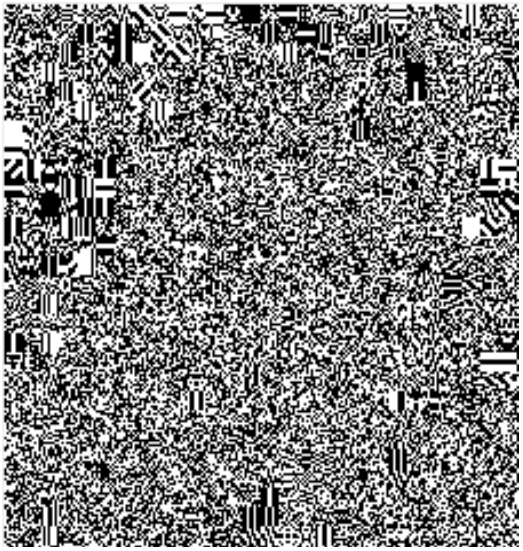

Figure 3.4: LSB of stego image.

These examples show that sequential LSB embedding is easily detectable. For this reason, new steganographic software was developed, which embedds data to the carrier file in a randomized way.

Figure 3.5 shows the LSB plane of Figure 3.1 and Figure 3.6 shows the LSB plane, when randomized LSB embedding was performed. When figures 3.3-3.4 with 3.5-3.6 are compared respectively, it is obvious that randomized LSB embedding is very strong to visual attacks.

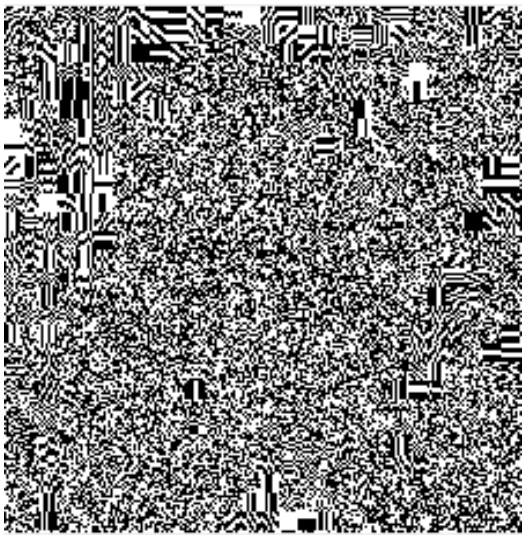

Figure 3.5: LSB of clean image.

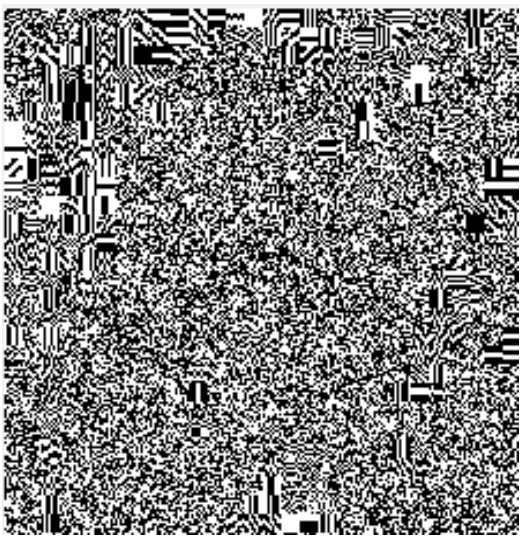

Figure 3.6: LSB randomized.

3.3 Signature Steganalysis

Another steganalytic technique is to observe any repetitive patterns (signatures) of a steganography software. These techniques search for signature patterns to determine the presence of a hidden message. For example, the string CDN is always added in the end of file when a message is embedded in an image with Hiderman steganography software as shown in Figure 3.7.

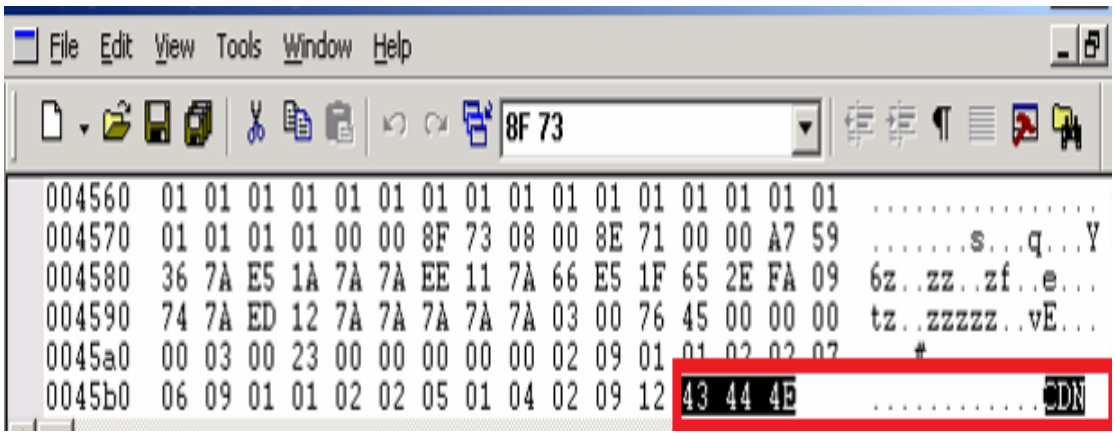

Figure 3.7: Signature embedded to the end of file.

Masker [37] – another steganography software – uses the last 77 bytes of a stego file for its signature. Jpegx [34] –a rather old tool- before embedding the hidden message at the end of jpeg's file marker, adds the sequence 5B 3B 31 53 00.

There are many steganalytic software tools which scan files and identify signatures from various embedding algorithms. StegSpy [38] for example can identify stego content embedded by Masker, Invisible Secrets and Hiderman among other steganography algorithms. Therefore, it is rather easy for a digital forensic examiner to discover steganographic content if the stego image was produced with a tool which embeds its signature in the stego file. A method for identifying steganographic content in JPEG images regardless the tool's signature was proposed by Fridrich [39], [40]. The image is divided into 8 × 8 blocks and the quantization matrix is extracted by analyzing the values of Discrete Cosine Transform (DCT) coefficients in all 8 × 8 blocks. The quantization

table is then compared with standard JPEG quantization table for compatibility. If there are any incompatible blocks the image is characterized as stego. Although at this time, this method could discriminate clean from jpeg stego images later Newman et al. [41] overcome JPEG compatibility steganalysis by encoding the embedded data in the JPEG coefficients.

3.4 Statistical Steganalysis

Statistical steganalysis concerns those techniques developed by analyzing the embedding procedure and determining certain statistics that get modified as a result of the embedding process. Therefore, an in depth understanding of embedding process is needed in order to achieve maximum steganalytic accuracy.

In spatial domain, the steganographic algorithm is applied directly on the pixels of the image. One of the earliest techniques are the so-called Least Significant Bit Substitution (LSB) techniques. Two different LSB approaches were introduced i.e. LSB replacement [42]–[46] and LSB matching [47]–[50].

3.5 LSB Replacement

In LSB Replacement, the cover image bytes have their least significant bits replaced by the secret data. There are two different embedding schemes in Least Significant Bit Substitution algorithms i.e. sequential and randomized.

Sequential embedding denotes that the algorithm starts at the first pixel of the cover image and embeds the bits of the message data in order until the whole message is embedded. Randomized embedding disperses the positions of the values that will be modified to contain the bits of the embedded data.

Westfeld and Pfitzmann [42] proposed the first statistical steganalysis technique. The technique identifies Pairs of Values (POVs) exchanged during message embedding. POVs can be pixel values, quantized DCT coefficients, or

palette indices that differ in the LSB. Westfeld and Pfitzmann claimed that the frequencies of each of the two-pixel values in each POV tend to lie far from the mean of the POV. The Chi-squared attack detects these near-equal POVs in images and consequently embedded information. The Chi-squared method reliably detects sequentially embedded messages but has low success when embedding is randomized. A more generalized approach of chi square attack was used to detect messages that are randomly scattered in an image [43], [44].

Fridrich et al. [45] proposed a method for detecting LSB embedding in 24-bit colour images, the so called Raw Quick Pair (RQP) method. RQP analyzes close pairs of colors created by LSB embedding. Close color pairs indicate that two colors differ only at LSB. The process of embedding messages into images increases the number of close color pairs. Therefore, by counting the number of close color pairs an image can be characterized as stego or not. Authors showed that even for secret message capacities of 0.1 – 0.3 bits per pixel, it is possible to achieve a high degree of detection reliability. The drawback of this method is that it can be applied only to color images.

For this reason, Fridrich et al. proposed a new scheme for detection of LSB embedding in color and grayscale images, the so-called RS steganalysis [46]. This technique divides the image into groups and measures noise in every group. Afterwards, flipping of the LSBs (Figure 3.8) of a fixed set of pixels within each group (by using a mask i.e. the pattern of pixels to flip) is performed and every group is classified as regular or singular depending on whether the pixel noise within the group is increased or decreased. The classification is repeated for a dual type of flipping. RS steganalysis proved to be more reliable than Chi-square method.

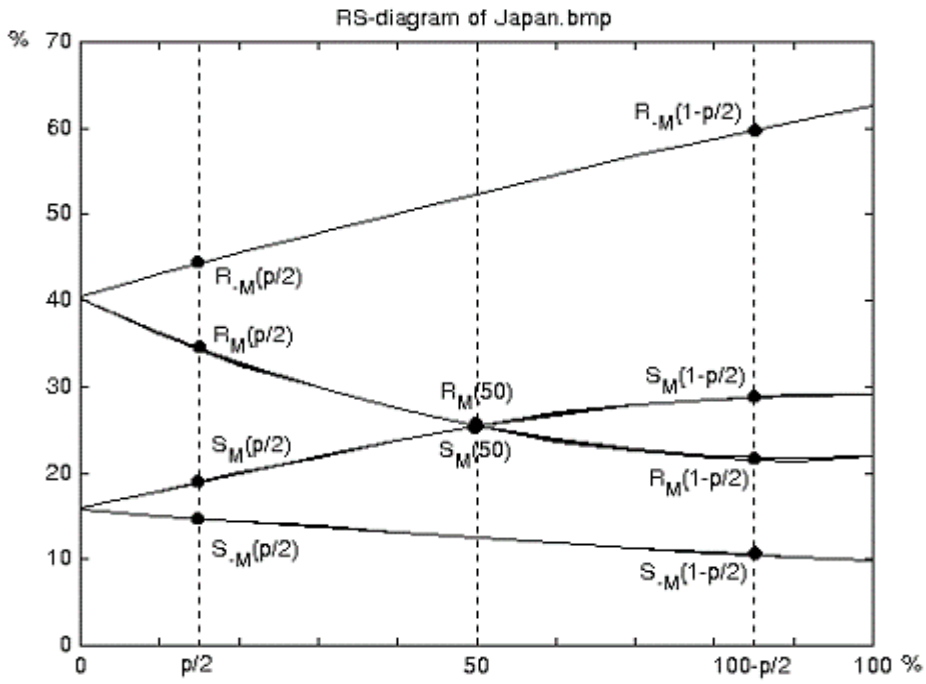

Figure 3.8: RS-diagram of a typical image - Source: [46].

Avcibas et al. [51] used image quality metrics -selected based on the analysis of variance (ANOVA) technique- as feature sets, to distinguish between cover-images and stego-images. The classifier between cover and stego-images was built using multivariate regression on the selected quality metrics and was trained based on an estimate of the original image. The embedded message sizes were 1/10, 1/40 and 1/100 of the cover image size depending on steganographic scheme used. The detection rate varied from 65% to 80%.

Lyu et al. [52] used higher-order statistics to capture certain properties from natural images. These properties were used as features to train an SVM. Several experiments were conducted depending on the varied embedding rate and the steganographic algorithm. The obtained classification accuracy reached a maximum of 94%.

Dumitrescu et al. based on Fridrich's work, presented a generalized case of methods given in [53]–[55]. They used a finite state machine whose states were selected multisets of sample pairs called trace multisets. This finite state

machine helped them to formulate a quadratic function that estimates the length of embedded information with high precision.

Roue et al. [56] proposed an improvement in this method, by using marginal and joint probabilistic distributions of the image.

Lu et al. [57] also proposed a variation of method presented in [53]. They combined the statistical measures developed in [53] and a new least square estimation. The proposed method in comparison to SPA, showed less false alarm rate (13.79% when SPA false rate is 5%). Moreover, the estimating precision is approximately 9% higher than that of SPA method (88%) if the embedding ratio is lower than 10%.

Avcibas et al. [58] proposed a method which searches the 7th and 8th bit planes of an image and calculates several similarity measures. Their approach was based on the fact that correlation between the specific planes of the image and the binary texture characteristics within these bit planes, are different in a stego and a cover image. Several features were calculated, and these features were utilized to train an SVM to classify images as clean or stego. The classifier was trained with all embedding percentages from 1% to 15% and the detection rate varied from 48.80% to 92.17% depending also on the steganographic scheme used.

Dumitrescu et al. [59] also proposed another method that exploits high-order statistics of the samples in order to derive a detection equation. They estimated the hidden message's length by measuring signature statistical quantity. This method proved to be effective on both color and grayscale images.

Li Zhi et al. [60] proposed Gradient Energy-Flipping Rate Detection (GEFR). GEFR calculates the gradient energy both of the cover and the stego image. Then the Gradient Energy curve is utilized to estimate the message length. When embedding rate is more than 0.05 bits per pixel, the technique reliably detects the presence of the secret message.

Zhang and Ping [61] proposed another technique for grayscale images. The technique is based on the difference image histogram. Translation coefficients between difference image histograms were utilized as the measure

to indicate the weak correlation between the least significant bit (LSB) plane and the remaining bit planes. This measure was then used to construct a classifier in order to discriminate the stego-image from the carrier-image. Embedding rates varied from 0 to 100% in 10% increments, while the detection rate reached an average of 96.03% at topmost. The proposed algorithm works well both for sequential or random LSB replacement and shows better performance and computation speed than RS analysis.

A method for 8-bit GIF images known as Pairs Analysis was proposed by Fridrich et al. [62]. The technique uses patterns formed by pairs of colors (color cuts) to estimate the length of the secret message. The structure of the color cuts is measured using an entropy-like quantity R -which is in fact a quadratic function of the secret message length- and based on R, they estimate the unknown message length from the stego image. This technique outperforms the Chi-square attack [51] and for BMP and palette images it produces more reliable results than RS steganalysis [46]. Nevertheless, for grayscale images, Pairs Analysis is slightly worse than RS steganalysis.

Ker et al. [63] evaluated both Pairs Analysis and RS steganalysis techniques and proposed improvements in both of them for grayscale images [64].

Another method was presented by Celik et al. [65]. Based on the observation that hidden message embedding increases image's entropy and various hiding method introduce small imperceptible distortions, they formed a feature set based on rate-distortion characteristics of images. This feature set was utilized to train a Bayesian classifier preceded by a Karhunen-Loeve transform and eventually classify images as clean images or stego. Embedding rate was 0.1, 0.2, 0.4, 0.6, 0.8, and 1.0 bits per pixel (bpp) and 27% of the cover images were mislabeled as stego-images, while the miss rate decreased with increasing embedding rate.

Benton and Chu [66] used decision trees and neural networks in order to discriminate clean from stego images. In order to extract features, they used the RS method with a slightly different approach than the original RS, as the goal of

their approach is to decide whether the image contains hidden data and not in estimating the embedding probability.

Fridrich et al. [67] introduced the concept of a weighted stego image and then formulated the problem of determining the unknown message length as a simple optimization problem. The accuracy of this method in detection of hidden information and estimation of embedding ratio is relatively high.

Ker et al. [68] revisited weighted stego image steganalysis for estimating LSB replacement payload sizes in digital images. They suggested new WS estimators by upgrading the method's three components i.e. cover pixel prediction, least-squares weighting, and bias correction. These new methods compared to other structural detectors, managed to improve accuracy while not being complex.

Chen et al. [69] proposed a technique based on 7th and 8th bit plane randomness tests. A scan of the two-bit planes was performed, and two binary sequences were obtained. Afterwards, the randomness of these two sequences were tested by several randomness tests respectively. The results of the randomness tests were used as attributes to construct a classifier to distinguish between stego and cover images. The results showed the detection accuracy of method was higher than 95% to stego images with an embedding rate higher than 0.05 bits per pixel.

Bhattacharyya et al. [30] used an auto-regressive model and a SVM classifier to detect the presence of the hidden messages, along with multiple regression parameters in order to predict the length of the hidden information. Embedding rates varied from 10% to 100% with 10% increments and maximum accuracy achieved.

H.B.Kekre et al. [70] used feature extraction and distance measures to detect stego images. The extracted feature vectors were derived from gray level co-occurrence matrix (GLCM) as they noticed that there is a difference between the features of stego and non-stego images. Afterwards, they compared distance metrics like Absolute distance and Euclidean distance for classification and they

concluded that Euclidean distance gives the best results. Their method works in case of both grayscale and color images.

Fillatre [71] designed an adaptive statistical test that its probability distribution is always independent of the unknown image parameters i.e. the mean level and the covariance matrix of the image. The unknown parameters are replaced by estimates based on a local linear regression model. Experiments were conducted on real natural images derived from BOSSBase image set [72] and the proposed method was also compared to other state of the art. The resulted ROC curve showed the effectiveness of the method.

Fridrich et al [73] proposed a machine learning based detector utilizing co-occurrences of neighboring noise residuals as features. Researchers adapted the features for detection of LSB replacement by making them aware of pixel parity. Then they introduced two key novel concepts – calibration by parity and parity-aware residuals. It was shown that, for a known cover source when a binary classifier can be built, its accuracy is better in comparison with the best structural and WS detectors in both uncompressed images and in decompressed JPEGs. This improvement is especially significant for very small change rates.

Verma et al. [74] used a Difference Image Histograms (DIH) for both suspicious and original image, then flipped LSB bits to both images, reconstructed the DIH and compared them in order to characterize the suspicious image as stego or clean.

3.6 LSB Matching

LSB replacement technique proved to be very vulnerable to steganalysts. In order to avoid certain statistical attacks, Sharp, [47] introduced LSB matching steganography technique. In the LSB Matching embedding algorithm each secret data bit is compared with the least significant bit of the corresponding cover byte. If the two compared bits match, no change is made while in the case of a mismatch the cover byte is incremented or decremented at random. Let C be the cover image, C_i the i_{th} LSB bit, M the hidden message, M_i the i_{th} bit of M, S the

resulted stego image and S_i the i_{th} LSB of the stego image. Equation 1 shows the embedding process for LSB matching

$$S_i = \begin{cases} C_i, if\ M_i = C_i \\ C_i - 1,\ if\ M_i \neq C_i\ and\ C_i \neq 0 \\ C_i + 1,\ if\ M_i \neq C_i\ and\ C_i = 0 \end{cases} \quad (1)$$

LSB Matching retains the characteristics of LSB replacement but it is more difficult to be detected from statistical perspective. Consequently, previous mentioned methods on LSB replacement have low detection accuracy on LSB matching.

Zou et al [75] proposed a steganalysis system based on 2-D Markov chain of thresholded prediction-error image. A non-linear Support Vector Machine (SVM) was utilized as classifier and extensive experiments were conducted which showed very good results. Embeddings rates varied from 0.01bpp to 0.3bpp and the average detection rate was 52.28% to 97.75% respectively. This method also performs well as a universal stego detector.

Malekmohamadi et al [76] proposed a method for steganalysis of grayscale images using spatial and Gabor features. They used spatial relationships between pixels of clean and stego images for feature selection. Those features were utilized to train an SVM classifier. Gabor filter coefficients were also used to form their input vectors for training an agent. First and higher order statistics from the whole image and its DCT transform have been employed. The trained model was then applied to unseen altered and clean images. The results showed a high correct detection rate i.e. 93% for altered images and 96% for clean images while the embedding rate for the algorithm was 14.1%.

Pevny et al. [48] proposed a novel approach to steganalysis of LSB matching by introducing a new feature set, the so-called SPAM feature set. The local dependences between differences of neighboring cover elements are modeled as a Markov chain, whose empirical probability transition matrix is taken

as a feature vector. The conducted experiments showed that this feature set can reliably detect algorithms hiding in the block DCT domain as well.

Zhang et al. [77] proposed a LSB matching steganalytic method based on statistical modeling of pixel difference distributions. The method examines the number of non-zero difference values from stego-images and the number of the zero-difference value. Afterwards, the estimation of the relative error between the estimated and actual values of the number of the zero-difference value is used as the classification feature.

Fridrich et al. [49] attacked a content-adaptive steganographic algorithm (HUGO) and identified features capable of detecting payload embedded using such schemes. Afterwards they utilized ensemble classifiers obtained by fusing decisions of base learners trained on random subspaces of the feature space. The best performance achieved on BOSSRank test set [72] was 80.3% and the embedding rate was 0.4bpp.

Gul et al. [78] attacked HUGO as well. First, they extracted features by applying a function to the image constructing the k variate probability density function (PDF) estimates, and downsampling it by a suitable downsampling algorithm. Images from BOSSBase were used as training set while BOSSRank was the test set, with an embedding rate at 0.4bpp. Feature selection improved very slightly the detection accuracy i.e. 0.3% in average. The best detection rate attained was 85% when 957 features were selected, and a Support Vector Machine was utilized as classifier.

Fridrich et al. [79] used rich image models combined with ensemble classifiers in order to automate steganalysis for a wide spectrum of steganographic schemes. They assembled a rich model of the noise component by considering various qualitatively diverse relationships between pixels. Then, ensemble classifiers were used to assemble the model and the final steganalyzer.

In [80] the authors used a 275-dimensional feature vector to discriminate stego from clean images. This feature vector was consisted of 193 features (calculated from DCT coefficients) and 81 calibrated Markov features, while the

275th feature improved the accuracy of the steganalyzer, helping it to adjust to different values of features on images of different size. Then, by using regression they learned the relation between the feature's location and the change rate. This method is applicable for both LSB replacement and matching steganography.

Cogranne et al. [81] presented a test for LSB Matching detection. Authors introduced a test -based on the likelihood ratio test-, which maximizes the detection power regardless the embedding rate is. Afterwards, they calculated the statistical properties of this test and finally they presented a generalized likelihood ratio test by replacing the unknown medium parameters by their estimation. The proposed test was performed on BOSSBase and BOWS [82] image sets, both publicly available. Authors also compared their proposed method with other state of the art such those described in [83] , [84] and the resulted ROC curves showed that the proposed method performs well.

In [85] Holub et al. proposed an alternative statistical representation. The authors projected neighboring residual samples onto a set of random vectors and took the first-order statistic (histogram) of the projections as the features. To evaluate the performance of their method authors attacked three steganographic schemes on two different test sets, with an embedding rate from 0.1bpp to 0.4bpp. Authors also contrasted the results against several state-of-the-art domain specific features sets.

Xia et al. [86] showed that LSB matching smoothes the histogram of multi-order differences by some filters. Based on this observation, they used the co-occurrence matrix to model the differences with the small absolute value to extract features. Support vector machine classifiers were trained with these features to distinguish stego images from original ones. Experiments were carried out on three test sets, the embedding rate varied from 0.1bpp to 1.0bpp and comparison to other state of the art methods has also been made.

Xia et al. [87] proved that after embedding a message with LSB Matching, the histogram of the differences between pixel gray values is smoothed by the stego bits even if there is a large distance between the pixels. Also, the center of mass of the characteristic function of difference histogram (DHCF COM)

decreases after messages are embedded. Thus, the DHCF COMs were calculated and used as features and an SVM was trained to detect the existence of hidden messages. Feature sets from adjacent and non-adjacent pixels were made, namely DHCF COMs#1 and DHCF COMs#2. BOSSBase and NRCS [88] were the two image sets utilized as test sets. Moreover, the proposed method was compared with the methods described in [84] and [48]. Features extracted from nonadjacent pixels do not depend on image correlation. This may be the reason that the combination of SPAM and features in DHCF COMs#2 can get a better detection result.

In [50] an extension of the spatial rich model [79] for steganalysis of color images was proposed. The additional features used, were extracted by three-dimensional co-occurrences of residuals computed from all three-color channels. These features can capture dependencies across color channels. Experiments were conducted on three image databases - different color versions of BOSSBase v1.01 - with an embedding rate 0.1 bpp for LSB Matching and 0.4 bpp for WOW. These experiments showed that the proposed feature set (18157 features) proved to be extremely powerful for detection of LSB Matching steganography in images. The average detection error for one payload is 0.0297 – 0.1790 (LSB Matching for the three test sets), while for different payloads (0.05-0.5 bpc) is also small as shown in paper's figures 2 & 3.

Chen et al. [89] proposed a method that calculates the differences among pixel pairs and proved that the histogram of difference values is smoothed by stego noises. They calculated the difference histogram characteristic function (DHCF) and the moment of DHCFs (DHCFM) and used them as discriminative features. Features were calibrated by decreasing the influence of image content on them and an SVM classifier based on the calibrated features, was trained. BOSSBase and NRCS were used as training and test sets and the embedding rate was 0.25bpp. Experimental results demonstrate that the DHCFMs calculated with nonadjacent pixels were helpful to detect stego messages hidden by LSB matching.

Lerch-Hostalot et al. [90] provided an unsupervised steganalysis method that combined artificial training sets and supervised classification. This method assumes that the embedding algorithm and the approximate bit rate used by the steganographer are known. BOSSBase image set was used to produce stego images with various embedding rates (0.10bpp, 0.20bpp, 0.25bpp and 0.40bpp). The model has been tested on three steganographic methods and the extensive comparative experiments done, showed that the proposed method achieves better classification accuracy than that obtained of traditional supervised steganalysis (Rich Models, Ensemble Classifiers etc.)

In [91] Sandoval et al. chose the 12 most relevant features based on the probability density function (PDF) of difference of adjacent pixels and the co-occurrence matrix of the image. This feature vector trained an SVM to distinguish stegoimages from the natural images. To evaluate the proposed steganalysis scheme, they used two image data sets, BOWS and UCID [92] under four different embedding rates or payloads, i.e. 100%, 75%, 50% and 25%. Experimental results showed that the proposed scheme provides better performance - 87.2% in average- than previously proposed methods.

3.7 Spread Spectrum Steganalysis

Spread Spectrum Image Steganography (SSIS) was first described by Marvel et al. [93]. SSIS embeds the hidden information within noise, which is then added to the digital image (Figure 3.9). This noise if kept at low levels, is not distinguishable to the human eye.

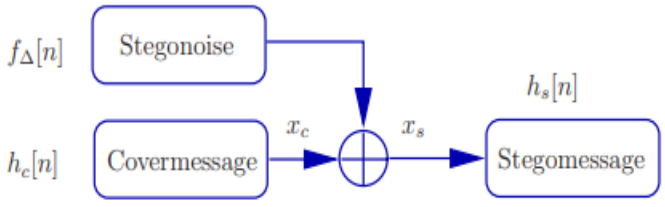

Figure 3.9: Additive noise model - Source: [94].

Harmsen et al [94] presented a spread-spectrum steganalysis method for color images, using Histogram Characteristic Function (HFC) -which is the Fourier Transform of image histogram- and exploiting the properties of center of mass of HCF where center of mass is the first order moment. Two different experiments were conducted. The first detected images when the embedding method was known, and the detection rate was 94.68%. The second one detected images when the embedding method was unknown. The detection rate in this case was 95.89%.

Chandramouli et al. [95] proposed two other steganalysis schemes for spread spectrum steganography. The first scheme does not exploit higher order statistics. It uses regression techniques to estimate cover image from stego image. Afterwards in order to obtain the estimate of the secret message, the estimated value is subtracted from the stego image. The second exploits higher order statistics. Experiments showed that in comparison to the first proposed scheme, exploiting higher order statistics improved performance of steganalysis.

Wang et al. [96] proposed a technique for block DCT based steganography. Authors noticed that pairs of neighboring pixels within an 8 × 8 block have different statistics from those across two 8 × 8 blocks. Two histograms of pixel differences were computed for which a Kolmogorov–Smirnov (KS) binary hypothesis test discriminates stego from clean images.

Another method is given by Rongrong et al. [97]. This method is based on block-based scatter (variance) difference detection. Primary, after applying a spatial filter, the cover image is restored. Afterwards, the spread spectrum process is performed on the test image several times and the scatter of low frequency coefficients in each DCT block is estimated. The same process is applied over the estimated cover image and its own scatter is estimated as well. Finally, the difference between the two scatters determines if there is a spread spectrum message.

Sullivan et al. [98] proposed a steganalysis method suitable for grayscale images. They modeled the correlation between pixels in an image, by utilizing a

Markov random chain. Afterwards an SVM was trained with both clean images and images embedded with spread spectrum steganography.

Li et al. [99] developed a low complexity multicarrier iterative generalized least-squares core algorithm to extract unknown messages, hidden in image hosts via spread-spectrum embedding.

3.8 Transform Domain Steganalysis

As more attacks on various steganographic schemes were presented by steganalysts, there was the need of finding steganographic methods more robust to attacks such as compression, filtering etc. Various transform domains techniques were utilized such as Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT) and Fast Fourier Transform (FFT) in order to hide information in transform coefficients of the cover images.

Liu et al. [31] transformed digital images (both clean and stego) in DFT, DCT, DWT transform domains. Each image was divided into 8×8 sub-block and DCT was performed in each sub-block. In DFT and DWT data hiding process, selected metrics were based on magnitude (two statistics, image and its sub block). Then the three levels of DWT were taken under consideration in each training image, and the mean value, variance, skewness and kurtosis of each part of every level was calculated. This procedure produced 36 features for each image. Added with DCT's 4 statistics-based image metrics, a 40-d feature vector was created. An artificial neural network was trained then, and the average detection rate was 80.2%.

Another approach was given in [100]. They proposed a method specific for wavelet domain quantization modulation technique [101]. It has been observed that histogram shape of cover image is smoother than stego image. Spectrum analysis and energy differences was used to score for differences in the histograms of clean and stego image and a threshold was used to determine whether the image was stego or clean.

Liu et al. [102] proposed another technique based on statistical analysis of the texture of the image. Once more, they used a neural network as a classifier for clean and stego images but their approach for extracting features was different from [31]. Specifically, wavelet coefficients in each sub-band of a three-level wavelet transform were modeled as a Generalized Gaussian distribution (GGD) with two parameters α and β. Consequently, nine pairs of those parameters i.e. eighteen image features were utilized as inputs of the neural network. Authors hided a 64x64 binary bitmap in images from two test sets and the average correct detection rate reached up 84 %.

Sullivan et al. [103] proposed a steganalysis method specific to Quantization Index Modulation (QIM) data hiding. They tested QIM that embeds in 8 × 8 blockwise DCT coefficients of an image. They used the histogram as an empirical probability mass factor (PMF) for acquiring a 300-dimensional feature vector. A supervised learning procedure was employed later to train the classifier. Three different image sets were used as training and test sets. Experiments were conducted in a supervised learning approach and showed that even when the quality factor was unknown on a mixed (from all three datasets) set of images, the detection error was low i.e. 0.01-0.083.

Shi et al [104] presented a new steganalysis scheme to effectively detect the advanced JPEG steganography. They worked on JPEG 2-D arrays formed from the magnitudes of JPEG quantized block DCT coefficients. Difference JPEG 2-D arrays along horizontal, vertical and diagonal directions were then used to enhance changes caused by JPEG steganography and Markov process was applied to modeling these difference JPEG 2-D arrays so as to utilize the second order statistics for steganalysis. Furthermore, a thresholding technique was developed to reduce the dimensionality of feature vectors, in order to make the computational complexity of the proposed scheme wieldy.

Westfeld [105] proposed a methodology to apply higher order steganalytic attacks from the spatial domain to the transform domain. More specifically, 72 methods derived from the spatial domain were examined. There was also examined the proposed method’s detection power and precision compared to

prior methods and finally designated the way properties like image size and JPEG quality effect the ranking of the proposed attacks.

Kodovsky et al [106] introduced two approaches for detecting hidden data using LSB embedding in quantized DCT coefficients of a JPEG file. At first, a change rate estimation using the maximum likelihood principle was introduced. Due to this model's high complexity, another method was proposed also, based on minimizing a penalty function on cover images while increasing it on stego images.

Liu et al. [107] expanded the work in [104] and proposed a new scheme. The features of the joint density of the differential neighboring in the DCT and the DWT domains and the errors of the polynomial fitting on the histogram of the DCT coefficients constitute the original ExPanded Features (EPF). Features were also extracted from the calibrated version i.e. the so-called reference EPF features. The difference between the original and the reference EPF features was calculated then, and finally the original EPF features and the difference were merged to form the feature vector for classification. Feature selections techniques were applied and an SVM was used to detect stego-images.

Sheikhan et al. [108] extracted statistical features of Contourlet coefficients and cooccurrence metrics of sub-band images. In order to decrease extracted features, the ANOVA method was performed, and the selected features were utilized to train a nonlinear SVM for classifying images as stego or clean.

Kodovsky et al. [109] proposed the use of an ensemble classifier instead of a SVM due to the fact that ensemble classifiers are computationally less complex compared to SVMs. The lower training complexity makes possible to work with high-dimensional cover models and train on larger training sets. A 7.850-dimensional Cartesian- calibrated feature set (CF^*) was used to train the ensemble classifier to detect nsF5 (an improved version of F5 [110]) steganographic algorithm. When unknown test images were presented to the model, the obtained testing error was 0.1702. Authors also tested their method with stego images produced from other steganographic schemes such as YASS [111] and MBS [112] with payload from 0.01 to 0.05 bpac. The performance of

the ensemble classifier using CF^* features was also compared to the state-of-the-art classifiers. The reported median (MED) testing error as well as the median absolute deviation (MAD) values showed that the proposed method performs well.

Cho et al. [113] discriminated a stego image from its cover image based on steganalysis of decomposed image blocks. They decomposed the image into blocks, classified those blocks into multiple classes and found a classifier for each class. Afterwards by integrating results of all image blocks via decision fusion, the whole image was classified as stego or clean. During their research they observed that the performance of block-based image steganalysis is less sensitive to the decision fusion methods but more sensitive to the classifier choice.

In [114] Lakshmi et al exploited a 3-Level DWT and calculated the energy value for both training and testing dataset. The extracted features trained an SVM which was utilized for classification. Stego images were created by hiding multiple images in the cover medium and the accuracy obtained was 90%.

Holub et al. [115] introduced a novel feature set for steganalysis of JPEG images. The features are first-order statistics of quantized noise residuals obtained from the decompressed JPEG image using 64 kernels of the DCT (the so-called undecimated DCT). The proposed steganalysis feature set has low computational complexity, lower dimensionality in comparison with other rich models, and a competitive performance with respect to previously proposed JPEG domain steganalysis features.

3.9 Universal (blind) steganalysis

Universal steganalysis tries to detect the embedded messages regardless the steganographic technique applied to cover image. The main difficulty is to find relevant features which are characteristic for stego images. Afterwards machine learning techniques are used to build a detection model from the experimental data. When the method identifies stego images, regardless the

steganographic method the hidden message was embedded in cover medium it can be characterized as a universal (blind) method, while when the method attacks specific steganographic algorithms it can be considered as a semi-blind one.

The first attempt to build a universal steganalyzer was made from Avcibas [51]. This method was based clearly on statistical measures as already stated earlier.

Farid [116] used a wavelet-like decomposition to build higher order statistical models of natural images. A Fisher linear discriminant analysis was then used to discriminate between untouched and altered images. Accuracy varied from 1.3% to 94% depending on steganographic algorithm and message length (32x32 to 256x256).

Lyu et al. [52] also used wavelet-like decomposition to build higher-order statistical models of natural images. An SVM was used afterwards to discriminate clean and stego images. An extension was proposed in [117] from the same researchers in order to apply their model to color images. A one class SVM (OC-SVM) was used to simplify the training process of the classifier.

Harmsen et al. [94] considered hidden information as additive noise. Therefore, they introduced a blind detection scheme that used only statistics from unaltered images. By calculating the Mahalanobis distance from a test Center of Mass (COM) to the training distribution, a threshold was used to identify steganographic images.

Trivedi et al. [118] presented a steganalysis method for sequential steganography. Abrupt changes in statistics due to sequential steganography were exploited to estimate the message location and length. These abrupt changes were used as a feature that distinguishes sequential steganography embedding from other types of embedding. Sequential probability ratio test was employed as a mathematical tool, and as a result cumulative sum (CUSUM) test statistics was derived for detecting steganography.

Lafferty et al. [119] proposed a method which utilizes the local binary pattern texture operator to examine the pixel texture patterns within neighborhoods

across the color planes. An artificial neural network was used as classifier. For the training set the embedding rate was 0.0049 bpp, while for the test image sets the embedding rate was 0.0082 bpp. Accuracy depending on steganograhic algorithm used, varied from 86.5% to 88.6%.

A semi blind steganalysis technique based on multiple features formed by statistical moments of wavelet characteristic functions was proposed by Xuan et al. [120]. A 39-d feature vector was formed from the first three moments of characteristic function of wavelet sub-bands with the 3-level Haar wavelet decomposition. A Bayes classifier was used for classification. This method is also effective for spread spectrum hiding methods.

Shi et al. [121] proposed a blind steganalysis system, in which the statistical moments of characteristic functions of the prediction-error image, the test image, and their wavelet sub-bands were selected as features. An artificial neural network was utilized as the classifier and the model's average accuracy reached 98.7%. Authors also compared the classifier by deploying their method with a Bayes classifier and proved that the artificial neural network had better classification results.

Lie et al. [122] used two image features in order to build a blind model. Their technique is based on the analysis of two properties in the spatial and DCT domains. A non-linear neural classifier based on these two extracted features was used to achieve classification. A database composed of 2088 plain and stego images (generated by using six different embedding schemes) was used for evaluation. The proposed model managed to give 90% positive detection rate regardless the embedding technique. The embedding rate varied from 0.01–2.66 bpp depending on the steganographic scheme.

Farid and Lyu again extended [123] their model to include phase statistics in addition to first and higher order magnitude statistics, extracted from multi-scale, multi-orientation image decompositions. Experiments were conducted on a large collection of images, concerning eight different steganographic embedding algorithms and results showed that this method is reliable.

Chen et al. [124] used the projection histogram of EM to extract features composed of two parts i.e. the moments of projection histogram (PH) and the moments of the characteristic function of PH. Features were extracted also from prediction-error image in order to enhance performance. An SVM was utilized as classifier. The proposed model was tested on six (6) different steganographic schemes and the average detection rate was 98.1%.

Sun et al. in [125] introduced a universal steganalysis method based on co-occurrence matrix of differential image. They calculated the forward difference in three directions (horizontal, vertical and diagonal), towards adjacent pixels to obtain three-directional differential images for a natural image. Then they set a threshold and removed the redundant information. The co-occurrence matrices of thresholded differential images was used as features for steganalysis. An SVM with RBF kernel was used as a classifier to discriminate stego images and cover images. This method is effective in steganographic schemes applied in spatial domain.

Zhao et al. [126] proposed a steganalysis algorithm for palette-Based images. More specifically they focused on cover images of GIF format, transformed from natural images. They extracted features from generalized difference images and color correlogram. A two-class classification scheme was used to differentiate cover images and stego images, with high accuracy when the embedding rate was no less than 20%.

Zong et al. [127] proposed a blind JPEG steganalysis method based on the correlation of inter- and intra-wavelet sub-bands in the wavelet domain and feature extraction from the co-occurrence matrix. A two-order wavelet decomposition was performed, the joint probability density of each sub-band's difference coefficients (with adjoining coefficients in the horizontal, vertical, and diagonal directions) is calculated and the entropy and energy were extracted from the joint probability density matrix as features. Then, the image was decomposed into three sub-bands, and the Probability Density Function (PDF) was extracted from each sub-band's wavelet coefficient. Finally, these three kind features were combined to detect the image. An SVM was utilized as a classifier.

Ghanbari et al. [128] proposed a new algorithm for steganalysis using GLCM matrix properties (Figure 3.10). They used a combined method of steganography based on both location and conversion to hide the information in the original image and called it image-steg1 image.

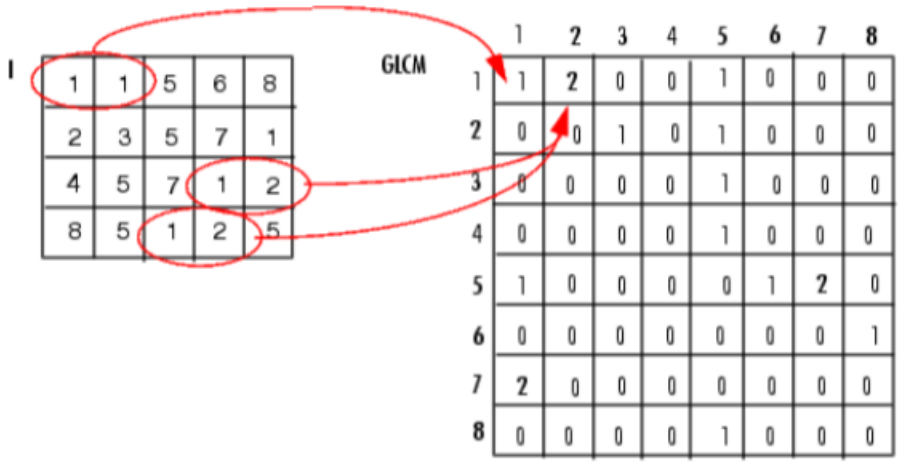

Figure 3.10: Creation of GLCM. Image (I) has 8 color levels – Source: [128].

Then, they hided the information in image-steg1 again and called it image-steg2. Using GLCM matrix properties, they discovered some different features in the GLCM of the original image and stego images. These features were extracted and used for training a multilayer perceptron (MLP) neural network.

Zhang et al. proposed [129] a universal steganalysis method for jpeg images based on sparse representation. Sparse representation, is to convey the main body of information with as little information as possible, thus simplifying the solving process of information processing. This method has high detection accuracy and overcomes the “over-fitting” problem of traditional classifiers.

Devi et al. [130] presented four different steganalysis techniques applicable when binary images (black and white) are used as a cover image. Their method improved steganalysis techniques by minimizing image-to image variations. Image-to-image variation is defined as the difference between the underlying statistic of one image and that of another. They estimated the cover image from the stego image, then they computed the difference between the two to minimize the image-to image variation and finally they extracted the feature set from this

difference. Their method can detect the stego object and estimate the length of the embedded message.

Verma [131] used a multilayer perceptron with backpropagation algorithm as a model for image classification. Moreover, he used Pre-processed Vectors Diagonal Back Propagation Algorithm (PVDBPA) to perform the operations which can detect the presence of hidden message. Furthermore, BMP steganalysis using Gray Level Co-Occurrence Matrix has been examined by using feature vectors and analyzed them through Euclidean distance which was taken as a measure.

In [132] Lu et al. proposed a steganalytic feature selection method based on the Fisher criterion, in which the separability of single-dimension and multiple dimension features, combined with measurement of the Euclidean distance, is analyzed. The proposed method has been used to analyze the features (in spatial and frequency domain) and select feature components to reduce the dimensionality. Experimental results showed that the proposed method can effectively reduce the feature dimension and also improve the steganalytic efficiency.

Tang et al. [133] proposed an adaptive steganalytic scheme based on embedding probabilities of pixels. Six different embedding rates (0.05 bpp to 0.5 bpp) to images from BOSSBase image set, were tested. Experiments evaluated on four typical adaptive steganographic methods, have shown the effectiveness of the proposed scheme, especially for low embedding rates, for example, lower than 0.20 bpp.

Qian et al. [134] were the first to introduce convolutional neural networks (CNN) in order to detect the existence of steganographic content. The proposed model (Figure 3.11) can capture the complex dependencies that are useful for steganalysis. Compared with other existing methods, this model can automatically learn feature representations with several convolutional layers. The feature extraction and classification steps are unified under a single architecture, which means the guidance of classification can be used during the feature extraction step. To evaluate the effectiveness of the developed model for

steganalysis, authors conducted experiments on three spatial domain steganographic algorithms on various payloads (0.3bpp -0.5bpp). Results compared to other state-of-the-art steganalysis methods were slightly worse.

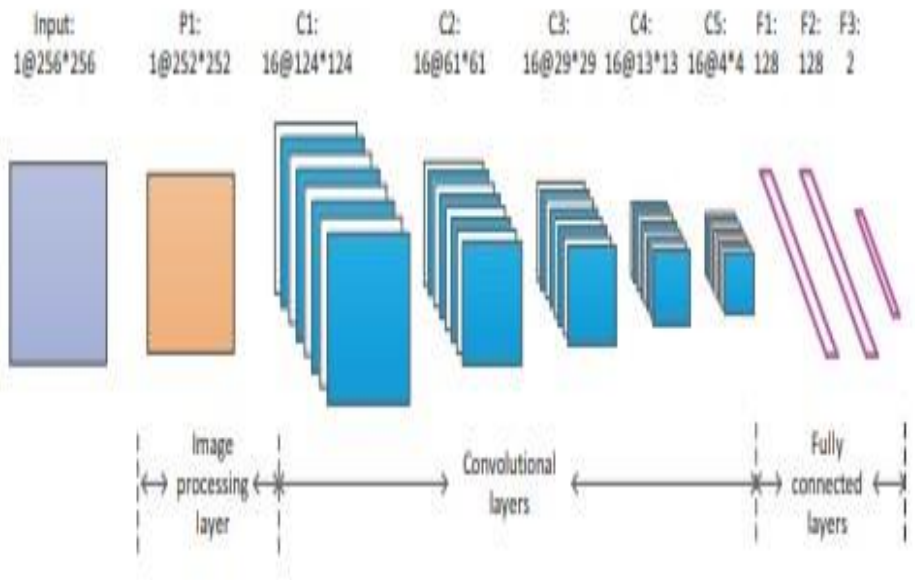

Figure 3.11: Qian's CNN Source: [134].

Desai et al. [135] developed a reduced dimensional merged feature set for universal image steganalysis using Fisher Criterion and ANOVA techniques. Features were extracted from wavelet sub-bands and binary similarity patterns extracted from DCT of an image were merged to make a combined feature set. Fisher criterion and ANOVA test were then applied to evaluate the combined feature vector score and then only those features were selected which were found sensitive in both feature selection methods. The reduced 15-dimensional feature vector was utilized to train an SVM classifier with RBF kernel. The proposed algorithm was tested against various steganography methods at different embedding rates. Stego images were generated using state of the art steganographic algorithms and two standard image databases: CorelDraw [136] and BSDS500 [137]. A 10-fold cross validation process was performed. The

proposed algorithm achieved overall 97% detection accuracy against various steganography methods.

Couchot et al. [138] proposed an architecture which embeds less convolutions, with much larger filters in the final convolutional layer. This approach is more general; therefore, it is able to deal with larger images and lower payloads. For a payload of 0.4 bpp the proposed CNN can detect stego images with an accuracy higher than 98%, whatever the steganographic algorithm chosen among three state-of-the-art, while it falls at most to 73.30% for the payload of 0.1.

Sajedi [139] proposed a method to discover special patterns that a steganography algorithm embeds in an image, the so-called Steganography Pattern Discovery (Figure 3.12) . An evolutionary method was utilized to extract the signature of stego images against clean images via fuzzy if–then rules. Then, an SVM classifier was employed to detect stego images with high accuracy. Embedding rate was 0.05bpp to 0.4bpp and the average accuracy on different steganographic methods varied from 79% to 91%.

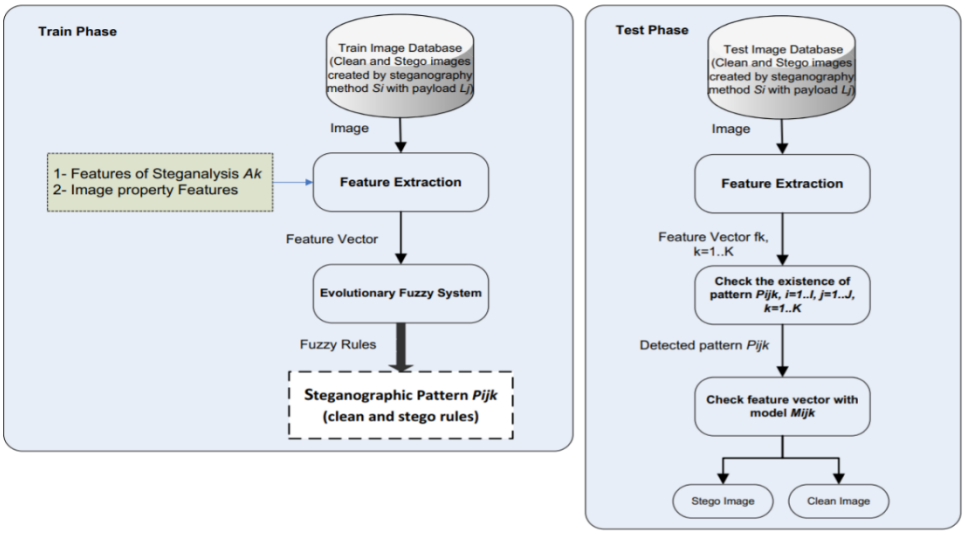

Figure 3.12: The block diagram of Steganograpy Pattern Discovery Source: [139].

Rostami et al. [140] proposed a feature selection method based on based on optimization process of Particle Swarm Optimization (PSO). In order to

improve the performance of the method, the proposed PSO is used along with the measure of Area Under the receiver operating characteristics Curve (AUC) as the fitness function. Experimental results of the proposed method on BOSSBase image set showed that even that PSO method leads to a higher feature vector, it outperforms other state of the art feature selection approaches as the classification accuracy is higher. Moreover, the embedding rate in the dataset was 0.4bpp and the classification accuracy reached 82.62% when an SVM was utilized as classifier.

Wu et al. [141] proposed a very deep CNN model, the deep residual network (DRN). DRN model usually has a large number of network layers, which proves to be effective to capture the complex statistics of digital images (Figure 3.13). Furthermore, DRN's residual learning (ResL) method actively strengthens the signal coming from secret messages, which is extremely beneficial for the discrimination between cover images and stego images.

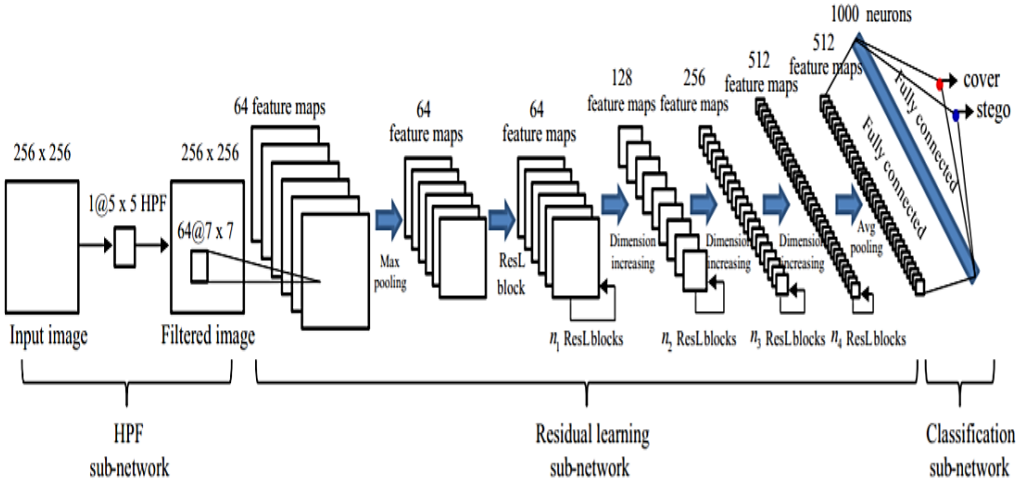

Figure 3.13: Architecture of the Deep Residual Network – Source: [141].

Experiments on BOSSBase dataset (embedding rate 0.4bpp) showed that the DRN model achieves low detection error rates – 6.48% in average - for the state of the art steganographic algorithms and outperforms the classical rich model method and several recently proposed CNN based methods.

Ye et al. [142] proposed a CNN based steganalyzer. The proposed CNN had different structure compared to the ones designed for computer vision tasks. Rather than a random strategy, the weights in first layer of the CNN are initialized with the basic high-pass filter set used in calculation of residual maps in Spatial Rich Model (SRM) [79]. Furthermore, a new activation function called truncated linear unit (TLU) was adopted in the model. Finally, the performance of the CNN based steganalyzer was boosted by incorporating the knowledge of selection channel. This approach proved capable of detecting several state-of-the-art steganographic schemes in spatial domain for a wide variety of payloads (0.05bpp – 0.5bpp) with high accuracy.

Nouri et al. [143] proposed a scheme in which the alteration of singular value curve was used to construct the steganalysis feature vector. Two spatial and JPEG based feature vectors were extracted in the proposed statistical exploitation. Experimental results on images from two datasets, embedded with relative payloads of 0.05, 0.1, 0.2 and 0.4 bpp showed the acceptable performance of the proposed feature vectors for both universal and JPEG based steganalysis methods.

Yedroudj et. al. [144] used the 30 filters of SRM [79] for the preprocessing step, a different activation function and a Batch Normalization Layer associated with a Scale Layer [145]. Furthermore, besides BOSSBase they utilized also BOWS2 database [82]. In a second experiment they virtually augmented the training set by performing the label-preserving flips and rotations and the results were satisfactory.

Finally, authors in [146] selected a manual design filter for the preprocessing layer. In addition, they integrated the knowledge of selection channel into the image preprocessing to enhance crucial residuals and initialized the network with parameters trained with a high payload rate data set to improve the performance of the network. The proposed network has fewer residual extractions and convolutional computations and therefore needs less computational resources.

3.10 Discussion

In this Chapter state-of-the-art research methods were thoroughly presented for every type of image steganalysis, according to the taxonomy presented in Section 2.2. Visual steganalysis is the simplest form of steganalysis. A Digital Forensics examiner depends solely into manual inspection of the image and tries to identify visual inconsistencies, in order to discriminate clean from stego images. This approach has good results when a) the cover medium is present and b) data were embedded into the cover medium by a poor steganography algorithm.

As more complex steganography algorithms were introduced throughout time, visual steganalysis became insufficient. On the contrary, specific steganalysis has excellent results in identifying a stego image. The major weakness of this approach is that the digital forensic examiner must have clues regarding the stegnography algorithm that was utilized. If not, the results will be be poor because the examiner would not know the appropriate software steganalysis tool to use.

The two most utilized methods of steganography concerns LSB and LSB matching steganography. The steganalysis methods proposed in the literature mainly focus on extracting relevant statistical features of the clean and stego images and utilize them to train machine learning classifiers. Although many of the aforementioned statistical methods have promising results, there are some disadvantages that should be stated:

- Datasets utilized by researchers are in general small. Their size varies from a few hundred to a few thousand images. To overcome this, many techniques were adopted like augmentation but there is always the risk of the overfitting [147].
- The proposed feature extraction methods typically output large dimensional feature vectors. Although, large dimensional feature vectors are very informative they are also more complex to train a classifier. This

leads to larger computational resources and generally larger training times.

- Furthermore, the proposed models can either identify specific steganography algorithms i.e. find the pattern for each one of them to classify images into clean and stego, or examine images of specific format. Hence, when stego images with data embedded by another steganography algorithm are examined, the proposed models should be trained again to identify these new algorithms.

Therefore, there is the need to employ new techniques that should incorporate:

- The ability to identify more complex steganography algorithms.
- The ability to be easily expanded in order to identify stego images derived from new steganography algorithms.
- To extract low dimensional feature vectors. This will decrease the computational complexity, lower significantly the training times and even when the model should be retrained, this will not be time consuming. Time waste in a digital investigation is crucial. It may lead to wrong conclusions, misjudgment of evidence and possible escape of guilt.

Models who have the aforementioned abilities should be considered as universal, which is the ultimate goal of steganalysis i.e. a model that can identify stego images regardless the embedding algorithm used. In recent times, researchers are working on this direction and employ deep learning techniques. The most utilized deep learning method for image steganalysis are the convolutional neural networks. Although this seems odd since convolutional neural networks need large training times, they are excellent feature extractors. Hence, lower - but very informative - dimensional feature vectors can be extracted and utilized to feed a classifier, which will discriminate then the images into clean and stego. This research direction was the one followed in this dissertation. A novel convolutional neural network is proposed, thoroughly

analyzed and compared against other similar CNNs and aforementioned state-of-the-art statistical feature extraction methods.

Chapter 4
Convolutional Neural Networks

4.1 Introduction

Convolutional neural networks are deep learning techniques that are widely used in conventional computer vision tasks [148] and proved to achieve state-of-the-art performance. Convolutional derives from the Latin word convolvere - “to convolve”- which means to roll together. In a mathematical perspective, a convolution is the integral measuring how much two functions overlap as one passes over the other.

Typically, Convolutional Neural Networks consist of only three types of layers: Convolution, Pooling (Max or Average) and Fully Connected. By stacking many of the prementioned layers, a convolutional neural network can be build. A typical CNN architecture is illustrated in Figure 4.1.

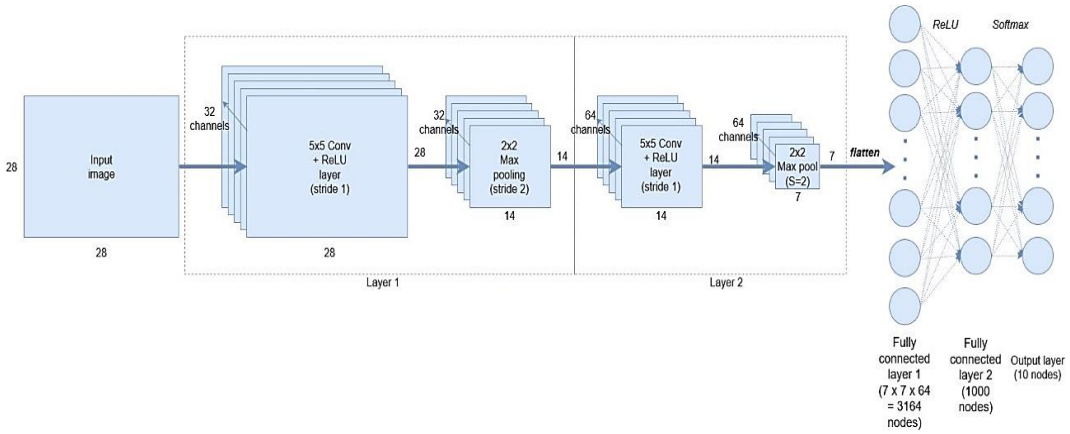

Figure 4.1: A simple convolutional neural network – Source: [149].

After presenting thousands or millions of images to the CNN, image descriptors can be learnt from layers close to the output i.e. the fully connected layers. These features can be used then to feed a classifier.

The last few years CNNs were also used in image steganalysis and compared to statistical methods showed promising results. Qian et al [134], Yang et al [150], Pibre et al [151] , Couchot [138], Bayar [152], Yedroudj et al. [144], Ye et al.[142], Xu et al. [153] and Jin et. al. [146] implemented and proposed CNNs suitable for image steganalysis. The differences to these proposals concerned network structure (i.e. the number of filters, the number of feature maps and the number of layers) and the examined steganographic algorithms. Qian et al [154] had a different approach which is similar with ours but in a very different manner. More specifically, they trained a CNN with stego images of high embedding rates and then they used the extracted features from various layers, to classify stego images with lower embedding rate.

In this dissertation a novel convolutional neural network for steganalysis is designed and implemented. The utilized layers are briefly described below.

4.2 Convolutional Layer

The principal purpose of a convolutional layer is to extract features from the input layer. The convolutional layer captures the spatial relationship between image pixels by learning its features utilizing small amount of input data. This is done by convolving the input image with a filter. Let I be an image which is fed to the input layer of the Convolutional Neural Network and F a fixed sized filter (*nxm*), which convolves with the image. The filtered image I_f is then determined as in equation 2:

$$I_f = I*F \tag{2}$$

where * indicates a 2D convolution.

A filter F is a matrix which slides over the image and convolves in parallel to a region of the input image. Large sized filters capture more features than smaller sized filters. The output of a convolutional layer is also known as feature map. Afterwards, the next convolutional layer similarly extracts features from the feature maps previously learned by a former convolutional layer. The output of these hierarchical feature extractors is feeding a fully connected neural network that performs the classification task and discriminates clean from stego images.

The information a feature map has after each convolution, relies on parameters like stride and padding. Stride (S) is the size of the step the convolution filter moves along the input image, while padding (P) pads the image volume with zeros around the border. Stride size is usually 1, meaning that the filter F slides pixel by pixel. By increasing the stride size, the filter slides over the input with larger intervals and thus has less overlap between the cells. As the filter slides over the input image the sum of the convolution goes into the feature map. In a convolutional neural network layer, different filters can be utilized for each one of the convolutional layers. In steganalysis, typical filter sizes are 3x3 and 5x5, but there are also some implementations with larger filters as well.

Let $I\,(w_1 x h_1 x d_1)$ an image fed into a convolutional neural layer where w_1 is the width, h_1 the height and d_1 the depth (number of channels). The output volume produced by the convolutional layer will be an image $I_{conv}\,(w_2 x h_2 x d_2)$ where d_2 is the number of filters also denoted as k. Its parameters are calculated as in equations 3-4.

$$w_2 = (w_1 - F + 2P)/S + 1 \quad (3)$$

$$h_2 = (h_1 - F + 2P)/S + 1 \quad (4)$$

The total number of learnable parameters after each convolutional layer is shown in equation 5.

$$parameters = ((m \times n) + 1) \times k \quad (5)$$

where m is the width of the filter and n is the height of the filter.

4.3 Dilated Convolutions

In a typical convolutional layer when more and larger feature maps are needed, larger filters are utilized, but this inevitably leads to an exponential growth of parameters that need to be calculated. Therefore, the computational needs are large. A dilated convolutional layer [155] on the other hand, performs convolution operations with modified filters. The layer expands the filters by inserting zeros between each filter element according to a dilation factor. The dilation factor determines the increase to the field of view of the filter. Dilated convolution is then defined as in equation 6.

$$I_f = I *_i F \tag{6}$$

where *i is the dilation factor.*

The main advantages the dilated convolutional layer has are the larger receptive field, the more efficient computation, the reduced memory consumption and the faster convergence of the network. In dilated convolutions, a $k \, x \, k$ sized filter, with a dilation factor r is enlarged to $k + (k - 1)x(r - 1)$ [156]. Due to these advantages, in the proposed convolutional neural network dilated convolutional layers are utilized, where the filter sizes are 7x7 and 5x5 with a dilation factor of 3. These values were emerged from extreme experimentation as described in Chapter 5.

4.4 Pooling Layer

A pooling layer divides the input information into regions and then computes the average or maximum values of each region. Therefore, an Average Polling Layer or a Max Pooling Layer can be exploited. Pooling layers are used to reduce the size of the inputs, reduce the number of parameters and hence speed up the computation.

Typically, CNNs use pooling layers with filter size 2x2, applied with a stride of 2. This filter downsamples every depth slice in the input by 2 along both width and height, discarding 75% of the activations. Let V_1 ($w_1 x h_1 x d_1$) an input volume

to the pooling layer. After performing pooling operation, an output volume V_2 $(w_2xh_2xd_2)$ with $d2=d1$ is produced, where its parameters are calculated as in equations 7-8.

$$w_2 = (w_1\text{-}F)/S+1 \quad (7)$$

$$h_2 = (h_1\text{-}F)/S+1 \quad (8)$$

4.5 Batch Normalization Layer

A batch normalization (BN) layer [157] is used to speed training. It normalizes each input channel across a mini batch. The batch normalization layer subtracts the mini-batch mean and divides by the mini-batch standard deviation. Afterwards, the layer shifts the input by an offset and scales it by a scale factor. Let $B=\{x_{1...m}\}$ a mini batch, μ_B the mini batch mean, σ_B^2 the mini batch variance, $\hat{x}_i$ the normalized values, ε is a constant for numerical stability, γ is the scale factor , β is the offset and y_i the linear transformations of the input (i.e.the output) which are calculated as shown in equations 9-12.

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i \quad (9)$$

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2 \quad (10)$$

$$\hat{x}_I = \frac{\hat{x}i-\mu_B}{\sqrt{\sigma_B^2+\varepsilon}} \quad (11)$$

$$y_i = \gamma\hat{x}_i + \beta \quad (12)$$

Parameters γ and β are learnable parameters that are updated during network training. In the proposed convolutional neural network, a batch normalization layer was used after each convolutional layer.

4.6 Dropout Layer

Dropout [158], is a technique used to improve or avoid the phenomenon of overfitting on neural networks. The specific layer during the training phase deactivates a percentage of neurons. This technique improves generalization because it forces the layer to activate different neurons each time to learn the same thing.

It must be noted that during the test phase the dropout is deactivated. In the proposed network one dropout layer was used with probability 0.5.

4.7 Activation Function - Leaky ReLU Layer

The most common used activation function in a convolutional neural network is the Rectified Linear Unit (ReLU). A ReLU activation function (equation 13) sets all negative inputs to zero.

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (13)$$

The main disadvantage of the ReLU activation function is the so called “dying’’ problem. A ReLU neuron is called “dead” if the output of the ReLU function for any given input is zero, i.e. the input to the activation function is less than zero. When a neuron gets negative and because the slope of ReLU in the negative range is also 0, the neuron will not recover. Therefore, the specific neuron cannot discriminate the input and becomes impractical. In this way and after training for a large number of epochs, a large part of the network may become absolutely useless.

A variation to ReLU is the Leaky ReLU activation function. A leaky ReLU layer also performs a threshold operation, but when an input value is less than zero it multiplies it by a fixed scalar (equation 14).

$$f(x) = \begin{cases} x, & x \geq 0 \\ a * x, & x < 0 \end{cases} \quad (14)$$

where α is a fixed scalar. A common value of α is 0.01. In the conducted experiments (presented in Chapter 6) the ‘dying’ problem was confirmed. Thus, Leaky ReLU was used as the activation function of the proposed convolutional neural network.

4.8 Fully Connected Layer

A fully connected layer is a layer in which its neurons are fully connected with the activations in the previous layer. Typically, there is one fully connected layer (at the end of the CNN) and the number of its neurons equals to the number of the classes. There are also many proposed architectures - depending on the classification task - that embed two or more fully connected layers. In the proposed method two fully connected layers are utilized.

Chapter 5
KarNet – A novel CNN for image steganalysis

5.1 Introduction

Embedded secret information to an image is considered as additive noise. Thus, a common strategy in steganalysis, is to enhance this noise and therefore increase signal-to-noise ratio between the weak stego signal (stego images only) and the image signal. Hence, a high-pass filter is applied to every image coming to the input layer of the convolutional neural network., This step makes CNNs to converge sooner and to capture better the discrepancies between a stego and a clean image. A high pass filter F (equation 15) is applied to every image of the dataset as a preprocessing step prior to presenting the image to the input layer of the CNN.

Another similar approach was proposed by Kim et al. in [159]. Authors suggested a binarized differential filter (BDF - equation 16) and made experiments with both BDF and the high-pass filter F. In this dissertation the high pass filter F was utilized as a preprocessing step. Therefore, every grayscale image sized 256x256x1 is filtered and the resulted size of the image which is presented to the input layer of the CNN is 252x252x1.

$$F = \frac{1}{12}\begin{pmatrix} -1 & 2 & -2 & 2 & -1 \\ 2 & -6 & 8 & -6 & 2 \\ -2 & 8 & -12 & 8 & -2 \\ 2 & -6 & 8 & -6 & 2 \\ -1 & 2 & -2 & 2 & -1 \end{pmatrix} \quad (15)$$

$$BDF = \begin{cases} 1, & f(x-1,y) \leq f(x,y) \\ -1, & otherwise \end{cases} \quad (16)$$

Each one of the images entering the input layer of our proposed convolutional neural network utilizes the filter described in equation 15. Other state-of-the-art convolutional neural networks [144] for spatial image steganalysis make use of the 30-basic high-pass filters used in the computation of residual maps from SRM method.

5.2 The examined architectures

In order to propose a novel convolutional neural network and to examine its discriminative capability as: i) a classifier and ii) a feature extractor, numerous experiments were conducted. These experiments concerned different architectures where the basic network elements that changed in each network were:

- the preprocessing steps i.e. the utilization of filter F or BDF,
- the number of layers,
- the type of layers (convolutional, batch normalization, dropout, pooling type, number of fully connected etc.),
- each layer settings (stride, filter size, dilation factor, padding etc.),
- number of neurons in the first fully connected layer,
- the hyperparameters settings (learning rate, learning decay, optimizer, etc.),
- the convergence times.

The criterion to keep or to change the examined architecture was the detected accuracy with respect to convergence time, especially for low embedding rates. The conducted experiments can be summarized in Table 5.1, while in Figure 5.1 the detected accuracy from each one of the examined convolutional neural networks is shown.

Table 5.1: The examined architectures.

ID	Number of layers	Dilation	Dilation Factor	Accuracy on validation set (%) – S-UNIWARD
1	14	Yes	2	72.79
2	14	Yes	2	74.01
3	14	Yes	2	74.29
4	14	Yes	3	78.05
5	17	Yes	3	76.62
6	17	Yes	3	76.24
7	20	Yes	3	73.95
8	17	Yes	3	75.69
9	17	Yes	3	73.04
10	17	Yes	3	74.38
11	17	Yes	3	77.45
12	17	Yes	3	78.91
13	17	Yes	3	78.97
14	17	Yes	3	79.40
15	17	Yes	4	71.29
16	17	Yes	4	72.37
17	17	Yes	4	72.04
18	17	Yes	2	75.81
19	17	Yes	2	75.79
20	17	Yes	2	76.41
21	17	Yes	2	75.83
22	17	Yes	2	75.38
23	17	Yes	2	75.79
24	20	Yes	2	75.79

ID	Number of layers	Dilation	Dilation Factor	Accuracy on validation set (%) – S-UNIWARD
25	19	Yes	2	78.29
26	19	Yes	2	71.56
27	19	Yes	2	72.64
28	19	Yes	2	76.00
29	19	Yes	2	78.15
30	***17***	***Yes***	***3***	***81.06***

It must be noted that Table 5.1 summarizes only the most important elements that changed to each network and not all of them. The number of the neurons in the first fully connected layer for all the experiments, were initially set to 1500 but in the final architecture of the proposed convolutional neural network this number was set to 250.

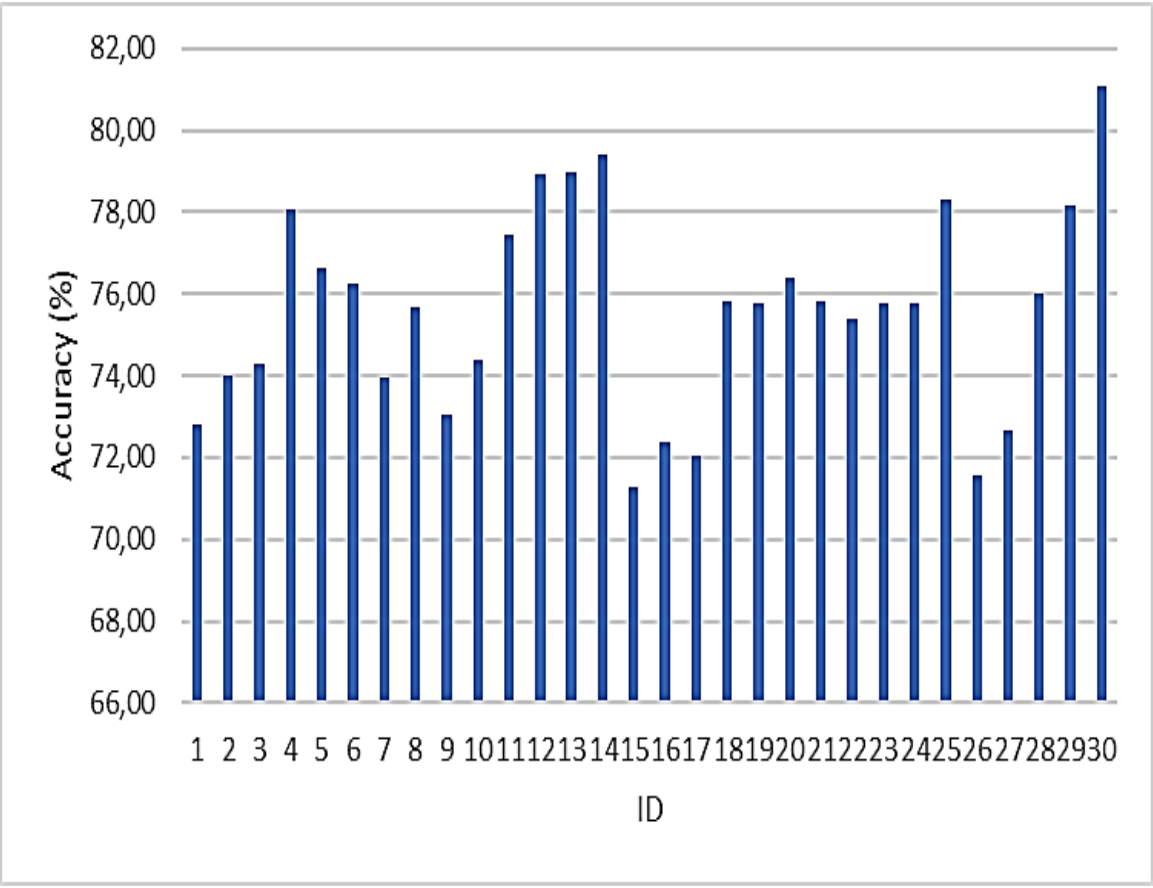

Figure 5.1: Accuracy on validation set of the examined CNN architectures.

Along with the above examined architectures, other proposals found in literature such as those in [134],[151] (Figures 5.2&5.3) were also examined and compared against each network.

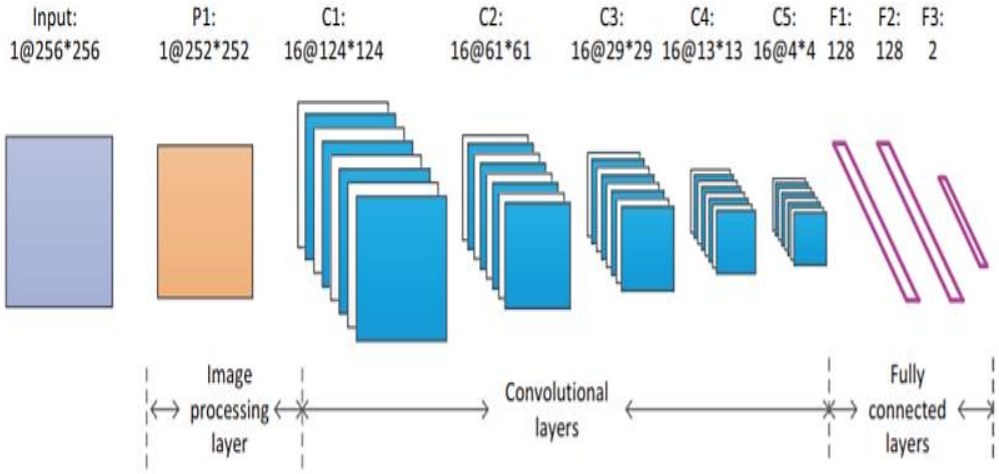

Figure 5.2: CNN in [134].

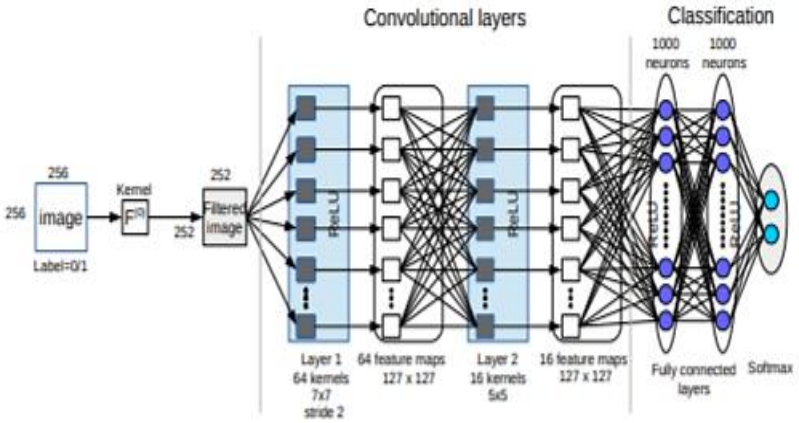

Figure 5.3: CNN in [151].

After deciding the network architecture, more experiments were conducted concerning the hyperparameters of the convolutional neural network i.e. the optimizer, the learning rate, learning decay, the validation patience etc.

5.3 The proposed architecture – KarNet

The following novel convolutional neural network - also known as the KarNet - shown in Figure 5.4 is proposed, which will be:

a) first evaluated as a novel dilated convolutional neural network for spatial image steganalysis and

b) utilized strictly as a feature extractor.

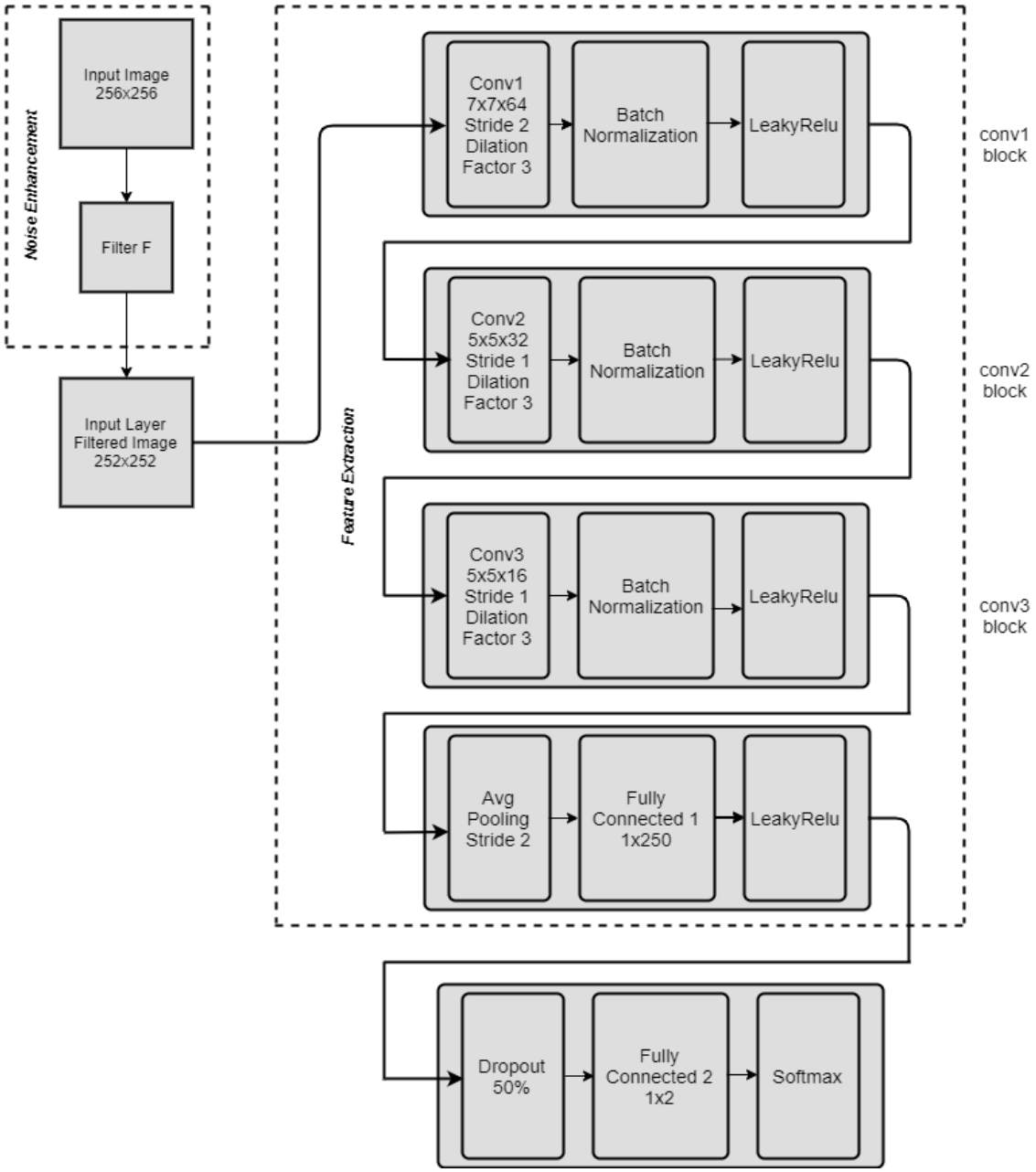

Figure 5.4: KarNet – The proposed CNN.

Initially, each one of the images of the training set is filtered (noise enhancement block in Figure 5.4) with the filter *F* described in equation 15. As a result, a filtered image sized 252x252x1 enters the first convolutional block. Three (3) convolutional blocks i.e. conv1, conv2, conv3 block were used as illustrated in Figure 5.4.

A conv block is defined as the utilization of:

- a dilated convolutional layer
- a batch normalization layer and
- a Leaky ReLU layer.

Moreover, dilated convolutional layers were used to increase the receptive field without increasing the number of learnable parameters. Thus, less computational resources were needed, and this led to faster training of the network. An example of dilated convolutions is shown in Figure 5.5.

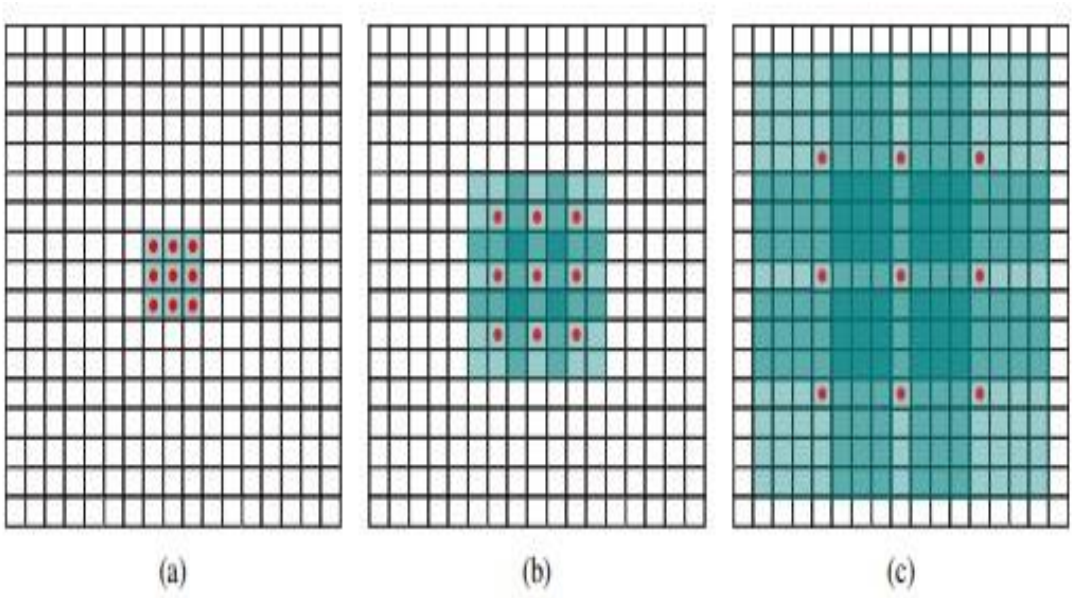

Figure 5.5: (a) 1-dilated convolution b) 2-dilated convolution c) 4-dilated convolution. Source : [155].

The dilation factor in the proposed architecture is three (3), meaning that the first filter sized 7x7 corresponds to an effective filter size of 19x19, while the second filter sized 5x5 corresponds to an effective filter size of 13x13.

The batch normalization layer was used to speed training and reduce the sensitivity to network initialization. KarNet uses three batch normalization layers, each one after the respective dilated convolutional layer. Layer weights are initialized by sampling from a normal distribution with zero mean and variance 0.01.

The activation function was the Leaky ReLU, to avoid and overcome the 'dying neurons' problem. Experiments that were also made in the beginning with ReLU as an activation function, confirmed the problem and highlighted the utilization of Leaky ReLU.

Additionally, an average pooling layer was used. Average pooling was chosen against max pooling, to consider all the activations in the pooling region. A stego signal is very weak and the modifications made to the image after embedding the secret message slightly alter pixels value. Therefore, the more information kept the better classification results can be achieved. Average pooling retains more information than max pooling, although in some scenarios max pooling may perform better. In our experiments both max and average pooling were examined but average pooling had better behavior.

Finally, a typical softmax layer followed by a classification layer was utilized to perform classification of the input images into clean or stego.

5.4 Network's training parameters

As described in Section 5.1 each one of the images in the dataset was filtered with the filter F shown in equation 15. Thus, the input layer of the CNN was fed with an 252x252x1 image.

Afterwards, each image moved from the input layer to the convolutional blocks described earlier. Each image was convolved with the respective filter in

each convolutional block and produced the respective feature maps. KarNet was set to be trained either for a maximum of 5000 epochs and early stopping with validation patience of 10 which means that if the validation loss was larger or equal to the previously smallest loss for 10 consecutive epochs, the network stopped its training. Typical adopted values of validation patience for CNN training are 2 [160], 3 [161], [162] or 5 [163]–[165] . In order to be more certain that our convolutional neural network is robust against overfitting, validation patience was set to 10.

The stochastic gradient descent with momentum (SGDM) optimizer was utilized and the momentum was set to 0.9. The learning rate was 0.0005 and a batch size of 256 (128 cover/stego pairs) was used. All the experiments were conducted into a workstation with two XEON processors (56 cores), 64GB of RAM and two Nvidia GTX 1080ti GPUs (3584 CUDA cores each) working in parallel. The average training time for all examined architectures was about 10 hours whilst there were architectures (the ones tried to classify images with the lowest embedding rate) which needed much more training time.

5.5 Determining the number of neurons in fully connected layer

Two fully connected layers were used. The first one consists of 250 neurons and the second one of two neurons (the number of classes i.e. clean / stego). The reason was to capture global image features and not local like convolutional layers that operate on a window of certain size. The first fully connected layer can model more complex global patterns.

This section concerns the conducted experiments to determine the optimal number of neurons in the first fully connected layer in respect to ratio accuracy/complexity. Moreover, performance of KarNet was examined for having 20,100,250,500,750,1000,1250 and 1500 neurons in the first fully connected layer, for both steganographic algorithms and all embedding rates.

Furthermore, for each examined number of neurons the feature vector was also extracted and trained the Random Forest classifier. The results obtained for KarNet are summarized in Tables 5.2&5.4 while Tables 5.3&5.5 show the respective results for the Random Forest classifier, for both steganographic algorithms and all embedding rates.

Table 5.2: KarNet accuracy - S-UNIWARD.

Embedding rate	**Number of neurons in first fully connected layer**						
(bpp)	**20**	**250**	**500**	**750**	**1000**	**1250**	**1500**
0.4	86.55	86.95	86.00	84.55	85.85	85.65	86.70
0.3	83.80	82.60	82.65	83.25	82.35	80.80	82.60
0.2	76.30	76.55	77.20	76.35	76.10	74.85	76.65
0.1	63.80	66.30	67.75	66.50	65.65	65.30	65.95

Table 5.3: Random Forest accuracy - S-UNIWARD.

Embedding rate	**Number of neurons**						
(bpp)	**20**	**250**	**500**	**750**	**1000**	**1250**	**1500**
0.4	86.45	85.85	86.35	86.00	85.90	86.50	86.95
0.3	83.00	82.70	82.35	82.90	81.95	81.30	81.65
0.2	76.65	76.35	77.25	76.40	76.00	75.95	75.30
0.1	65.15	66.25	67.20	67.75	65.90	66.85	66.80

Table 5.4: KarNet accuracy – WOW.

Embedding rate	**Number of neurons in first fully connected layer**						
(bpp)	**20**	**250**	**500**	**750**	**1000**	**1250**	**1500**
0.4	84.60	85.10	85.25	85.30	85.45	83.55	81.50
0.3	80.85	79.80	79.30	80.50	79.75	80.25	79.05
0.2	70.25	72.35	72.25	72.50	73.40	73.10	70.90
0.1	58.70	61.50	60.85	59.70	60.20	59.25	60.90

Table 5.5: Random Forest accuracy – WOW.

Embedding rate (bpp)	Number of neurons						
	20	**250**	**500**	**750**	**1000**	**1250**	**1500**
0.4	84.70	84.80	85.90	85.65	84.55	84.45	84.55
0.3	79.75	79.25	79.70	80.65	79.90	79.30	79.40
0.2	72.45	73.10	72.75	72.40	73.70	72.60	71.45
0.1	59.35	62.30	61.25	59.85	60.75	58.80	60.20

The criterions to choose the “best” number of neurons were to achieve:

a) high accuracy in relation to low computational complexity
b) very good network performance to low embedding rates and
c) similar or better results for Random Forest classifier.

Comparing the above tables, it is obvious that for S-UNIWARD steganographic algorithm, the best average behavior is achieved when selecting 500 neurons in the first fully connected layer while for the WOW algorithm this number is 250. The number of the learnable parameters for each one of the examined networks is shown in Table 5.6

Table 5.6: Number of parameters per network.

Number of neurons in Fully Connected 1 layer	Number of parameters
20	1,337614 x 10^6
250	15,944224 x 10^6
500	31,820974 x 10^6
750	47,697724 x 10^6
1000	63,574474 x 10^6
1250	79,451224 x 10^6
1500	95,327974 x 10^6

From Table 5.6 it is noticed that if the number of neurons in the first fully connected layer is set to 250, a 50% decrease in the number of learnable parameters is achieved, compared to the ones if 500 neurons were chosen. Moreover, the results for both methods (KarNet – Random Forest) and for both number of neurons (250 – 500) are nearly the same. Therefore, the number of neurons in the first fully connected layer of KarNet was set to 250 due to lowest computational complexity.

5.6 Learnable parameters

The activations and the number of learnable parameters (per layer/total network) of the proposed convolutional neural network were calculated and are presented in Table 5.7.

Table 5.7: Learnable paramaters of KarNet.

Layer	Activations	Layer Parameters Analysis	Total Layer Parameters	Total Network Parameters
Input	252x252x1	0	0	0
Conv1 – 64 7x7x1 convolutions, Stride 2, Dilation Factor 3x3	126x126x64	Weights 7x7x1x64 Bias 1x1x1x64	3200	3200
Batch Normalization	126x126x64	Offset 1x1x64 Scale 1x1x64	128	3328
Leaky Relu – scale 0.01	126x126x64	0	0	0
Conv2 – 32 5x5x64 convolutions, Stride 1, Dilation Factor 3x3	126x126x32	Weights 5x5x64x32 Bias 1x1x1x32	51232	54560
Batch Normalization	126x126x32	Offset 1x1x32 Scale 1x1x32	64	54624
Leaky Relu – scale 0.01	126x126x32	0	0	54624
Conv3 – 16 5x5x32 convolutions, Stride 1, Dilation Factor 3x3	126x126x16	Weights 5x5x32x16 Bias 1x1x1x16	12816	67440

Batch Normalization	126x126x16	Offset 1x1x16 Scale 1x1x16	32	67472
Leaky Relu – scale 0.01	126x126x16	0	0	67472
Average Pooling, Stride 2x2	63x63x16	0	0	67472
FC1, fully connected with 250 neurons	1x1x250	Weights 250x63504 Bias 250x1	15876250	15943722
Leaky Relu – scale 0.01	1x1x250	0	0	15943722
Dropout – 50%	1x1x250	0	0	15943722
FC2, fully connected with 2 neurons	1x1x2	Weights 2x250 Bias 2x1	502	15944224
Softmax	1x1x2	0	0	15944224
Classification	1x1x2	0	0	15944224

Therefore, the number of learnable parameters of the proposed network was $15{,}944224*10^6$.

5.7 Proposed architecture's differences with other state-of-the-art networks

Apart from the design and implementation of KarNet the differences between the proposed CNN with other state of-the-art CNNs utilized for spatial image steganalysis like IAS-CNN [146], Ye-Net [142], and Yedrouj-Net [144] are examined. Each architecture of the aforementioned convolutional neural networks is shown in Figures 5.6-5.8.

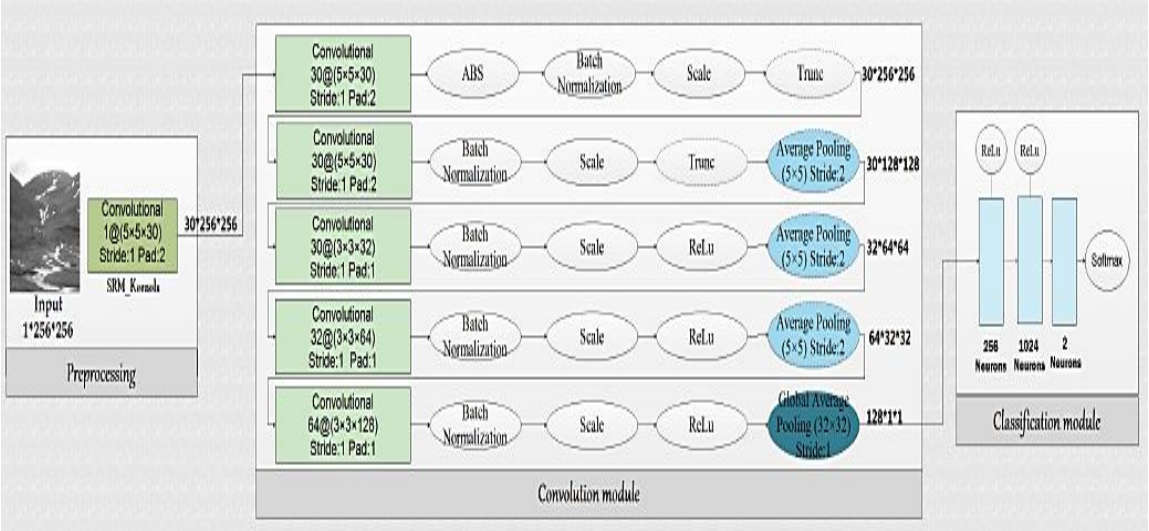

Figure 5.6: Yedrouj-Net architecture – Source: [144].

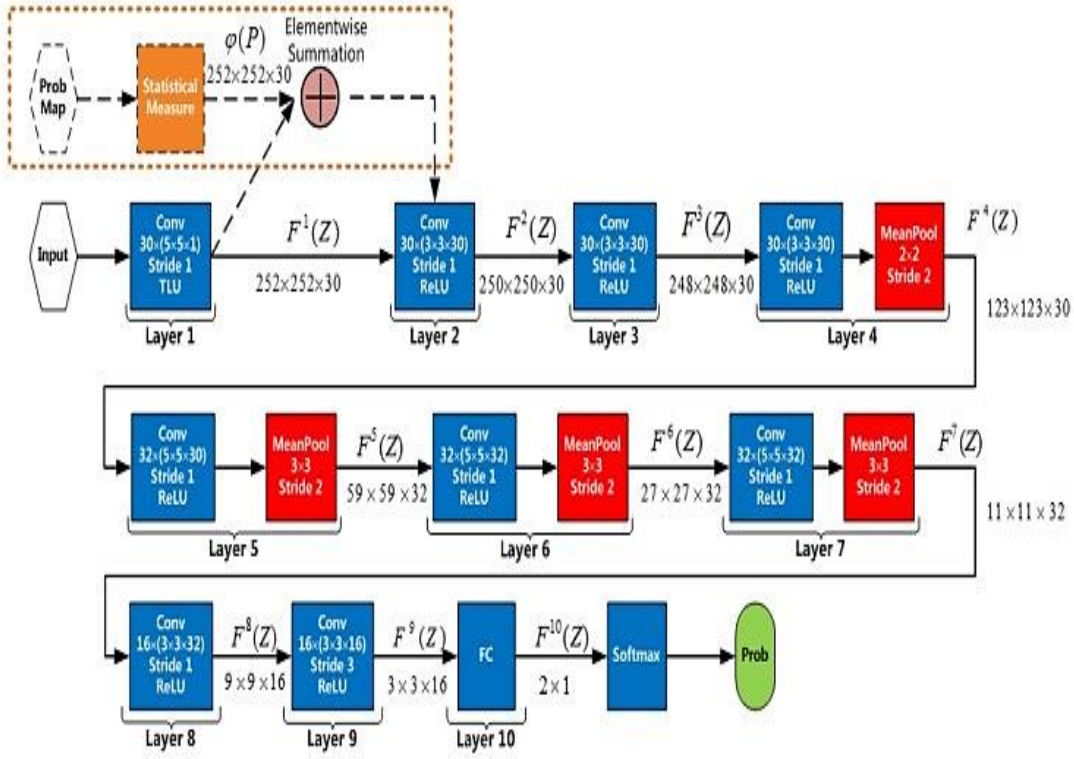

Figure 5.7: Ye-Net architecture– Source: [142].

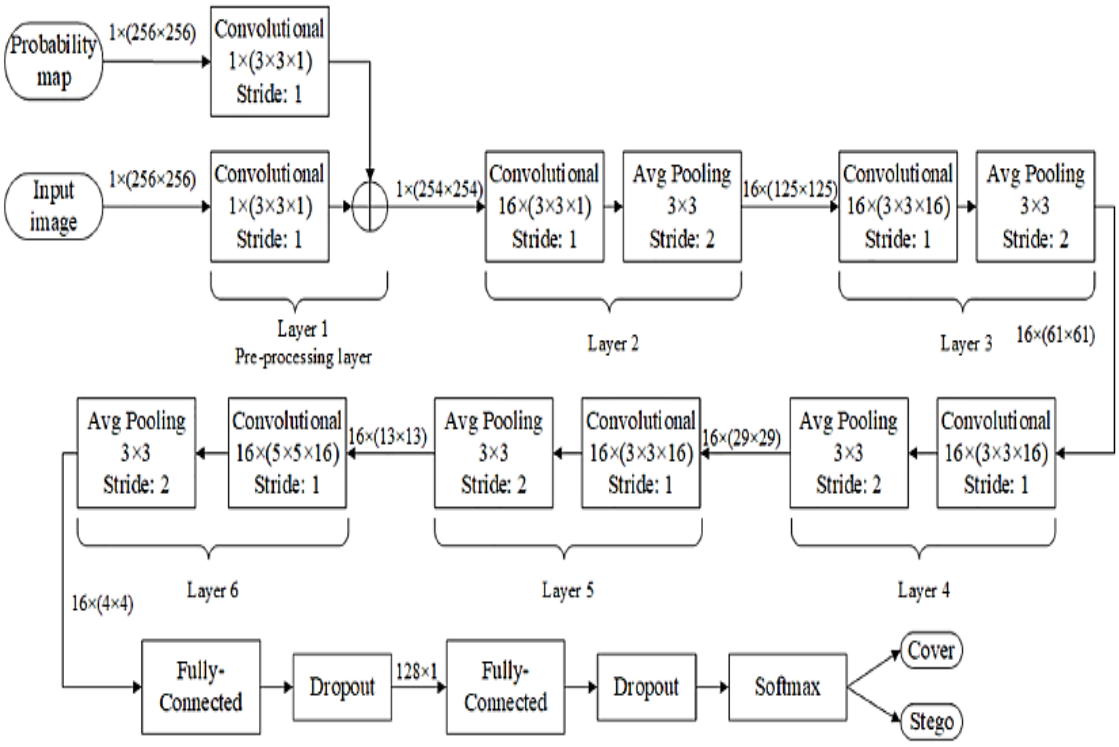

Figure 5.8: IAS-CNN architecture - Source:[146].

Table 5.8 summarizes the most important building blocks of each state-of-the-art- convolutional neural networks used for spatial image staganalysis and highlights the differences to our proposed CNN, the KarNet.

Comparing the state-of-the-art CNNs with KarNet several similarities and differences can be distinguished:

- KarNet uses the high pass filter F described in equation 15. The other networks use one (IAS-CNN) or the 30 basic filters described in SRM method either for preprocessing input images (Yedroudj-Net) or to initialize the weights of the first layer (Ye-Net).
- All proposed convolutional neural networks utilize images sized 256x256.
- KarNet uses less convolutional layers than the others and moreover it is the only CNN that uses dilated convolutions.
- KarNet and IAS-CNN use dropout layer to prevent overfitting. The rest examined architectures do not.

- Average pooling is utilized except Ye-Net which uses mean pooling.
- A significant difference between the networks is the activation function. KarNet is the only one that uses LeakyReLU.
- There is a plethora of different number of fully connected layers. KarNet and IAS-CNN use two (2).

Table 5.8: Basic building blocks of KarNet and state-of-the-art CNNs.

	Convolutional Neural Network			
Features	**KarNet**	**Yedroudj-Net**	**Ye-Net**	**IAS-CNN**
Number of filters for preprocessing	1	30	1	1
Input image (prior to preprocessing)	256x256	256x256	256x256	256x256
Number of conv layers	3	5	9	5
Dilated convolutions	Yes	No	No	No
Batch Normalization layer	Yes	Yes	No	No
Absolute Value layer	No	Yes	No	No
Pooling	Average	Average -Global Average	Mean	Average
Activation Function	LeakyReLU	Truncation-ReLU	TLU - ReLU	ReLU
Dropout layer	Yes	No	No	Yes
Number of fully connected layers	2	3	1	2

5.8 The dataset

The literature review in Chapter 3, besides the presentation of state-of-the-art methods in every domain of steganalysis also gave us useful information regarding the datasets (public or proprietary) that other researchers used. This information is presented in Table 5.9. The links where each one of these datasets can be found online are given in Appendix (Table I.6).

Table 5.9: Datasets and Number of papers that they were used.

Image dataset	Number of papers found	Publicly available found
BOSSBase	22	Yes
Corel	8	Yes
NRCS	7	Yes
UCID	5	Yes
USC	5	Yes
BOWS – BOWS2	4	Yes
Philip Greenspun	4	Yes
BSDS	2	Yes
CBIR	2	Yes
Kodak	2	Yes

Observing Table 5.9 some useful conclusions can be obtained:

- ✓ The majority of the authors chose to use publicly available datasets as benchmark. This makes the comparison between similar methods easier and the reader of their articles can determine the added value that the proposed method contributes in steganalysis.
- ✓ BOSSBase [72] is by far the most utilized dataset.

In order our results to be directly comparable with the ones to the literature, BOSSBase v1.01 was used as the main dataset. BOSSBase v1.01 contains 10000 grayscale images of pgm format sized 512x512 (Figure 5.9).

These images were split into four (4) equal parts and thus 40000 images sized 256x256 were extracted (Figure 5.10). It must be stated here that images were split the images prior to embedding, to avoid manipulation of the classifiers results.

Figure 5.9: Sample cover images from BOSSBase dataset.

Figure 5.10: A cover image split into four equal parts.

Two different adaptive steganographic algorithms were examined by their Matlab implementations [166], i.e. Spatial-Universal Wavelet Relative Distortion (S-UNIWARD) [17] and Wavelet Obtained Weights (WOW) [18]. These two steganographic algorithms first estimate the distortion caused in an image when a message is embedded, and afterwards embed a small quantity of the embedded message in image regions where the distortion was found small. This embedding procedure makes statistical steganalysis more difficult because the alterations in the statistical features of the images produced by the embedded message are very small.

In total, 40000 stego images were produced with different embedding rates (0.1-0.2-0.3-0.4bpp) and an overall of 80000 clean and stego images were in each one of the four datasets (one per embedding rate).

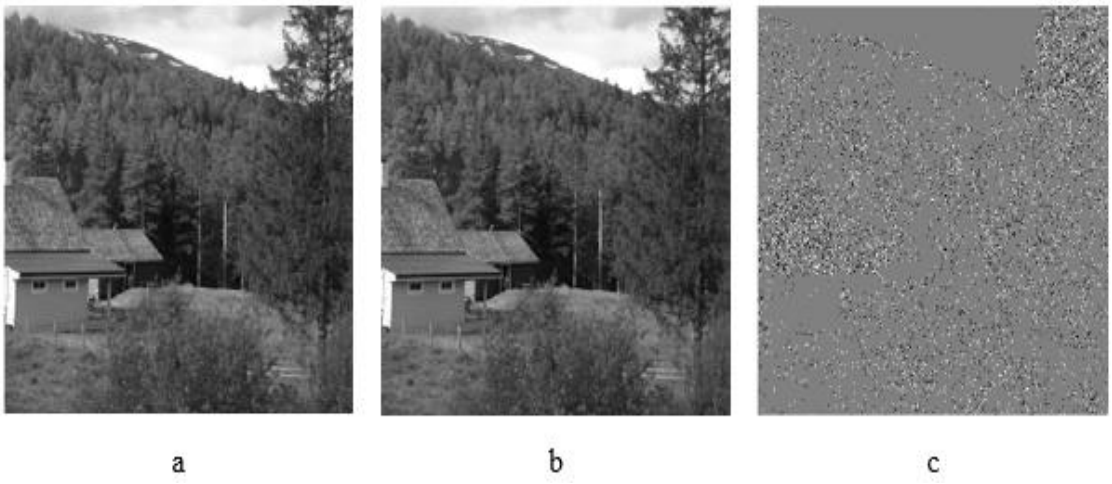

Figure 5.11: a) Cover image b) Image after applying the S-UNIWARD algorithm c) The distortion the stego algorithm resulted +1=white -1=black.

Figures 5.11&5.12 show an example of each steganographic algorithm (S-UNIWARD & WOW respectively) applied to the same image along with the resulted distortion, while Figure 5.13 shows the LSB plane for both steganographic algorithms.

a b c

Figure 5.12: a) Cover image b) Stego image after applying the WOW algorithm c) The distortion the stego algorithm resulted. +1=white -1=black.

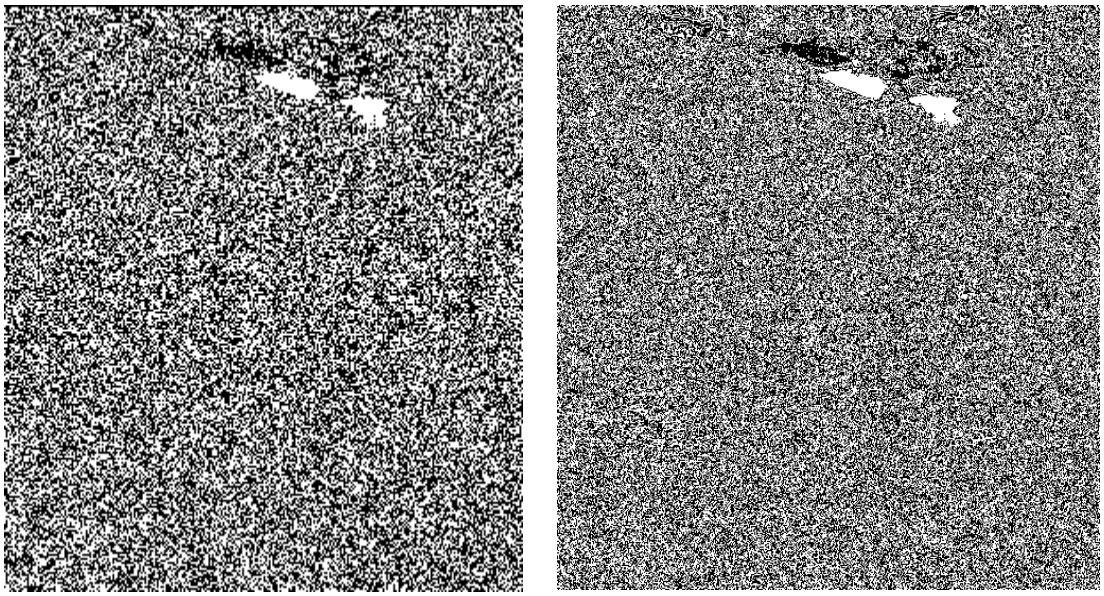

Figure 5.13: LSB plane of stego image: left) S-UNIWARD right) WOW.

In order to see if our proposed convolutional neural network generalizes well, 2000 of these images (1000 clean and 1000 stego) were never presented to KarNet and were used as a test set to evaluate our method's performance. The rest of the dataset - i.e.78000 images - was then split into a training set and a validation set (90%-10% respectively) as shown in Table 5.10.

Table 5.10: Dataset split.

Number of images			
Set	**Clean**	**Stego**	**Total**
Training	35100	35100	70200
Validation	3900	3900	7800
Test	1000	1000	2000

5.9 Metrics used

Although the dataset was perfectly balanced, in order to evaluate better the proposed method, other metrics such as Precision, Recall, F1 score and the Receiver Operating Characteristic (ROC) Area were also used. Prior giving the definitions for these metrics, other terms such as True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN) must be defined.

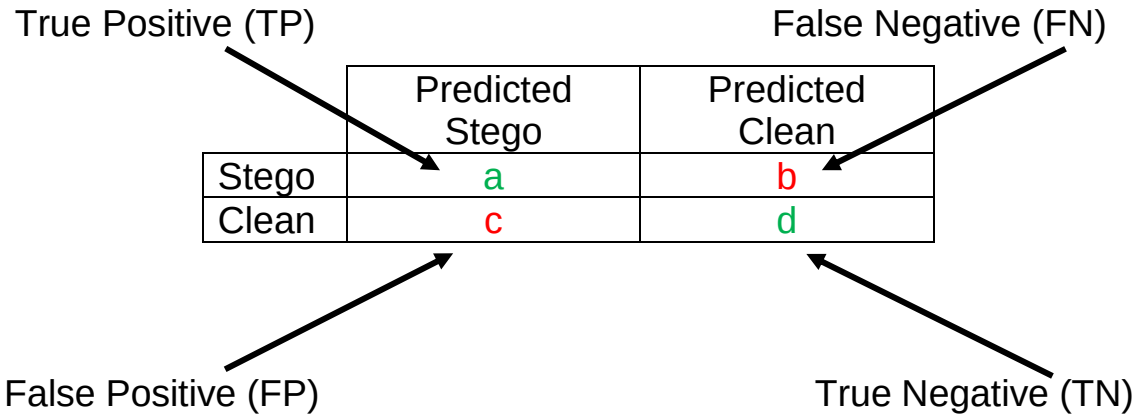

Figure 5.14: Basic statistical terms.

Figure 5.14 shows a confusion matrix according to our binary (stego and clean images) classification problem indicating each one of aforementioned terms. Let a, b, c & d be the predictions of our classifier, where a & d represent correct predictions and b & c false predictions.

A true positive occurs when the model correctly predicts the positive class, i.e. the stego image. Likewise, a true negative occurs when the model correctly predicts the negative class i.e. the clean image. A false

positive occurs when the model incorrectly predicts the positive class, i.e. the stego image. Likewise, a false negative occurs when the model incorrectly predicts the negative class, i.e. the clean image.

True Positive rate (TPR) is also called sensitivity, while the True Negative Rate (TNR) is also called specificity. Sensitivity and specificity are inversely proportional, meaning that as the sensitivity increases, the specificity decreases and vice versa. Equations 17&18 show the formulas for both *Precision* & *Recall* respectively.

$$Precision = \frac{TP}{TP+FP} = \frac{correctly\ predicted\ stego}{correctly\ predicted\ stego + false\ predicted\ stego} \quad (17)$$

$$Recall = \frac{TP}{TP+FN} = \frac{correctly\ predicted\ stego}{correctly\ predicted\ stego + false\ predicted\ clean} \quad (18)$$

Likewise, in equation 19 the formula for *F1* score is given, which actually represents the harmonic mean of precision and recall.

$$F1 = 2 * \frac{precision * recall}{precision + recall} \quad (19)$$

The Receiver Operating Curve (ROC) can show whether a classifier is performing well in general. They give the same result regardless of what the class probabilities are, i.e. they consider equally the positive (stego) and negative (clean) classes. In contrast, a Precision Recall Curve (PRC) would be more useful if the proposed method were focused in how the classifier was behaving on one class. Therefore, the Area Under Curve (AUC) value was chosen as a metric for the proposed model. The ROC curve is plotted with TP rate against the FP rate where TP rate is on y-axis and FP rate is on the x-axis.

5.10 Experimental results

Initially, KarNet was used as a typical convolutional neural network to discriminate clean from stego images. After training the network, the 2000 unseen images from the test set were presented to KarNet and the metrics discussed in Section 5.9 were calculated. Precision, recall and F1 score detected for each steganographic algorithm and for all embedding rates is shown in Tables 5.11&5.12, while Figures 5.15 – 5.18 show classification metrics such as accuracy, detection error, sensitivity and specificity of the proposed convolutional neural network. Figures 5.19&5.20 show the ROC curve for each one of the examined steganographic algorithms for all embedding rates along with the respective AUC value.

Table 5.11: Combined output matrix for S-UNIWARD - all embedding rates.

Embedding rate (bpp)	Class	Precicion	Recall	F1	ROC Area	Accuracy
0.4	Clean	0.862	0.880	0.871	0.947	86.95%
	Stego	0.877	0.859	0.868		
0.3	Clean	0.824	0.829	0.826	0.916	82.60%
	Stego	0.828	0.823	0.825		
0.2	Clean	0.734	0.832	0.780	0.854	76.55%
	Stego	0.806	0.699	0.748		
0.1	Clean	0.644	0.730	0.684	0.729	66.30%
	Stego	0.688	0.596	0.639		

Table 5.12: Combined output matrix for WOW - all embedding rates.

Embedding rate (bpp)	Class	Precicion	Recall	F1	ROC Area	Accuracy
0.4	Clean	0.808	0.920	0.861	0.931	85.10%
	Stego	0.907	0.782	0.840		
0.3	Clean	0.795	0.803	0.799	0.881	79.80%
	Stego	0.801	0.793	0.797		
0.2	Clean	0.746	0.678	0.710	0.802	72.35%
	Stego	0.705	0.769	0.736		
0.1	Clean	0.605	0.662	0.632	0.673	61.50%
	Stego	0.627	0.568	0.596		

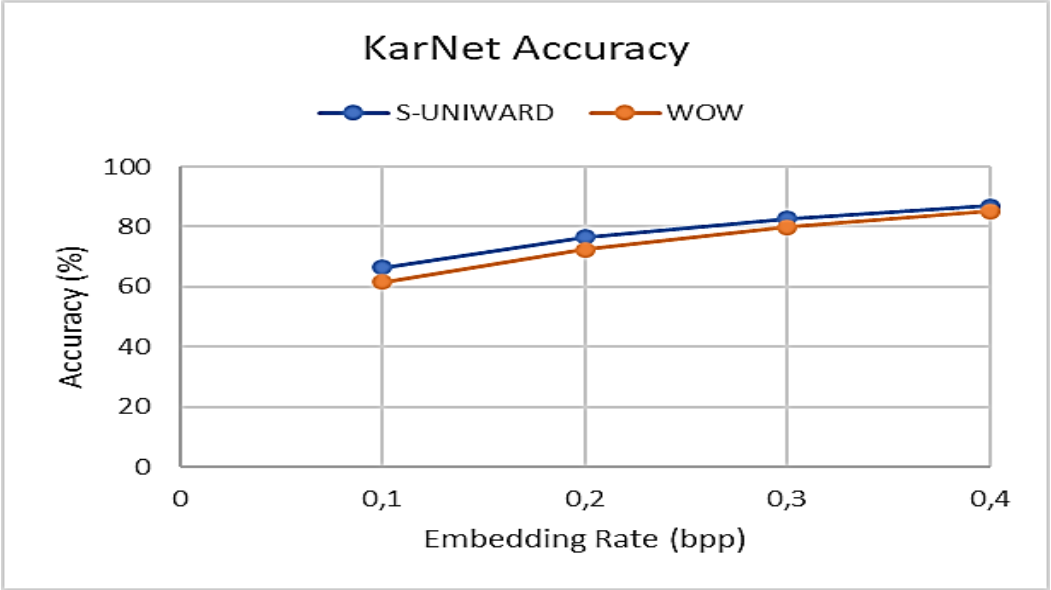

Figure 5.15: Detected accuracy of KarNet.

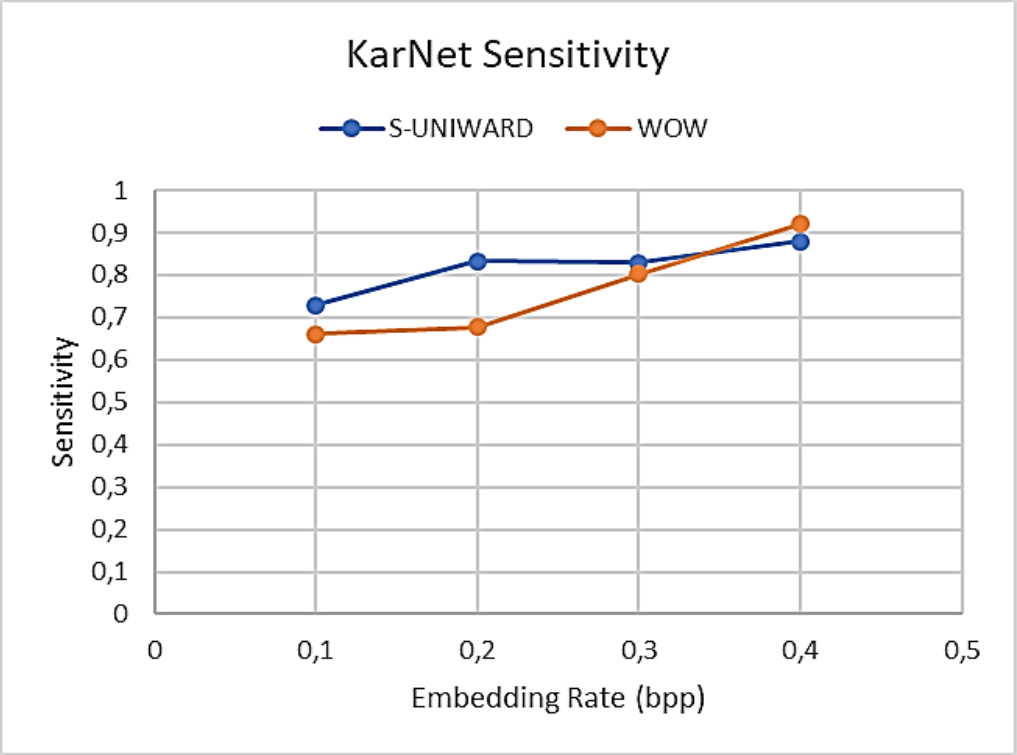

Figure 5.16: Sensitivity of KarNet.

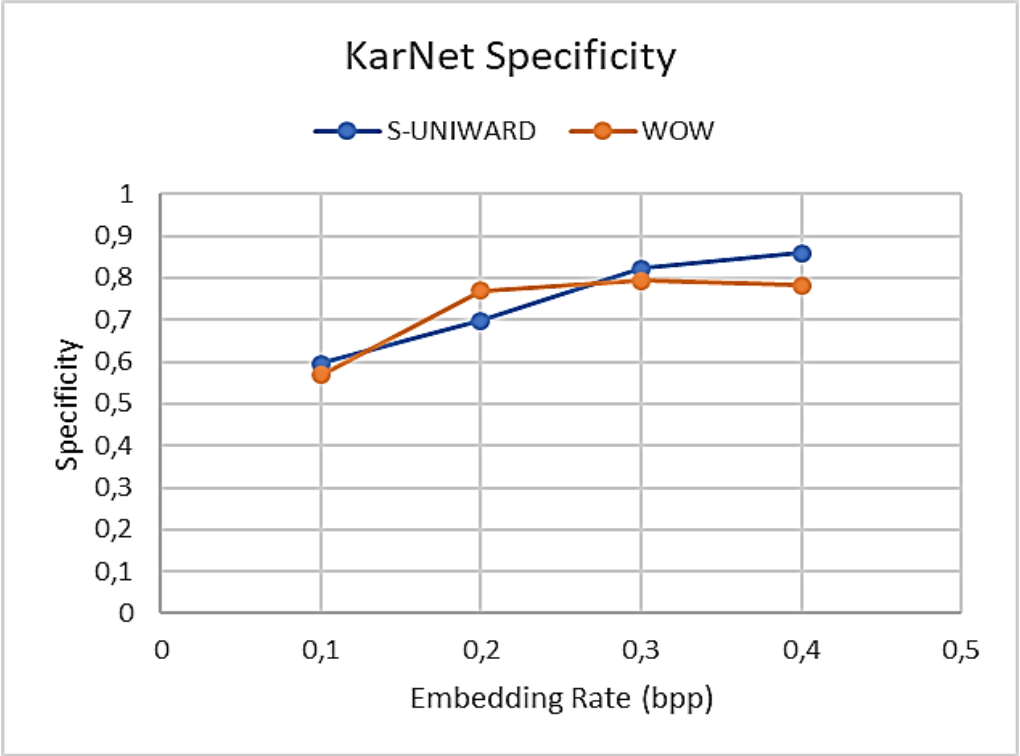

Figure 5.17: Specificity of KarNet.

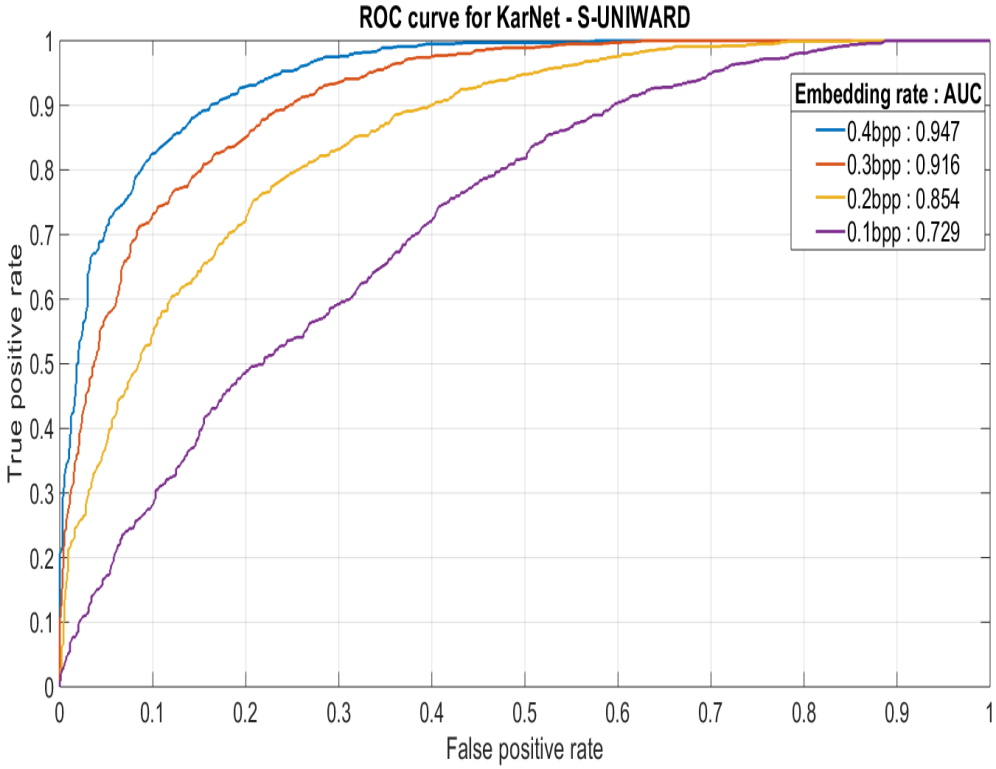

Figure 5.18: ROC curve for KarNet - S-UNIWARD.

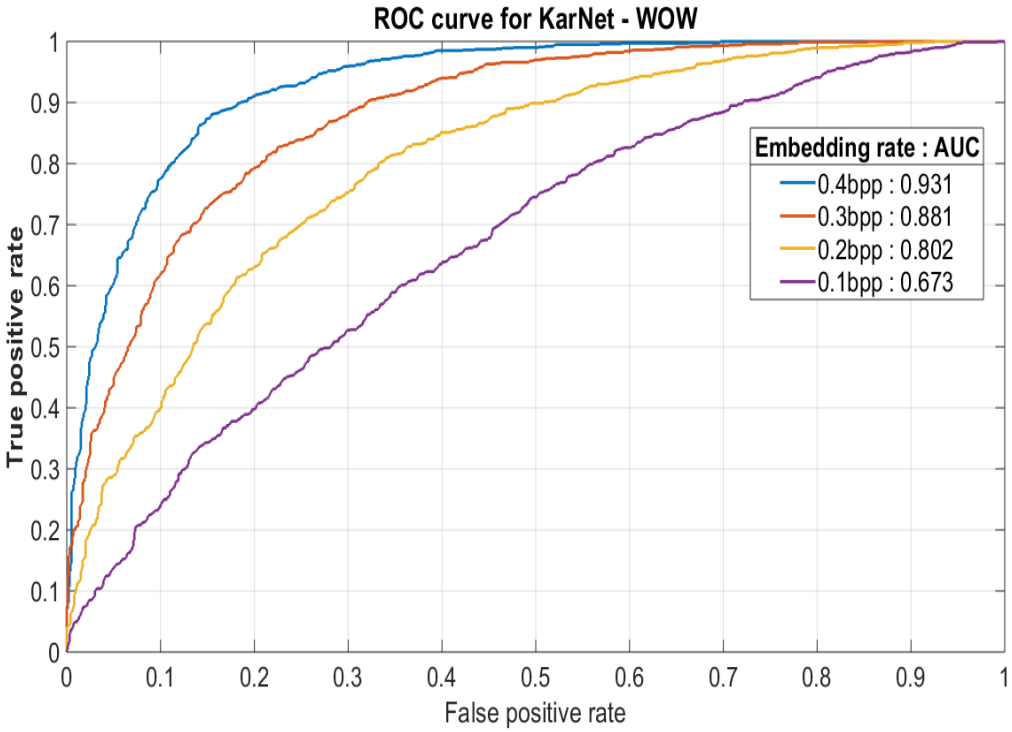

Figure 5.19: ROC curve for KarNet – WOW.

The obtained results shown in Tables 5.11&5.12 and in Figures 5.15-5.19 show that KarNet can identify clean from stego images very well. In this point it must also be noted accuracy of the KarNet is denoted as the mean accuracy of both classes i.e. clean and stego.

The resulted confusion matrices in Tables 5.13&5.14 shows the accuracy of the proposed dilated convolutional neural network per class, while Figures 5.20-5.23 show the accuracy per class and for all embedding rates for every examined steganographic algorithm i.e. S-UNIWARD & WOW.

Table 5.13: KarNet confusion matrix for S-UNIWARD and all embedding rates.

Embedding Rate ⇨	***0.4bpp***		***0.3bpp***		***0.2bpp***		***0.1bpp***	
	Classified as							
Actual Class	***Clean***	***Stego***	***Clean***	***Stego***	***Clean***	***Stego***	***Clean***	***Stego***
Clean	880	120	829	171	832	168	730	270
Stego	141	859	177	823	301	699	404	596

Table 5.14: KarNet confusion matrix for WOW and all embedding rates.

Embedding Rate ⇨	***0.4bpp***		***0.3bpp***		***0.2bpp***		***0.1bpp***	
	Classified as							
Actual Class	***Clean***	***Stego***	***Clean***	***Stego***	***Clean***	***Stego***	***Clean***	***Stego***
Clean	920	80	803	197	678	322	662	338
Stego	218	782	207	793	231	769	432	568

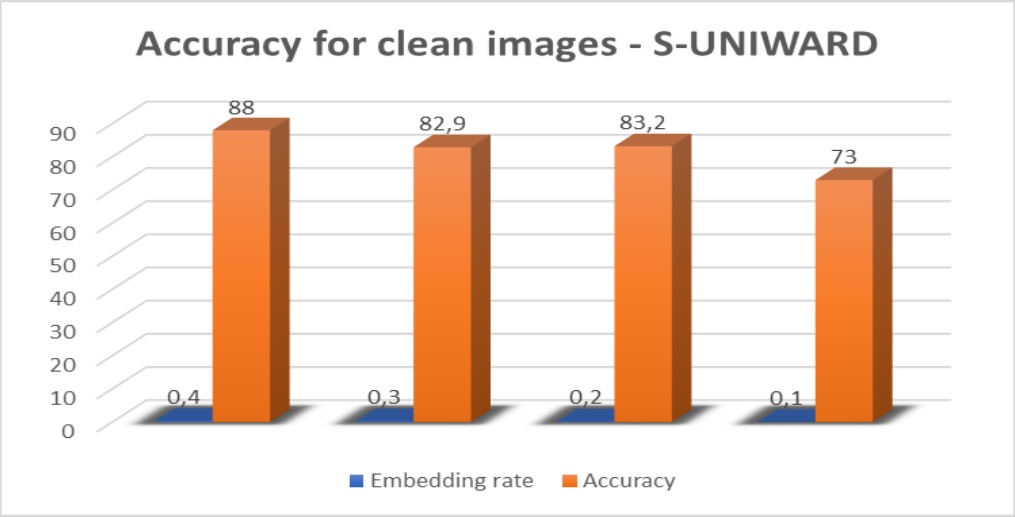

Figure 5.20: KarNet's accuracy for clean images – S-UNIWARD.

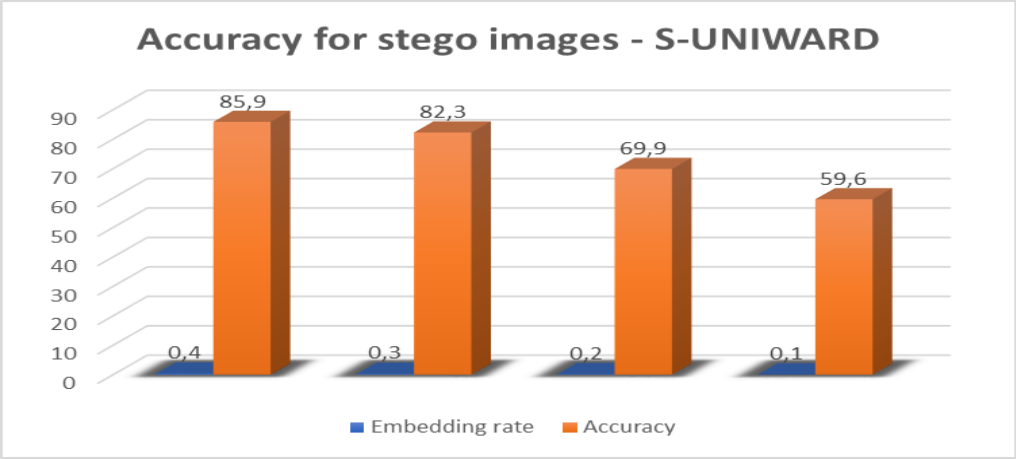

Figure 5.21: KarNet's accuracy for stego images – S-UNIWARD.

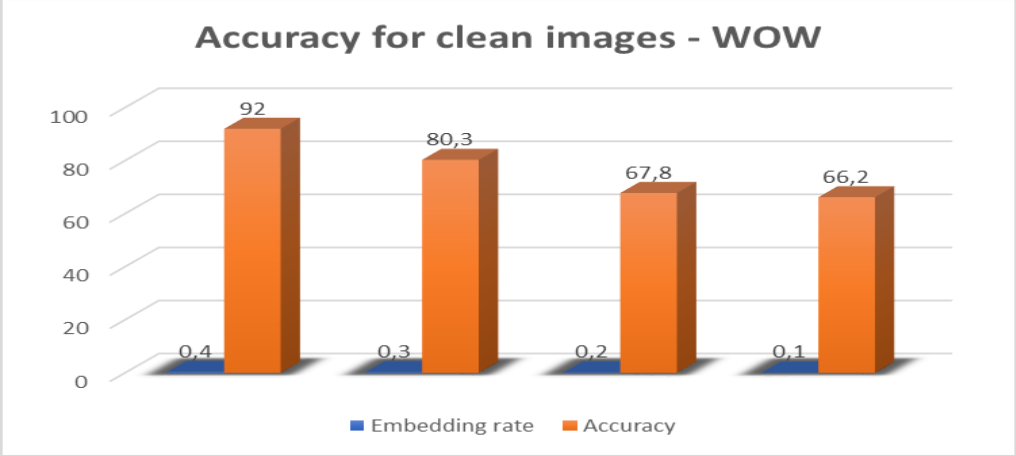

Figure 5.22: KarNet's accuracy for clean images – WOW.

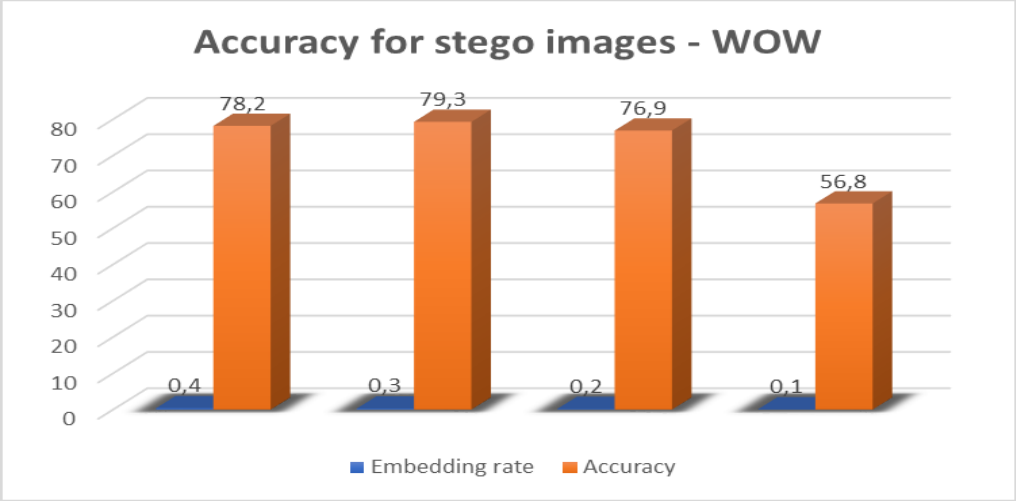

Figure 5.23: KarNet's accuracy for stego images – WOW.

5.11 Comparison of KarNet performance against to state-of-art CNNs

KarNet was compared against other state-of-art convolutional neural networks such as those described in [144],[142],[146] and summarized in Table 5.8. Experimental results are shown in Table 5.15. In Figures 5.24-5.27 the comparison of KarNet against the aforementioned CNN's for embedding rates of 0.4bpp & 0.2bpp is given.

Table 5.15: Steganalysis error probability (%) - KarNet against state-of-the-art CNNs.

	S-UNIWARD		WOW	
Method \ Embedding rate (bpp)	**0.2**	**0.4**	**0.2**	**0.4**
KarNet	**23.45**	**13.05**	**27.65**	14.90
Yedrudj-Net	36.70	22.80	27.80	**14.10**
Ye-Net	33.18	23.74	28.08	20.44
IAS-CNN	37.60	24.95	31.85	19.25

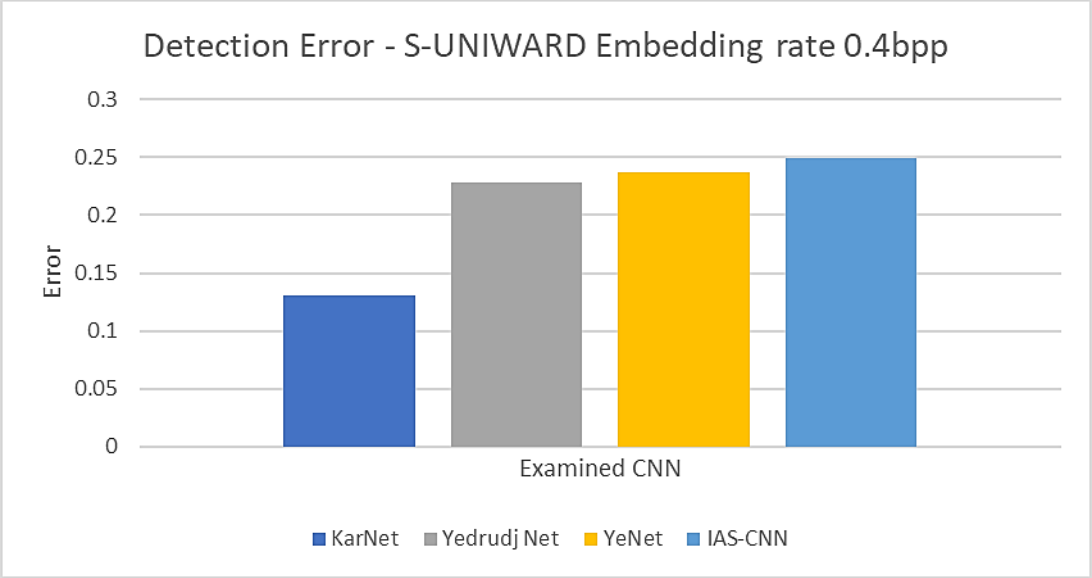

Figure 5.24: Error of KarNet against other CNNs – S-UNIWARD - 0.4bpp.

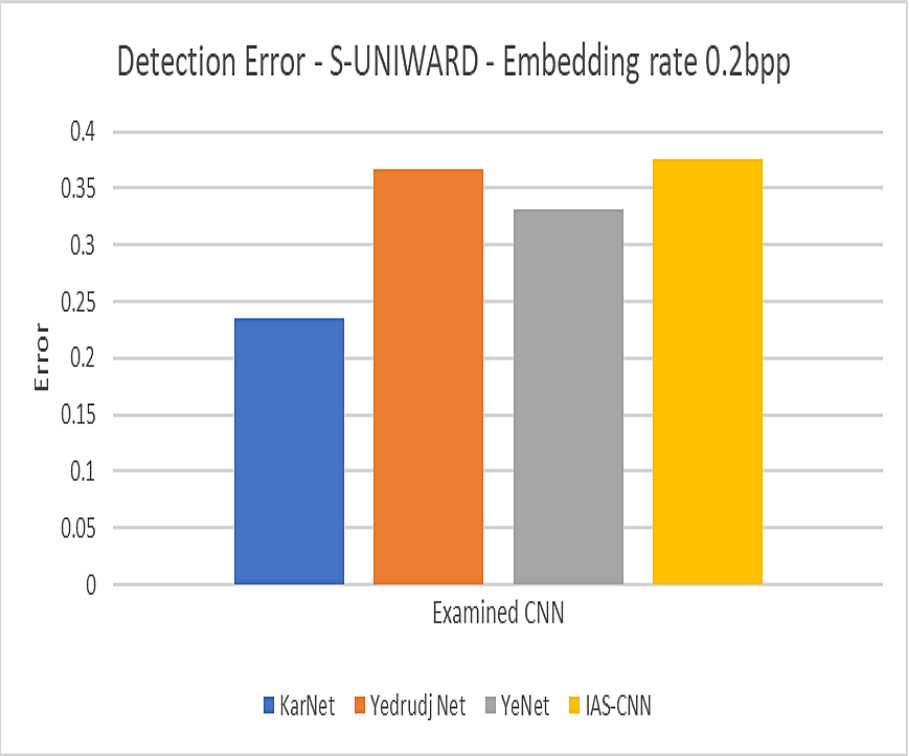

Figure 5.25**:** Error of KarNet against other CNNs – S-UNIWARD - 0.2bpp.

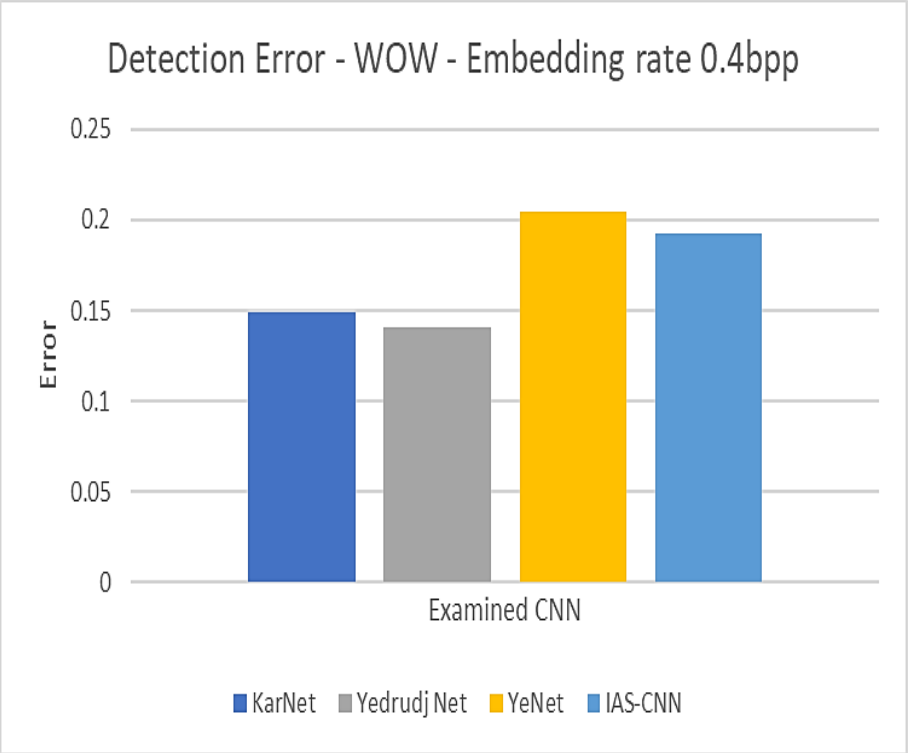

Figure 5.26: Error of KarNet against other CNNs – WOW - 0.4bpp.

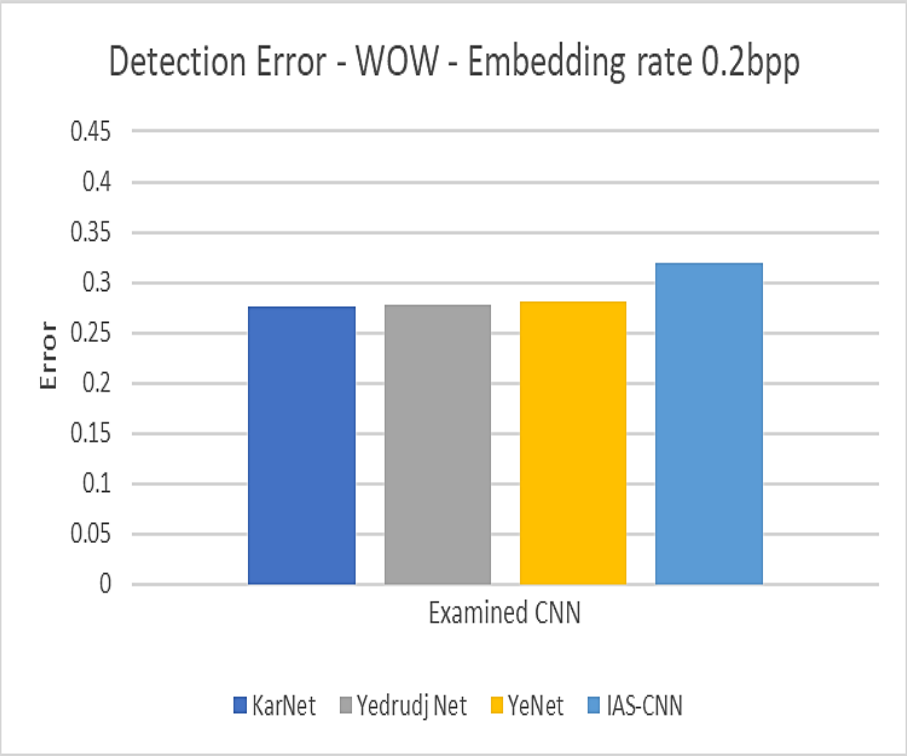

Figure 5.27: Error of KarNet against other CNNs – WOW - 0.2bpp.

5.12 Discussion

A novel convolutional neural network was designed and implemented for spatial image steganalysis. The main differences against other state-of the-art similar convolutional neural networks are the utilization of dilated convolutions and the use of Leaky ReLU as an activation function.

Dilated convolutions were utilized to increase the receptive field of the convolution without increasing the size of the kernel. This way the number of learnable parameters was not increased, and the computational complexity was lower. Furthermore, the utilization of LeakyReLU as an activation function has overcome the ReLU 'dying neuron' problem. These two factors were crucial to achieve excellent classification results even for very low embedding rates.

Moreover, KarNet was compared against other state-of-the-art convolutional neural networks used in spatial image steganalysis and it was

proved that our proposed CNN outperforms them. For S-UNIWARD algorithm, KarNet outperforms all other state-of-the-art CNNs. It achieves a lowest error from 9.75% to 13.25% than Yedrudj-Net. The comparison with IAS-CNN shows the same results; KarNet is better from 11.90% to 14.15%. The results when comparing with Ye-Net are the same; the error is lowest from 9.73% to 10.69%. For WOW algorithm the obtained results are almost identical. KarNet outperforms Ye-Net and IAS-CNN and only Yedrudj-Net shows slightly worse error probability (0.15% at 0.2 bpp) or slightly better (0.8% at 0.4bpp).

Chapter 6
KarNet as Feature Extractor

6.1 Introduction

The second major objective of this research was to investigate whether the softmax layer of a convolutional neural network could be replaced by another traditional machine learning classifier with the same or better results. Thus, the efficiency of the convolutional neural network solely as a feature extractor, should be examined.

The idea of using a CNN for feature extraction is not new [148],[167]. Feature extraction from a CNN can be performed either from a pretrained CNN (transfer learning) [168] or by designing and training a CNN from scratch which is the case studied in this dissertation.

In order to use a convolutional neural network as a feature extractor, the activations of the first fully connected layer have to be extracted and form the feature vector for each one of the images in the training set. KarNet was trained with a training set consisting of 70200 images and the number of neurons in the first fully connected layer were 250. Consequently, a feature matrix sized 70200x250 was formed and utilized to train the Random Forest classifier. Figure 6.1 shows the block diagram of the process.

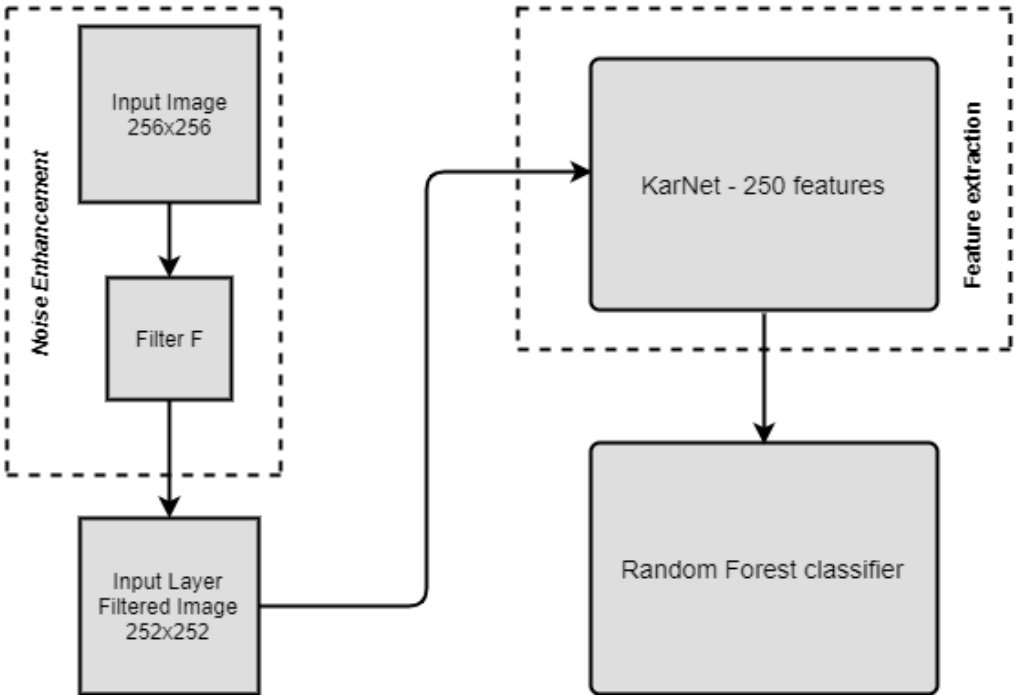

Figure 6.1: KarNet as feature extractor with Random Forest classifier.

The three convolutional blocks of the KarNet were used as a feature extractor/selector and the extracted 250-dimensional feature vector (from the first fully connected layer) was utilized to feed a Random Forest classifier. Figure 6.2 shows the extracted features in each one of the three conv blocks.

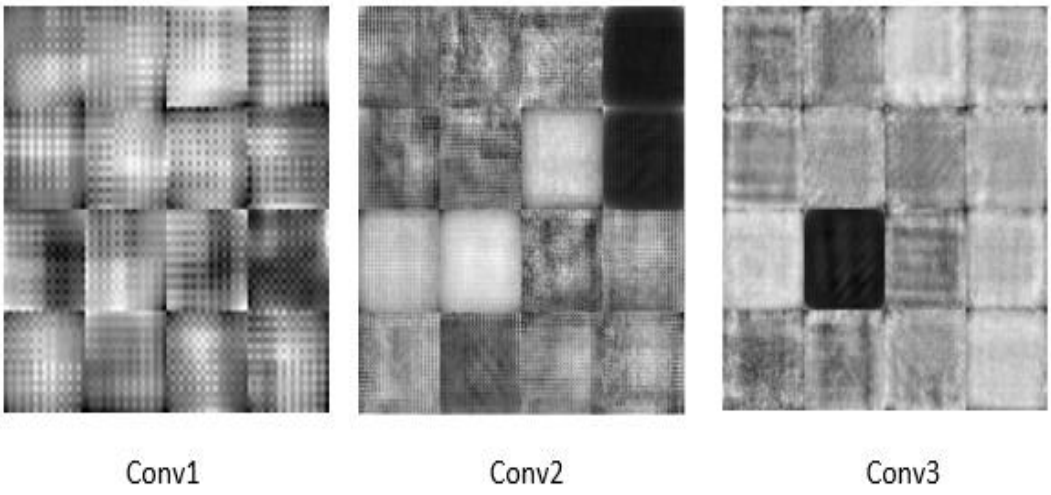

Figure 6.2: The extracted features in the three conv blocks.

6.2 The classifier

The extracted feature vector was utilized to feed a Random Forest classifier. A Random Forest is an ensemble of Decision Trees (DT), usually trained with the “bagging” method (Figure 6.3) [16]. Random forest actually improves bagging, by using at each split of each tree only a small subset of features rather than the total.

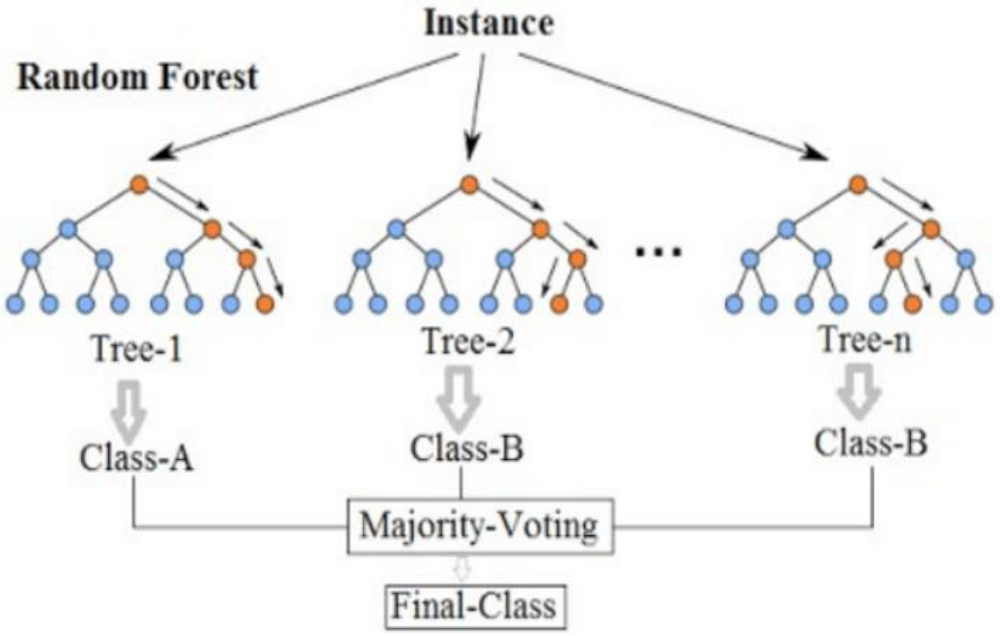

Figure 6.3: A typical Random Forest - Source : [169].

Given n available features, a subset will have a total of $\sqrt{n}$ features selected at random. Therefore, the algorithm by following the above strategy decorrelates each utilized tree. Each tree is making a decision (class) and the class with the maximum number of votes becomes the algorithm’s prediction.

It is a very handy and fast algorithm, typically with high accuracy even with the default hyperparameters and it is robust to overfitting [16]. The only limitation relies on the number of trees; the larger the number of trees the slower the

algorithm becomes but the higher the accuracy. It is found [16], [170] that 1000 is a well-chosen number of trees to have accurate results.

The reasons why a Random Forest classifier was chosen instead of others were:

- The algorithm can easily handle binary features, categorical features, and numerical features. The data need almost no pre-processing since they do not need to be rescaled or transformed.
- Parallel execution can be chosen. This leads to less computation time.
- Since the algorithm uses subsets of features, the algorithm can handle high dimensional data as well. Furthermore, this leads to faster training times than a simple decision tree.
- The algorithm is robust to outliers.
- Although this is not our case, the algorithm balances the error in unbalanced data sets. Random forest tries to minimize the overall error rate.

The number of leaves of the Random Forest classifier was set to 16 ($\sim\sqrt{250}$) and 1000 trees were constructed. Experiments with more (2000-3000-5000-10000) and less (500) trees were also conducted but the optimal number was found to be 1000.

6.3 Experimental Results

Tables 6.1 – 6.4 show the combined confusion matrix of the trained Random Forest classifier along with other classification metrics, while Figure 6.4 shows the classification accuracy of the Random Forest classifier for both embedding algorithms and for all embedding rates.

Table 6.1: Combined confusion matrix for S-UNIWARD - all embedding rates.

Embedding Rate ⇨	**0.4bpp**		**0.3bpp**		**0.2bpp**		**0.1bpp**	
	Classified as							
Actual Class	***Clean***	***Stego***	***Clean***	***Stego***	***Clean***	***Stego***	***Clean***	***Stego***
Clean	881	119	863	137	805	195	754	246
Stego	164	836	209	791	278	722	429	571

Table 6.2: Combined output matrix for S-UNIWARD - all embedding rates.

Embedding rate	**Class**	**Precicion**	**Recall**	**F1**	**ROC Area**	**Accuracy**
0.4	Clean	0.843	0.881	0.862	0.944	85.85%
	Stego	0.875	0.836	0.855		
0.3	Clean	0.805	0.863	0.833	0.917	82.70%
	Stego	0.852	0.791	0.821		
0.2	Clean	0.743	0.805	0.773	0.857	76.35%
	Stego	0.787	0.722	0.753		
0.1	Clean	0.637	0.754	0.691	0.740	66.25%
	Stego	0.699	0.571	0.629		

Table 6.3: Combined confusion matrix for WOW and all embedding rates.

Embedding Rate ⇨	**0.4bpp**		**0.3bpp**		**0.2bpp**		**0.1bpp**	
	Classified As							
Actual Class	***Clean***	***Stego***	***Clean***	***Stego***	***Clean***	***Stego***	***Clean***	***Stego***
Clean	871	129	829	171	764	236	675	325
Stego	175	825	244	756	302	698	429	571

Table 6.4: Combined output matrix for WOW and all embedding rates.

Embedding rate	Class	Precicion	Recall	F1	ROC Area	Accuracy
0.4	Clean	0.833	0.871	0.851	0.930	84.80%
	Stego	0.865	0.825	0.844		
0.3	Clean	0.773	0.829	0.800	0.884	79.25%
	Stego	0.816	0.756	0.785		
0.2	Clean	0.717	0.764	0.740	0.807	73.10%
	Stego	0.747	0.698	0.722		
0.1	Clean	0.611	0.675	0.642	0.678	62.30%
	Stego	0.637	0.571	0.602		

In Figures 6.5 - 6.6 the accuracy detected by KarNet's softmax classifier and the Random Forest classifier for both steganographic algorithms is compared.

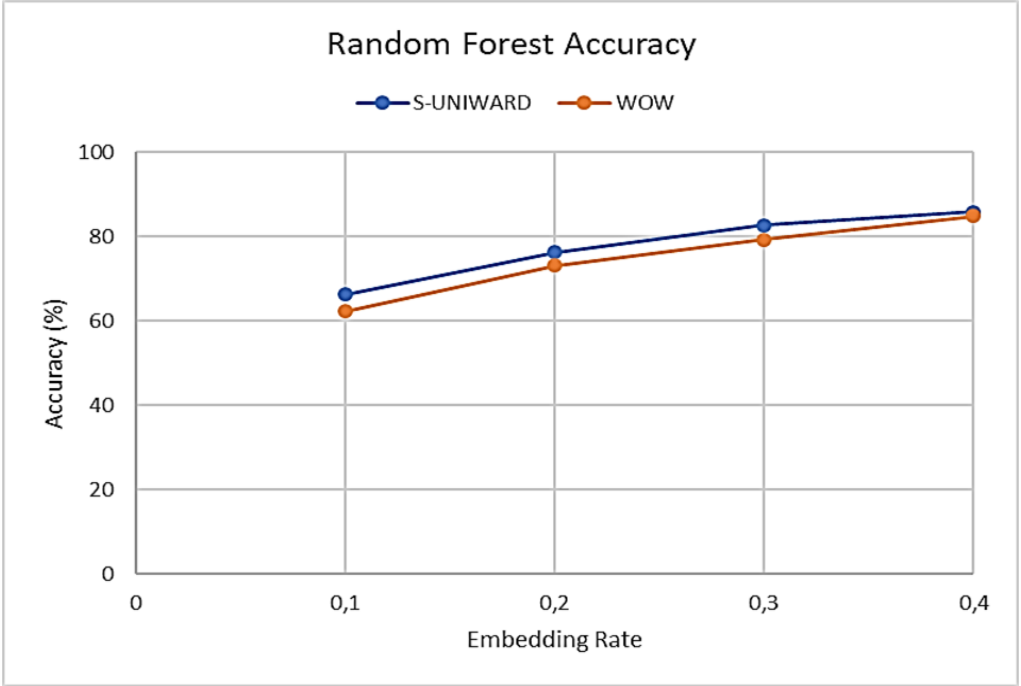

Figure 6.4: Detected accuracy of the Random Forest classifier.

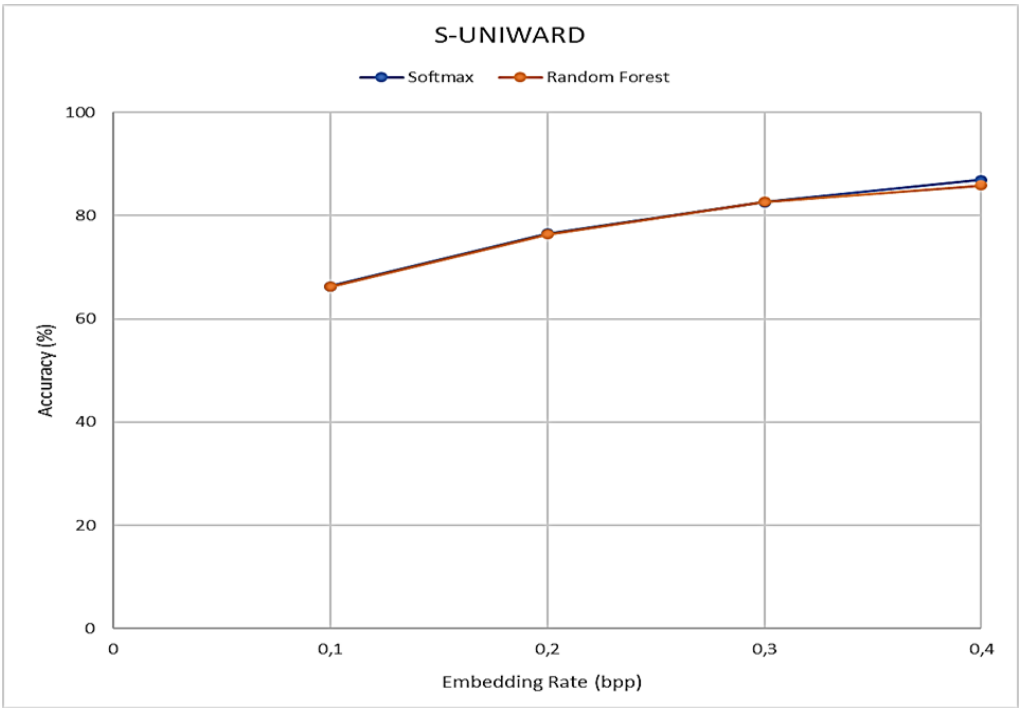

Figure 6.5: Comparison of the Softmax and the Random Forest Classifier for the S-UNIWARD algorithm.

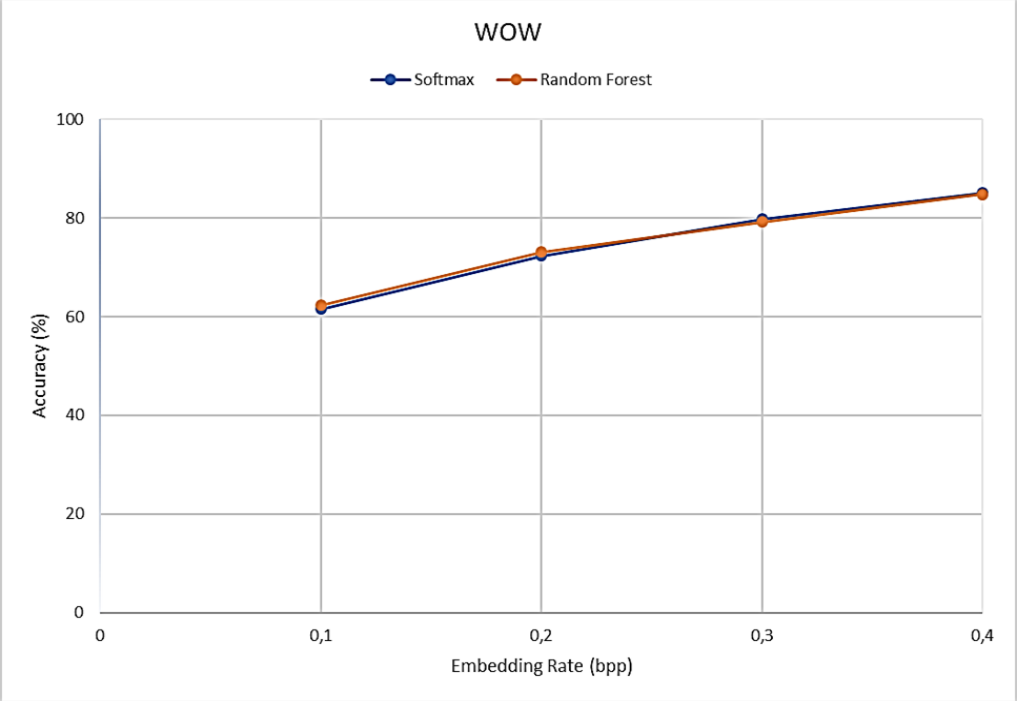

Figure 6.6: Comparison of the Softmax and the Random Forest Classifier for the WOW algorithm.

Looking at the obtained results concerning the classification accuracy from the softmax classifier a typical CNN has, in comparison with the ones of the Random Forest classifier (trained by the extracted feature vector), it is clear that the Random Forest classifier has equal (S-UNIWARD) or better (WOW) results. This proves our research question that a CNN could be used as a feature extractor and the classification step could be done by another traditional machine learning method.

6.4 Comparison of the proposed method to state-of-the-art feature extractors

Furthermore, our method (KarNet as feature extractor) was compared to the state-of-the-art methods proposed to the literature i.e. Subtractive Pixel Adjacency Matrix (SPAM) [48] and Spatial Rich Model (SRM). SPAM method extracts only 686 image features while SRM extracts 34671 features.

SPAM method computes the differences between adjacent pixels along eight directions and afterwards it uses a second order Markov chain to extract the 686 image features. The model can be extended by changing the order of the Markov chain and the range of differences between adjacent pixels T. In the conducted experiments a second order Markov chain was chosen, and the value of T was 3.

SRM is another state-of-the-art method used in steganalysis. It computes 106 different submodels (co-occurrence matrices), including their differently quantized versions and it produces a 34671-dimensional feature vector.

Both SRM and SPAM feature extractors were applied to the same dataset and the extracted features were then used to train the same classifier i.e. the Random Forest. The number of leaves of the classifier was set to 26 ($\sim\sqrt{686}$) for SPAM method, and 186 ($\sim\sqrt{34671}$) for SRM method.

The number of constructed trees was the same (i.e. 1000) as in our proposed method. Tables 6.5 - 6.8 show the classification results for each one of the examined steganographic algorithms and for all embedding rates, while

Figures 6.7 & 6.8 show the accuracy achieved by state-of-the-art methods in comparison to our hybrid classification model proposal.

Table 6.5: Output for SPAM method – S-UNIWARD.

Embedding rate (bpp)	Class	Precicion	Recall	F1	ROC Area	Accuracy
0.4	Clean	0.656	0.768	0.708	0.794	68.30%
	Stego	0.720	0.598	0.654		
0.3	Clean	0.611	0.736	0.668	0.732	63.35%
	Stego	0.668	0.531	0.592		
0.2	Clean	0.569	0.702	0.628	0.670	58.50%
	Stego	0.611	0.468	0.530		
0.1	Clean	0.522	0.670	0.587	0.570	52.85%
	Stego	0.540	0.387	0.451		

Table 6.6: Output for SPAM method – WOW.

Embedding rate (bpp)	Class	Precicion	Recall	F1	ROC Area	Accuracy
0.4	Clean	0.650	0.745	0.694	0.777	67.15%
	Stego	0.701	0.598	0.645		
0.3	Clean	0.610	0.701	0.652	0.725	62.65%
	Stego	0.649	0.552	0.596		
0.2	Clean	0.574	0.666	0.617	0.654	58.60%
	Stego	0.602	0.506	0.550		
0.1	Clean	0.523	0.618	0.566	0.558	52.70%
	Stego	0.533	0.436	0.480		

Table 6.7: Output for SRM method – S-UNIWARD.

Embedding rate (bpp)	Class	Precicion	Recall	F1	ROC Area	Accuracy
0.4	Clean	0.676	0.794	0.730	0.824	70.65%
	Stego	0.750	0.619	0.678		
0.3	Clean	0.617	0.763	0.682	0.751	64.50%
	Stego	0.690	0.527	0.598		
0.2	Clean	0.571	0.724	0.638	0.668	59.00%
	Stego	0.623	0.456	0.527		
0.1	Clean	0.536	0.694	0.605	0.589	54.65%
	Stego	0.566	0.399	0.468		

Table 6.8: Output for SRM method – WOW.

Embedding rate (bpp)	Class	Precicion	Recall	F1	ROC Area	Accuracy
0.4	Clean	0.717	0.796	0.755	0.854	74.10%
	Stego	0.771	0.686	0.726		
0.3	Clean	0.675	0.753	0.712	0.795	69.55%
	Stego	0.721	0.638	0.677		
0.2	Clean	0.627	0.693	0.658	0.725	64.00%
	Stego	0.657	0.587	0.620		
0.1	Clean	0.565	0.634	0.597	0.631	57.25%
	Stego	0.583	0.511	0.544		

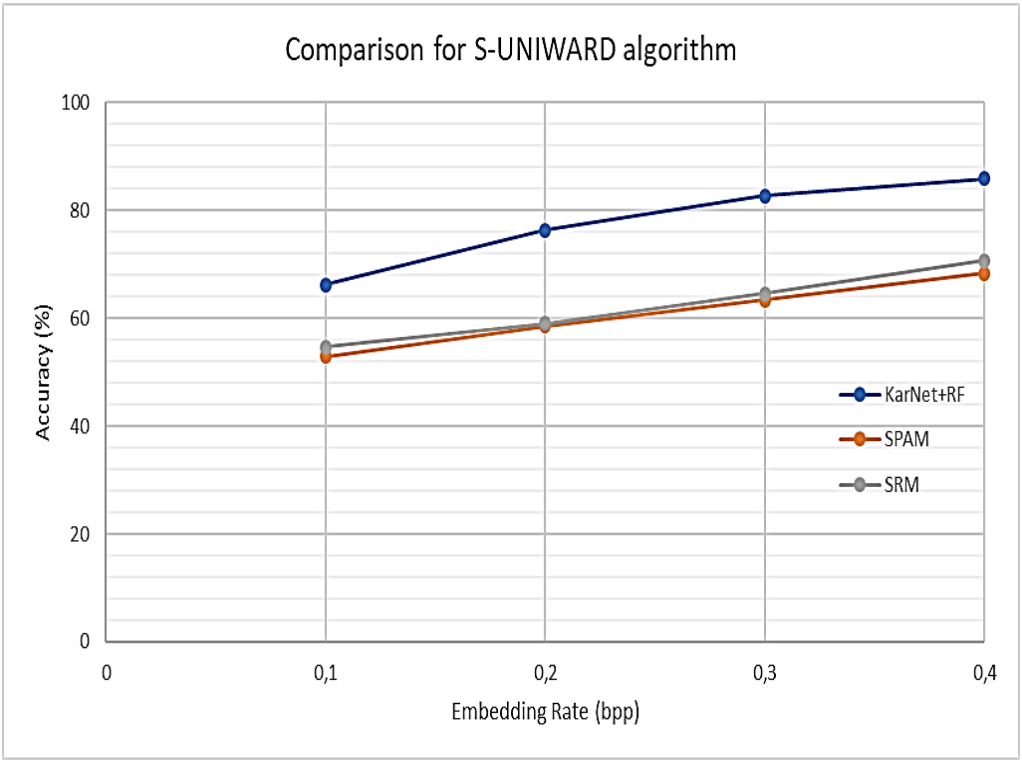

Figure 6.7: Comparison of all methods for the S-UNIWARD algorithm.

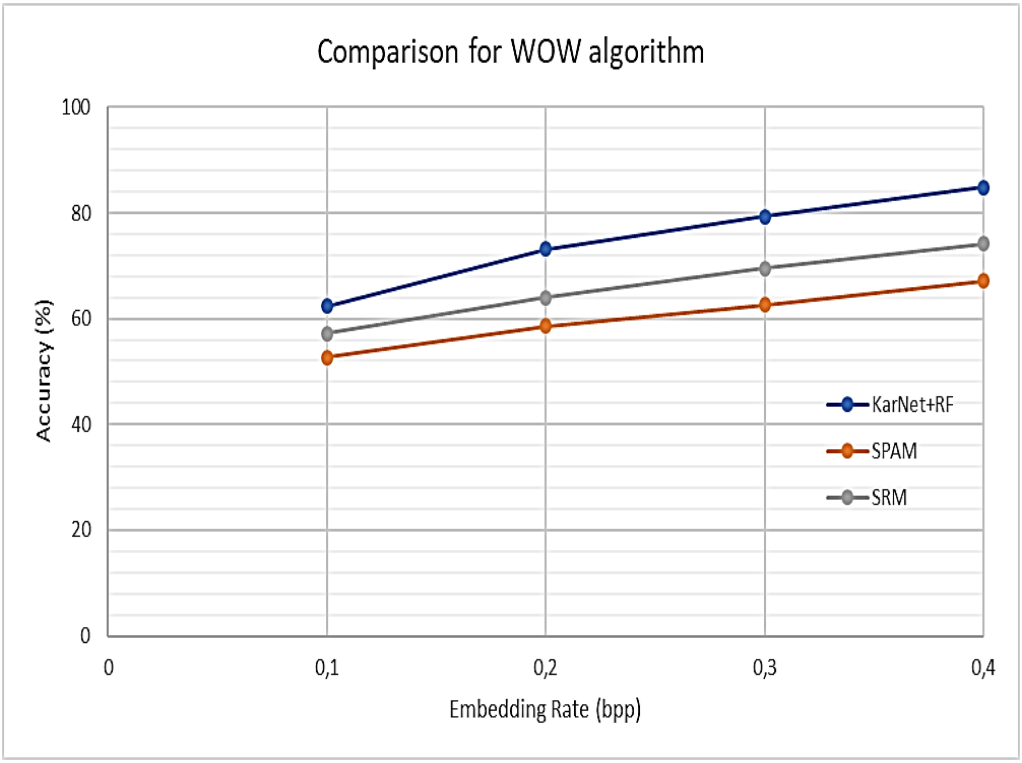

Figure 6.8: Comparison of all methods for the WOW algorithm.

6.5 Discussion

In this Chapter it was explored whether the softmax classification layer of a convolutional neural network could be replaced by another classifier with similar results. In order to do this, the activations from the first fully connected layer of KarNet were extracted and formed the feature vector that trained a Random Forest classifier. Experimental results showed that the research hypothesis is correct. The same and, in many cases, better classification results were achieved than the traditional softmax classifier that a convolutional neural network utilizes.

Moreover, the proposed hybrid classification scheme was compared against state-of-the-art feature extraction methods ie. SPAM and SRM. The obtained results proved that our proposed method outperforms in all cases (both steganographic algorithms and all embedding rates), SPAM and SRM methods. More specifically, concerning S-UNIWARD steganographic algorithm our method achieves 13.40% (at 0.1bpp) to 19.35% (at 0.3bpp) better classification accuracy

than the SPAM method, while for the SRM method the proposed research approach is 11.60%-18.20% more accurate. The same comparison for WOW steganographic algorithm shows that for the SPAM method 9.60% (at 0.1bpp) to 17.65% (at 0.4bpp) better classification accuracy was obtained and for the SRM method the results are 5.05%-10.70% in favor of the proposed classification scheme.

Chapter 7
Conclusions and Future Work

Coming to a conclusion, about which method is more effective – in any domain- is not an easy task. There are many parameters that a digital forensic examiner must know in advance, in order to give a safe answer before deciding which method to employ. These parameters include the existence or not of the cover image, the prior knowledge of the embedded data, findings of steganography software in a suspect's computer etc. However, if it is assumed that in the majority of the cases only the stego object is known, statistical steganalysis techniques - in any domain- are more robust and more effective than signature steganalysis. This is met for both specific and universal statistical steganalysis.

In specific statistical steganalysis the proposed methods focus to the embedding procedure and attempt to find image features or statistical measures changed by the embedding algorithm. Thus, this steganalytic approach has excellent accuracy only when performed on the specific steganographic algorithm, but even a small change in the embedding algorithm usually results to low steganalytic accuracy. For this reason, universal statistical steganalysis is used. These methods can detect hidden message's existence regardless the steganographic technique used to embed secret message to the digital image. Typically, classification is performed based on extracted features that are dependent to a widespread diversity of embedding procedures. These methods

provide less accurate results than specific statistical steganalysis methods, but they can detect unseen steganographic content. Moreover, they are more flexible than the specific ones and slight changes to classification schemes may lead to the detection of more embedding algorithms as well. Nowadays utilization of deep learning techniques boosted research in steganalysis domain and provided new insights into research. In this dissertation it was demonstrated that only by utilizing dilated convolutions achieved better results than other state-of-the-art methods, without increasing the number of learnable parameters i.e. reducing computational complexity. Moreover, it was demonstrated that hybrid classification schemes perform better and faster. Although convolutional neural networks have shown excellent results in comparison to previous methods, their main disadvantages i.e. the extensive training times for adding more steganographic algorithms, limitations to image dimensions due to heavy computational load etc. must be overcome to be more sufficient. Therefore, the need to adopt hybrid classification schemes in order to overcome these limitations, becomes critical.

The obtained promising results offer a good basis to investigate more thoroughly the proposed convolutional neural network and the hybrid classification scheme. In the future the proposed convolutional neural network should be thoroughly reconsidered and slightly revised. More specifically, future research directions could be:

- Make slight modifications to the KarNet in order to detect features from more embedding steganography algorithms. These modifications may need different preprocessing strategy, the utilization of slightly different filters and changes to KarNet's training hyperparamaters.
- Experiment with images of larger size.
- Experiment with other initializers like He [171] or Glorot [172].
- Experiment with other steganography domains also, like transform domain and Spread Spectrum Image Steganography (SSIS).

- Examine whether other hybrid deep neural network models can capture and identify - in real time - images embedded with stego content.
- Integrate fuzzy rules and evolutionary algorithms especially for feature selection.

The ultimate – yet unreachable - goal for a steganalyst, is to employ a steganalysis technique that could detect any type of steganographic embedding algorithm with low computational needs and excellent accuracy. We strongly believe that universal steganalysis combined with deep learning techniques will boost research and will provide digital forensic examiners new software tools to uncover seen of the unseen.

Bibliography

[1] D. Kahn, *The codebreakers : the story of secret writing*, 1st ed. New York, NY, USA: Macmillan, 1967.

[2] R. L. Tonsetic, *Special Operations During the American Revolution*. Havertown, PA: Casemate Publishers, 2013.

[3] N. F. Johnson and S. Jajodia, “Exploring steganography: Seeing the unseen,” *IEEE Computer*, vol. 31, no. 2, pp. 26–34, 1998.

[4] R. J. Anderson and F. A. P. Petitcolas, “On The Limits of Steganography,” *IEEE Journal of Selected Areas in Communications*, vol. 16, no. 4, pp. 474–481, 1998.

[5] W. Bender, W. Butera, D. Gruhl, R. Hwang, F. J. Paiz, and S. Pogreb, “Applications for data hiding,” *IBM Systems Journal*, vol. 39, no. 3.4, pp. 547–568, Apr. 2010.

[6] J. C. Ingemar, M. L. Miller, A. B. Jeffrey, J. Fridrich, and T. Kalker, *Digital Watermarking and Steganography*, 2nd ed. Elsevier Inc., 2008.

[7] G. Palmer, “A Road Map for Digital Forensic Research,” in *Proceedings of the 2001 Digital Forensics Research Workshop (DFRWS 2004)*, New York, 2001, pp. 1–42.

[8] Daily Mail, “Porn video reveals Al Qaeda planns to hijack cruise ships and execute passengers | Daily Mail Online,” 01-May-2012. [Online]. Available: https://www.dailymail.co.uk/news/article-2137848/Porn-video-reveals-Al-Qaeda-planns-hijack-cruise-ships-execute-passengers.html#ixzz1uIgxpire. [Accessed: 09-Jul-2019].

[9] Wired, “Bin Laden: Steganography Master? | WIRED,” Jul-2001. [Online]. Available: https://www.wired.com/2001/02/bin-laden-steganography-master/?currentPage=all. [Accessed: 09-Jul-2019].

[10] Frank Gardner, “How do terrorists communicate? - BBC News,” 2013. [Online]. Available: https://www.bbc.com/news/world-24784756. [Accessed: 09-Jul-2019].

[11] H. N. Lily, “MuslimCrypt Steganography App Helps Jihadists Send Secret Messages | WIRED,” Mar-2018. [Online]. Available: https://www.wired.com/story/muslimcrypt-steganography/. [Accessed: 09-Jul-2019].

[12] “Steghide.” [Online]. Available: http://steghide.sourceforge.net/index.php. [Accessed: 04-May-2020].

[13] “Hide Files, Encrypt File Encryption Software | InvisibleSecrets.” [Online]. Available: https://www.east-tec.com/invisiblesecrets/. [Accessed: 04-May-

2020].

[14] “The SNOW Home Page.” [Online]. Available: http://www.darkside.com.au/snow/index.html. [Accessed: 04-May-2020].

[15] K. Karampidis, E. Kavallieratou, and G. Papadourakis, “A review of image steganalysis techniques for digital forensics,” *Journal of Information Security and Applications*, vol. 40, pp. 217–235, Jun. 2018.

[16] L. Breiman, “Random Forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[17] V. Holub, J. Fridrich, and T. Denemark, “Universal distortion function for steganography in an arbitrary domain,” *EURASIP Journal on Information Security*, vol. 2014, no. 1, p. 1, Dec. 2014.

[18] V. Holub and J. Fridrich, “Designing steganographic distortion using directional filters,” in *WIFS 2012 - Proceedings of the 2012 IEEE International Workshop on Information Forensics and Security*, Tenerife,Spain, 2012, pp. 234–239.

[19] B. Ionescu, H. Müller, R. Péteri, Y. D. Cid, V. Liauchuk, V. Kovalev, D. Klimuk, A. Tarasau, A. Ben Abacha, S. A. Hasan, V. Datla, J. Liu, D. Demner-Fushman, D. T. Dang-Nguyen, L. Piras, M. Riegler, M. T. Tran, M. Lux, C. Gurrin, O. Pelka, C. M. Friedrich, A. Garcìa Seco de Herrera, N. Garcia, E. Kavallieratou, C. R. del Blanco, C. Cuevas, N. Vasillopoulos, K. Karampidis, J. Chamberlain, A. Clark, and A. Campello, “ImageCLEF 2019: Multimedia Retrieval in Medicine, Lifelogging, Security and Nature,” in *CLEF 2019*, Lugano Switzerland, 2019, vol. 11696 LNCS, pp. 358–386.

[20] “About CLEF.” [Online]. Available: http://www.clef-campaign.org/. [Accessed: 06-May-2020].

[21] “ImageCLEFsecurity | ImageCLEF / LifeCLEF - Multimedia Retrieval in CLEF.” [Online]. Available: https://www.imageclef.org/2019/security. [Accessed: 06-May-2020].

[22] K. Karampidis, N. Vasillopoulos, C. Rodrguez, E. Kavallieratou, C. R. C. del Blanco, and N. Garcia, “Overview of the ImageCLEFsecurity 2019 task,” Lugano, Switzerland, 2019.

[23] S. Katzenbeisser and F. A. P. Petitcolas, *Information hiding techniques for steganography and digital watermarking*. Artech House, Inc., 2000.

[24] F. Djebbar, B. Ayad, K. A. Meraim, and H. Hamam, “Comparative study of digital audio steganography techniques,” *Eurasip Journal on Audio, Speech, and Music Processing*, vol. 2012, no. 1, pp. 1–16, Oct. 2012.

[25] Rupanshi, “Audio Steganography by Direct Sequence Spread Spectrum,” *International Journal of Computer Trends and Technology*, vol. 13, no. 2, 2014.

[26] M. M. Sadek, A. S. Khalifa, and M. G. M. Mostafa, “Video steganography: a comprehensive review,” *Multimedia Tools and Applications*, vol. 74, no. 17, pp. 7063–7094, Sep. 2015.

[27] J. Lubacz, W. Mazurczyk, and K. Szczypiorski, “Principles and overview of network steganography,” *IEEE Communications Magazine*, vol. 52, no. 5, pp. 225–229, 2014.

[28] J. Collins and S. Agaian, “Trends Toward Real-Time Network Data Steganography,” *International Journal of Network Security & Its Applications*, vol. 8, no. 2, pp. 01–21, Apr. 2016.

[29] A. Febryan, T. W. Purboyo, and R. E. Saputra, “Steganography Methods on Text, Audio, Image and Video: A Survey,” *International Journal of Applied Engineering Research*, vol. 12, pp. 10485–10490, 2017.

[30] S. Bhattacharyya and G. Sanyal, “Steganalysis of LSB Image Steganography using Multiple Regression and Auto Regressive (AR) Model,” *International Journal of Computer Technology and Applications*, vol. 2, no. 4, pp. 1069–1077, 2011.

[31] Liu Shaohui, Yao Hongxun, and Gao Wen, “Neural network based steganalysis in still images,” in *2003 International Conference on Multimedia and Expo. ICME ’03. Proceedings (Cat. No.03TH8698)*, Baltimore, MD, USA, 2003, pp. II–509.

[32] S. Tan and B. Li, “Stacked convolutional auto-encoders for steganalysis of digital images,” in *2014 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA 2014*, Chiang Mai, Thailand, 2014, pp. 1–4.

[33] “Breaking a steganography software: Camouflage.” [Online]. Available: http://www.guillermito2.net/stegano/camouflage/index.html. [Accessed: 07-Dec-2017].

[34] “Breaking a steganography software: JpegX.” [Online]. Available: http://www.guillermito2.net/stegano/jpegx/index.html. [Accessed: 07-Dec-2017].

[35] “Analyzing steganography softwares.” [Online]. Available: http://www.guillermito2.net/stegano/. [Accessed: 28-Nov-2017].

[36] M. Rana, “Parameter Evaluation and Comparison of algorithms used in Steganography,” *International Journal of Engineering Science and Computing*, p. 8134, 2016.

[37] “Hide files and folders - Masker 7.5.” [Online]. Available: http://www.softpuls.com/masker/. [Accessed: 28-Nov-2017].

[38] “StegSpy.” [Online]. Available: http://www.spy-hunter.com/stegspydownload.htm. [Accessed: 05-May-2020].

[39] J. Fridrich and M. Goljan, “Practical steganalysis of digital images: state of the art,” in *Proc. SPIE 4675 , Security and Watermarking of Multimedia Contents IV*, San Jose, California, USA, 2002, vol. 4675, pp. 1–13.

[40] J. Fridrich, M. Goljan, and R. Du, “Steganalysis based on JPEG compatibility,” in *Proc. SPIE 4518, Multimedia Systems and Applications IV*, Denver, USA, 2001, pp. 275–280.

[41] R. E. Newman, I. S. Moskowitz, L. W. Chang, and M. M. Brahmadesam, “A steganographic embedding undetectable by JPEG compatibility steganalysis,” in *Petitcolas F.A.P. (eds) Information Hiding. IH 2002. Lecture Notes in Computer Science*, vol. 2578, Springer, Berlin, Heidelberg, 2003, pp. 258–277.

[42] A. Westfeld and A. Pfitzmann, “Attacks on Steganographic Systems,” Springer Berlin Heidelberg, 2000, pp. 61–76.

[43] A. Westfeld, “Detecting low embedding rates,” in *Petitcolas F.A.P. (eds) Information Hiding. IH 2002. Lecture Notes in Computer Science*, 2003, vol. 2578, pp. 324–339.

[44] R. Chandramouli, M. Kharrazi, and N. Memon, “Image Steganography and Steganalysis: Concepts and Practice,” in *Kalker T., Cox I., Ro Y.M. (eds) Digital Watermarking. IWDW 2003.*, Seoul, Korea, 2003, pp. 35–49.

[45] J. Fridrich and M. Long, “Steganalysis of LSB encoding in color images,” in *2000 IEEE International Conference on Multimedia and Expo. ICME2000. Proceedings. Latest Advances in the Fast Changing World of Multimedia (Cat. No.00TH8532)*, New York, NY, USA, 2000, vol. 3, pp. 1279–1282.

[46] J. Fridrich, M. Goljan, and R. Du, “Reliable detection of LSB steganography in color and grayscale images,” in *Proceedings of the 2001 workshop on Multimedia and security new challenges - MM&Sec ’01*, New York, NY, USA, 2001, pp. 27–30.

[47] T. Sharp, “An implementation of key-based digital signal steganography,” in *Moskowitz I.S. (eds) Information Hiding. IH 2001. Lecture Notes in Computer Science*, Pittsburgh, PA, USA, 2001, vol. 2137, no. 9, pp. 13–26.

[48] T. Pevný, P. Bas, and J. Fridrich, “Steganalysis by subtractive pixel adjacency matrix,” *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp. 215–224, Jun. 2010.

[49] J. Fridrich, J. Kodovský, V. Holub, and M. Goljan, “Steganalysis of content-adaptive steganography in spatial domain,” in *Filler T., Pevný T., Craver S., Ker A. (eds) Information Hiding. IH 2011. Lecture Notes in Computer Science*, Prague, Czech Republic, 2011, vol. 6958, pp. 102–117.

[50] M. Goljan, J. Fridrich, and R. Cogranne, “Rich model for Steganalysis of

color images," in *2014 IEEE International Workshop on Information Forensics and Security (WIFS)*, Atlanta, GA, USA, 2014, pp. 185–190.

[51] I. Avcibaş, N. Memon, and B. Sankur, "Steganalysis of watermarking techniques using image quality metrics," in *Proceedings of SPIE, Security and Watermarking of Multimedia Contents III*, San Jose, CA, United States, 2001, vol. 4314, pp. 523–531.

[52] S. Lyu and H. Farid, "Detecting hidden messages using higher-order statistics and support vector machines," in *Petitcolas F.A.P. (eds) Information Hiding. IH 2002. Lecture Notes in Computer Science*, Noordwijkerhout, The Netherlands, 2003, vol. 2578, pp. 340–354.

[53] S. Dumitrescu, Xiaolin Wu, and Zhe Wang, "Detection of LSB steganography via sample pair analysis," *IEEE Transactions on Signal Processing*, vol. 51, no. 7, pp. 1995–2007, Jul. 2003.

[54] S. Dumitrescu, Xiaolin Wu, and N. Memon, "On steganalysis of random LSB embedding in continuous-tone images," in *Proceedings. International Conference on Image Processing*, Rochester, NY, USA, 2002, vol. 3, pp. 641–644.

[55] S. Dumitrescu and Xiaolin Wu, "Steganalysis of LSB embedding in multimedia signals," in *Proceedings. IEEE International Conference on Multimedia and Expo*, Lausanne, Switzerland, 2002, pp. 581–584.

[56] B. Roue, P. Bas, and J.-M. Chassery, "Improving LSB steganalysis using marginal and joint probabilistic distributions," in *Proceedings of the 2004 multimedia and security workshop on Multimedia and security - MM&Sec '04*, New York, NY, USA, 2004, pp. 75–80.

[57] P. Lu, X. Luo, Q. Tang, and L. Shen, "An Improved Sample Pairs Method for Detection of LSB Embedding," Springer Berlin Heidelberg, 2004, pp. 116–127.

[58] İ. Avcıbaş, M. Kharrazi, N. Memon, and B. Sankur, "Image Steganalysis with Binary Similarity Measures," *EURASIP Journal on Advances in Signal Processing*, vol. 2005, no. 17, pp. 2749–2757, 2005.

[59] S. Dumitrescu and Xiaolin Wu, "A new framework of LSB steganalysis of digital media," *IEEE Transactions on Signal Processing*, vol. 53, no. 10, pp. 3936–3947, Oct. 2005.

[60] Li Zhi, Sui Ai Fen, and Yang Yi Xian, "A LSB steganography detection algorithm," in *14th IEEE Proceedings on Personal, Indoor and Mobile Radio Communications, 2003. PIMRC 2003.*, Beijing, China, 2003, pp. 2780–2783.

[61] T. Zhang and X. Ping, "Reliable detection of LSB steganography based on the difference image histogram," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP*

'03)., Hong Kong, China, 2003, vol. 3, pp. 545–548.

[62] J. Fridrich, M. Goljan, and D. Soukal, "Higher-order statistical steganalysis of palette images," in *Proc. SPIE 5020, Security and Watermarking of Multimedia Contents V*, Santa Clara, CA, USA, 2003, vol. 5020, pp. 178–190.

[63] A. D. Ker, "Quantitative evaluation of pairs and RS steganalysis," in *Proc. SPIE 5306, Security, Steganography, and Watermarking of Multimedia Contents VI*, San Jose, California, USA, 2004, pp. 83–97.

[64] A. D. Ker, "Improved Detection of LSB Steganography in Grayscale Images," in *Fridrich J. (eds) Information Hiding. IH 2004. Lecture Notes in Computer Science*, Toronto, Canada, 2004, pp. 97–115.

[65] M. U. Celik, "Universal image steganalysis using rate-distortion curves," in *Proc. SPIE 5306, Security, Steganography, and Watermarking of Multimedia Contents VI*, San Jose, California, USA, 2004, vol. 5306, pp. 467–476.

[66] R. Benton and H. Chu, "Soft computing approach to steganalysis of LSB embedding in digital images," in *ITRE 2005. 3rd International Conference on Information Technology: Research and Education, 2005.*, Hsinchu, Taiwan, 2005, pp. 105–109.

[67] J. Fridrich and M. Goljan, "On estimation of secret message length in LSB steganography in spatial domain," in *Proc. SPIE 5306, Security, Steganography, and Watermarking of Multimedia Contents VI*, San Jose, California, USA, 2004, pp. 23–34.

[68] A. D. Ker and B. Rainer, "Revisiting Weighted Stego-Image Steganalysis," in *Proc. SPIE 6819, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, San Jose, California, USA, 2008, vol. 6819, pp. 1–17.

[69] Xiang-dong Chen, Feng Sun, and Wei Sun, "Detect LSB Steganography with Bit Plane Randomness Tests," in *2006 6th World Congress on Intelligent Control and Automation*, Dalian, China, 2006, pp. 10306–10309.

[70] H. B. Kekre, A. A. Athawale, and S. A. Patki, "Steganalysis of LSB Embedded Images Using Gray Level Co- Occurrence Matrix images," *International Journal of Image Processing (IJIP),* vol. 5, no. 1, pp. 36–45, 2011.

[71] L. Fillatre, "Adaptive Steganalysis of Least Significant Bit Replacement in Grayscale Natural Images," *IEEE Transactions on Signal Processing*, vol. 60, no. 2, pp. 556–569, Feb. 2012.

[72] "BOSS Web page." [Online]. Available: http://agents.fel.cvut.cz/stegodata/. [Accessed: 28-Nov-2017].

[73] J. Fridrich and J. Kodovský, "Steganalysis of LSB replacement using

parity-aware features," in *Kirchner M., Ghosal D. (eds) Information Hiding. IH 2012. Lecture Notes in Computer Science*, Berkeley, CA, USA, 2013, vol. 7692, pp. 31–45.

[74] S. Verma, S. Sood, and S. K. Ranade, "Relevance of Steganalysis using DIH on LSB Stegnography," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 4, no. 2, pp. 835–838, 2014.

[75] D. Zou, Y. Q. Shi, W. Su, and G. Xuan, "Steganalysis based on Markov model of thresholded prediction-error image," in *2006 IEEE International Conference on Multimedia and Expo*, Toronto, Ont., Canada, 2006, vol. 2006, pp. 1365–1368.

[76] H. Malekmohamadi and S. Ghaemmaghami, "Steganalysis of LSB based image steganography using spatial and frequency domain features," in *2009 IEEE International Conference on Multimedia and Expo*, New York, NY, USA, 2009, pp. 1744–1747.

[77] T. Zhang, W. Li, Y. Zhang, E. Zheng, and X. Ping, "Steganalysis of LSB matching based on statistical modeling of pixel difference distributions," *Information Sciences*, vol. 180, no. 23, pp. 4685–4694, 2010.

[78] G. Gul and F. Kurugollu, "A new methodology in steganalysis: Breaking highly undetectable steganograpy (HUGO)," in *Filler T., Pevný T., Craver S., Ker A. (eds) Information Hiding. IH 2011. Lecture Notes in Computer Science*, Prague, Czech Republic, 2011, vol. 6958, pp. 71–84.

[79] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, 2012.

[80] T. Pevny, J. Fridrich, and A. D. Ker, "From Blind to Quantitative Steganalysis," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 445–454, Apr. 2012.

[81] R. Cogranne and F. Retraint, "An Asymptotically Uniformly Most Powerful Test for LSB Matching Detection," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 3, pp. 464–476, Mar. 2013.

[82] "BOWS-2 Web page." [Online]. Available: http://bows2.ec-lille.fr/. [Accessed: 28-Nov-2017].

[83] A. D. Ker, "Steganalysis of LSB matching in grayscale images," *IEEE Signal Processing Letters*, vol. 12, no. 6, pp. 441–444, Jun. 2005.

[84] J. Zhang, I. J. Cox, and G. Doerr, "Steganalysis for LSB Matching in Images with High-frequency Noise," in *2007 IEEE 9th Workshop on Multimedia Signal Processing*, Crete, Greece, 2007, pp. 385–388.

[85] V. Holub and J. Fridrich, "Random Projections of Residuals for Digital Image Steganalysis," *IEEE Transactions on Information Forensics and*

Security, vol. 8, no. 12, pp. 1996–2006, Dec. 2013.

[86] Z. Xia, X. Wang, X. Sun, and B. Wang, “Steganalysis of least significant bit matching using multi-order differences,” *Security and Communication Networks*, vol. 7, no. 8, pp. 1283–1291, Aug. 2014.

[87] Z. Xia, X. Wang, X. Sun, Q. Liu, and N. Xiong, “Steganalysis of LSB matching using differences between nonadjacent pixels,” *Multimedia Tools and Applications*, vol. 75, no. 4, pp. 1947–1962, Feb. 2016.

[88] “NRCS Photo Gallery Home.” [Online]. Available: https://photogallery.sc.egov.usda.gov/res/sites/photogallery/. [Accessed: 28-Nov-2017].

[89] X. Chen, G. Gao, D. Liu, and X. Zhihua, “Steganalysis of LSB matching using characteristic function moment of pixel differences,” *China Communications*, vol. 13, no. 7, pp. 66–73, Jul. 2016.

[90] D. Lerch-Hostalot and D. Megías, “Unsupervised steganalysis based on artificial training sets,” *Engineering Applications of Artificial Intelligence*, vol. 50, pp. 45–59, 2016.

[91] O. Juarez-Sandoval, M. Cedillo-Hernandez, G. Sanchez-Perez, K. Toscano-Medina, H. Perez-Meana, and M. Nakano-Miyatake, “Compact image steganalysis for LSB-matching steganography,” in *2017 5th International Workshop on Biometrics and Forensics (IWBF)*, Coventry, UK, 2017, pp. 1–6.

[92] “UCIDv2.0.” [Online]. Available: http://jasoncantarella.com/downloads/. [Accessed: 30-Nov-2017].

[93] L. M. Marvel and C. G. Boncelet, “Methodology of Spread-Spectrum Image Steganography,” Aberdeen, 1998.

[94] J. J. Harmsen and W. A. Pearlman, “Steganalysis of additive-noise modelable information hiding,” in *Proc. SPIE 5020, Security and Watermarking of Multimedia Contents V*, Santa Clara, CA, USA, 2003, pp. 131–142.

[95] R. Chandramouli and K. P. Subbalakshmi, “Active steganalysis of spread spectrum image steganography,” in *Proceedings of the 2003 International Symposium on Circuits and Systems, 2003. ISCAS ’03.*, Bangkok, Thailand, 2003, vol. 3, pp. 830–833.

[96] Ying Wang and P. Moulin, “Steganalysis of block-DCT image steganography,” in *IEEE Workshop on Statistical Signal Processing*, St. Louis, MO, USA, 2003, pp. 339–342.

[97] R. Ji, H. Yao, S. Liu, L. Wang, and J. Sun, “A new steganalysis method for adaptive spread spectrum steganography,” in *2006 International Conference on Intelligent Information Hiding and Multimedia*, Pasadena, CA, USA, 2006, pp. 365–368.

[98] K. Sullivan, U. Madhow, S. Chandrasekaran, and B. S. Manjunath, “Steganalysis of spread spectrum data hiding exploiting cover memory,” in *Proc. SPIE 5681, Security, Steganography, and Watermarking of Multimedia Contents VII*, San Jose, California, USA, 2005, pp. 38–46.

[99] Ming Li, M. K. Kulhandjian, D. A. Pados, S. N. Batalama, and M. J. Medley, “Extracting Spread-Spectrum Hidden Data From Digital Media,” *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 7, pp. 1201–1210, Jul. 2013.

[100] S. Liu, H. Yao, and W. Gao, “Steganalysis of data hiding techniques in wavelet domain,” in *Coding and Computing, 2004. Proceedings. ITCC 2004.*, Las Vegas, NV, USA, 2004, vol. 1, pp. 751–754.

[101] B. Chen and G. W. Wornell, “Quantization index modulation: a class of provably good methods for digital watermarking and information embedding,” *IEEE Transactions on Information Theory*, vol. 47, no. 4, pp. 1423–1443, May 2001.

[102] S. Liu, H. Yao, and W. Gao, “Steganalysis based on wavelet texture analysis and neural network,” in *Fifth World Congress on Intelligent Control and Automation (IEEE Cat. No.04EX788)*, Hangzhou, China, 2004, vol. 5, pp. 4066–4069.

[103] K. Sullivan, Z. Bi, U. Madhow, S. Chandrosekaran, and B. S. Manjunath, “Steganalysis of quantization index modulation data hiding,” in *2004 International Conference on Image Processing, 2004. ICIP ’04.*, Singapore, Singapore, 2004, vol. 2, pp. 1165–1168.

[104] Y. Q. Shi, C. Chen, and W. Chen, “A Markov process based approach to effective attacking JPEG steganography,” in *Camenisch J.L., Collberg C.S., Johnson N.F., Sallee P. (eds) Information Hiding. IH 2006. Lecture Notes in Computer Science*, Berlin, Heidelberg, 2007, vol. 4437 LNCS, pp. 249–264.

[105] A. Westfeld, “Generic Adoption of Spatial Steganalysis to Transformed Domain,” in *Solanki K., Sullivan K., Madhow U. (eds) Information Hiding. IH 2008. Lecture Notes in Computer Science*, Berlin, Heidelberg, 2008, pp. 161–177.

[106] J. Kodovský and J. Fridrich, “Quantitative steganalysis of LSB embedding in JPEG domain,” in *Proceedings of the 12th ACM workshop on Multimedia and security - MM&Sec ’10*, Roma, Italy, 2010, pp. 187–198.

[107] Q. Liu, A. H. Sung, M. Qiao, Z. Chen, and B. Ribeiro, “An improved approach to steganalysis of JPEG images,” *Information Sciences*, vol. 180, no. 9, pp. 1643–1655, May 2010.

[108] M. Sheikhan, M. S. Moin, and M. Pezhmanpour, “Blind image steganalysis via joint co-occurrence matrix and statistical moments of contourlet transform,” in *2010 10th International Conference on Intelligent Systems*

Design and Applications, Cairo, Egypt, 2010, pp. 368–372.

[109] J. Kodovsky, J. Fridrich, and V. Holub, “Ensemble Classifiers for Steganalysis of Digital Media,” *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 432–444, Apr. 2012.

[110] A. Westfeld, “F5 — A Steganographic Algorithm High Capacity Despite Better Steganalysis,” in *Moskowitz I.S. (eds) Information Hiding. IH 2001. Lecture Notes in Computer Science*, Pittsburgh, PA, USA, 2001, vol. 2137, pp. 289–302.

[111] K. Solanki, A. Sarkar, and B. S. Manjunath, “YASS: Yet another steganographic scheme that resists blind steganalysis,” in *Furon T., Cayre F., Doërr G., Bas P. (eds) Information Hiding. IH 2007. Lecture Notes in Computer Science*, 2007, vol. 4567, pp. 16–31.

[112] Sallee P., “Model-Based Steganography,” in *Digital Watermarking*, vol. 2939, R. Y. M. Kalker T., Cox I., Ed. Springer, Berlin, Heidelberg, 2004, pp. 154–167.

[113] S. Cho, B.-H. Cha, M. Gawecki, and C.-C. Jay Kuo, “Block-based image steganalysis: Algorithm and performance evaluation,” *Journal of Visual Communication and Image Representation*, vol. 24, no. 7, pp. 846–856, 2013.

[114] N. V. S. Sree Rathna Lakshmi, “A novel steganalytic algorithm based on III level DWT with energy as feature,” *Research Journal of Applied Sciences, Engineering and Technology*, vol. 7, no. 19, pp. 4100–4105, 2014.

[115] V. Holub and J. Fridrich, “Low-Complexity Features for JPEG Steganalysis Using Undecimated DCT,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, pp. 219–228, Feb. 2015.

[116] H. Farid, “Detecting hidden messages using higher-order statistical models,” in *Proceedings. International Conference on Image Processing*, Rochester, NY, USA, 2002, vol. 2, pp. 905–908.

[117] S. Lyu and H. Farid, “Steganalysis using color wavelet statistics and one-class support vector machines,” in *SPIE 5306, Security, Steganography, and Watermarking of Multimedia Contents VI*, San Jose, California, USA, 2004, pp. 35–45.

[118] S. Trivedi and R. Chandramouli, “Active steganalysis of sequential steganography,” in *Proc. SPIE 5020, Security and Watermarking of Multimedia Contents V*, Santa Clara, CA, USA, 2003, vol. 5020, no. 13, pp. 123–130.

[119] P. Lafferty and F. Ahmed, “Texture-based steganalysis: results for color images,” in *Mathematics of Data/Image Coding, Compression, and Encryption VII, with Applications*, Denver, USA, 2004, pp. 145–151.

[120] G. Xuan, Y. Q. Shi, J. Gao, D. Zou, C. Yang, Z. Zhang, P. Chai, C. Chen,

and W. Chen, “Steganalysis Based on Multiple Features Formed by Statistical Moments of Wavelet Characteristic Functions,” Springer Berlin Heidelberg, 2005, pp. 262–277.

[121] Y. Q. Shi, G. Xuan, D. Zou, J. Gao, C. Yang, Z. Zhang, P. Chai, W. Chen, and C. Chen, “Image steganalysis based on moments of characteristic functions using wavelet decomposition, prediction-error image, and neural network,” in *2005 IEEE International Conference on Multimedia and Expo*, Amsterdam, Netherlands, 2005, vol. 2005, pp. 269–272.

[122] Wen-Nung Lie and Guo-Shiang Lin, “A feature-based classification technique for blind image steganalysis,” *IEEE Transactions on Multimedia*, vol. 7, no. 6, pp. 1007–1020, Dec. 2005.

[123] S. Lyu and H. Farid, “Steganalysis Using Higher-Order Image Statistics,” *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 1, pp. 111–119, Mar. 2006.

[124] Xiaochuan Chen, Yunhong Wang, Tieniu Tan, and Lei Guo, “Blind Image Steganalysis Based on Statistical Analysis of Empirical Matrix,” in *18th International Conference on Pattern Recognition (ICPR’06)*, Hong Kong, China, 2006, pp. 1107–1110.

[125] Z. Sun, M. Hui, and C. Guan, “Steganalysis Based on Co-occurrence Matrix of Differential Image,” in *2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, Harbin, China, 2008, pp. 1097–1100.

[126] H. Zhao, H. Wang, and M. K. Khan, “Steganalysis for palette-based images using generalized difference image and color correlogram,” *Signal Processing*, vol. 91, no. 11, pp. 2595–2605, 2011.

[127] H. Zong, F. Liu, and X. Luo, “Blind image steganalysis based on wavelet coefficient correlation,” *Digital Investigation*, vol. 9, no. 1, pp. 58–68, 2012.

[128] S. Ghanbari, M. Keshtegary, and N. Ghanbari, “New Steganalysis Method using GLCM and Neural Network,” *International Journal of Computer Applications*, vol. 42, no. 7, pp. 46–50, 2012.

[129] Z. Zhang, D. Hu, Y. Yang, and B. Su, “A Universal Digital Image Steganalysis Method Based on Sparse Representation,” in *2013 Ninth International Conference on Computational Intelligence and Security*, Leshan, China, 2013, pp. 437–441.

[130] M. Devi and N. Sharma, “Improvements of Steganography Parameter in Binary Images and JPEG Images against Steganalysis,” *International Journal Of Engineering Sciences & Research Technology*, vol. 2, no. 8, 2013.

[131] A. K. Verma, “A Non- Blind Steganalysis Through Neural Network Approach,” *International Journal of Multidisciplinary Consortium*, vol. 1, no.

1, 2014.

[132] J. Lu, F. Liu, and X. Luo, “Selection of image features for steganalysis based on the Fisher criterion,” *Digital Investigation*, vol. 11, no. 1, pp. 57–66, 2014.

[133] W. Tang, H. Li, W. Luo, and J. Huang, “Adaptive steganalysis based on embedding probabilities of pixels,” *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 4, pp. 734–744, 2016.

[134] Y. Qian, J. Dong, W. Wang, and T. Tan, “Deep learning for steganalysis via convolutional neural networks,” in *Proc. SPIE 9409, Media Watermarking, Security, and Forensics 2015, 94090J*, San Francisco, USA, 2015.

[135] M. B. Desai, S. V Patel, and B. Prajapati, “ANOVA and Fisher Criterion based Feature Selection for Lower Dimensional Universal Image Steganalysis,” *International Journal of Image Processing*, vol. 10, no. 3, pp. 145–160, 2016.

[136] “CorelDraw image database.” [Online]. Available: http://www.corel.com/. [Accessed: 30-Nov-2017].

[137] “UC Berkeley Computer Vision Group.” [Online]. Available: https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/. [Accessed: 30-Nov-2017].

[138] J.-F. Couchot, R. Couturier, C. Guyeux, and M. Salomon, “Steganalysis via a Convolutional Neural Network using Large Convolution Filters for Embedding Process with Same Stego Key,” *arXiv:1605.07946*, pp. 1–24, May 2016.

[139] H. Sajedi, “Steganalysis based on steganography pattern discovery,” *Journal of Information Security and Applications*, vol. 30, pp. 3–14, Oct. 2016.

[140] V. Rostami and A. S. Khiavi, “Particle Swarm Optimization based feature selection with novel fitness function for image steganalysis,” in *2016 Artificial Intelligence and Robotics (IRANOPEN)*, Qazvin, Iran, 2016, pp. 109–114.

[141] S. Wu, S. Zhong, and Y. Liu, “Deep residual learning for image steganalysis,” *Multimedia Tools and Applications*, Springer US, pp. 1–17, 15-Feb-2017.

[142] J. Ye, J. Ni, and Y. Yi, “Deep Learning Hierarchical Representations for Image Steganalysis,” *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2545–2557, Nov. 2017.

[143] R. Nouri and A. Mansouri, “Digital image steganalysis based on the reciprocal singular value curve,” *Multimedia Tools and Applications*, vol. 76, no. 6, pp. 8745–8756, Mar. 2017.

[144] M. Yedroudj, F. Comby, and M. Chaumont, “Yedrouj-Net: An efficient CNN for spatial steganalysis,” in *CASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, Calgary, Alberta, Canada, 2018, pp. 2092–2096.

[145] “Caffe | Scale Layer.” [Online]. Available: https://caffe.berkeleyvision.org/tutorial/layers/scale.html. [Accessed: 10-May-2020].

[146] Z. Jin, Y. Yang, Y. Chen, and Y. Chen, “IAS-CNN: Image adaptive steganalysis via convolutional neural network combined with selection channel,” *International Journal of Distributed Sensor Networks*, vol. 16, no. 3, 2020.

[147] T. Dietterich, “Overfitting and Undercomputing in Machine Learning,” *ACM Computing Surveys*, vol. 27, no. 3, 1995.

[148] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, “CNN features off-the-shelf: An astounding baseline for recognition,” *arXiv:1403.6382*, 2014.

[149] “Convolutional Neural Networks Tutorial in PyTorch - Adventures in Machine Learning,” 2018. [Online]. Available: https://adventuresinmachinelearning.com/convolutional-neural-networks-tutorial-in-pytorch/. [Accessed: 27-Apr-2020].

[150] J. Yang, Y.-Q. Shi, E. K. Wong, and X. Kang, “JPEG Steganalysis Based on DenseNet,” *arXiv:1711.09335*, Nov. 2017.

[151] L. Pibre, J. Pasquet, D. Ienco, and M. Chaumont, “Deep learning is a good steganalysis tool when embedding key is reused for different images, even if there is a cover sourcemismatch,” in *Media Watermarking, Security, and Forensics*, San Francisco, California, 2016, pp. 1–11.

[152] B. Bayar and M. C. Stamm, “A Deep Learning Approach to Universal Image Manipulation Detection Using a New Convolutional Layer,” in *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security - IH&MMSec ’16*, New York, NY, USA, 2016, pp. 5–10.

[153] G. Xu, H.-Z. Wu, and Y.-Q. Shi, “Structural Design of Convolutional Neural Networks for Steganalysis,” *IEEE Signal Processing Letters*, vol. 23, no. 5, pp. 708–712, May 2016.

[154] Y. Qian, J. Dong, W. Wang, and T. Tan, “Learning and transferring representations for image steganalysis using convolutional neural network,” in *2016 IEEE International Conference on Image Processing (ICIP)*, Phoenix, AZ, USA, 2016, pp. 2752–2756.

[155] F. Yu and V. Koltun, “Multi-Scale Context Aggregation by Dilated Convolutions,” in *International Conference on Learning Representations*

2016, San Huan, Puerto Rico, 2016.

[156] Y. Li, X. Zhang, and D. Chen, “CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 1091–1100.

[157] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” in *ICML’15: Proceedings of the 32nd International Conference on Machine Learning*, Lille, France, 2015, pp. 448–456.

[158] N. Srivastava, G. Hinton, A. Krizhevsky, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[159] Dong-Hyun Kim and Hae-Yeoun Lee, “Convolutional Neural Network-based Steganalysis on Spatial Domain,” *INTERNATIONAL JOURNAL OF MATHEMATICS AND COMPUTERS IN SIMULATION*, vol. 11, pp. 225–229, 2017.

[160] E. Ragusa, P. Gastaldo, and R. Zunino, “Fast Transfer Learning for Image Polarity Detection,” in *Oneto L., Navarin N., Sperduti A., Anguita D. (eds) Recent Advances in Big Data and Deep Learning. INNSBDDL 2019. Proceedings of the International Neural Networks Society*, Sestri Levante, Genova, Italy, 2019, pp. 27–37.

[161] A. B. Risum and R. Bro, “Using deep learning to evaluate peaks in chromatographic data,” *Talanta*, vol. 204, pp. 255–260, Nov. 2019.

[162] E. Casilari, R. Lora-Rivera, and F. García-Lagos, “A wearable fall detection system using deep learning,” in *Wotawa F., Friedrich G., Pill I., Koitz-Hristov R., Ali M. (eds) Advances and Trends in Artificial Intelligence. From Theory to Practice. IEA/AIE 2019. Lecture Notes in Computer Science*, Graz, Austria, 2019, vol. 11606, pp. 445–456.

[163] C. Kofler, R. Muhr, and G. Spöck, “Classifying image stacks of specular silicon wafer back surface regions: Performance comparison of CNNs and SVMs,” *Sensors (Switzerland)*, vol. 19, no. 9, May 2019.

[164] S. Potluri, S. Ahmed, and C. Diedrich, “Convolutional neural networks for multi-class intrusion detection system,” in *Groza A., Prasath R. (eds) Mining Intelligence and Knowledge Exploration. MIKE 2018. Lecture Notes in Computer Science*, Cluj-Napoca, Romania, 2018, vol. 11308, pp. 225–238.

[165] N. F. Lepora, A. Church, C. De Kerckhove, R. Hadsell, and J. Lloyd, “From Pixels to Percepts: Highly Robust Edge Perception and Contour Following Using Deep Learning and an Optical Biomimetic Tactile Sensor,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2101–2107, Apr. 2019.

[166] “DDE Download Section.” [Online]. Available: http://dde.binghamton.edu/download/. [Accessed: 07-Nov-2019].

[167] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, “Deep Feature Extraction and Classification of Hyperspectral Images Based on Convolutional Neural Networks,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 10, pp. 6232–6251, Oct. 2016.

[168] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[169] “Random Forest Classification - Towards Data Science.” [Online]. Available: https://towardsdatascience.com/random-forest-classification-and-its-implementation-d5d840dbead0. [Accessed: 28-Apr-2020].

[170] D. R. Cutler, T. C. Edwards, K. H. Beard, A. Cutler, K. T. Hess, J. Gibson, and J. J. Lawler, “Random forests for classification in ecology.,” *Ecology*, vol. 88, no. 11, pp. 2783–92, Nov. 2007.

[171] K. He, X. Zhang, S. Ren, and J. Sun, “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, 2015, pp. 1026–1034.

[172] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, PMLR 9*, Sardinia, Italy, 2010, pp. 249–256.

Appendix

Table A.1: Synoptic presentation of LSB methods.

Authors - Ref	Year	Database	# of images	Method	Accuracy – Detection rate – Error rate
Westfeld and Pfitzmann [42]	2000	No database used	5	Chi-squared detects of POVs	Various tests depending on steganography tool (Steganos, S-Tools, Jsteg, EZStego) and size of embedding message
Westfeld [43]	2003	No database used	7	Chi-squared detects of POVs	Various tests for 10 versions per true colour image with different steganographic message sizes
Fridrich et al. [45]	2000	Color images, 350x250 pixels, stored as JPEGs	300	Raw Quick Pairs method. Statistical analysis of the image colors in the RGB cube	Various tests showing threshold and error probability for several different test message sizes and different secret message sizes.
Fridrich et al. [46]	2001	No database used	3	RS steganalysis	Various tests and results depending on initial bias, steganographic tool (Steganos, S-Tools, Hide4PGP) and image used (its size).
Avcibas et al. [51]	2001	Images were obtained from (1)	1800	Similarity measures between 7^{th} and 8^{th} plane	Various tests and results depending on embedding percentage (1%-15%) and steganographic scheme (Outguess-, Outguess+, F5, LSB, LSB±).
Lyu et al. [52]	2003	Images were obtained from (1)	1800	Higher order Statistics – SVM Classifier	Various tests and results depending on embedding message length (32x32-256x256) and steganographic scheme (JSteg, Outguess-, Outguess+, EZStego,LSB) and classification method (Fisher Linear discriminant analysis or SVM).

Authors - Ref	Year	Database	# of images	Method	Accuracy – Detection rate – Error rate
Dumitrescu et al. [53]–[55]	2003	No database used	29	Finite state machine	Various tests and results depending on embedding message length
Roue et al. [56]	2004	Kodak image database (2)	108	Marginal and joint probabilistic distributions of the image	Accuracy 70%
Lu et al. [57]	2004	No database used	4	Finite state machine with a new least square estimation	Various tests and results depending on embedding message length
Avcibas et al. [58]	2005	Images obtained from (3)	22	Analysis of Variance (ANOVA) - multivariate regression	Performance varies from 75% to 100% depending the watermarking algorithm used.
Dumitrescu et al. [59]	2005	Same as used in [26-28] plus ten colored high-resolution (2310x1814) uncompressed scanned images	39	High-order statistics of the samples	Various tests and results depending on embedding message length
Li Zhi et al. [60]	2003	No database used	4	Gradient Energy-Flipping Rate Detection (GEFR)	Various tests and results depending on embedding rate.
Zhang and Ping [61]	2003	USC-SIP1 Image database (4) CBIR Image Database (5)	5	Translation coefficients between difference image histograms	Various tests and results depending on embedding ratio.
Fridrich et al. [62]	2003	Color GIF images by 4 different digital cameras, stored as high-quality JPEG images and later resampled to 800×600 px.	180	Pairs Analysis	Various tests and results

Authors - Ref	Year	Database	# of images	Method	Accuracy – Detection rate – Error rate
Ker [64]	2004	2200 uncompressed images, 512 × 512. 5000 JPEG images, 900×600. 10000 JPEG images, sizes varying between 890 × 560 and 1050×691. 7500 JPEG images of very variable quality	24700	Improved RS & Pair Analysis	Various tests and results depending on embedding message length
Celik et al. [65]	2004	Kodak Photo CD Images (2)	108	Feature set based on rate-distortion characteristics of images. Bayes classifier preceded by a Karhunen-Loeve transform	Various tests and results depending on embedding rate (0 bpp – 1.0bpp)
Benton and Chu [66]	2005	No database used	1000	RS for feature extraction. DT and ANN for classification.	Various tests and results depending on embedding rate and classification method (decision tree & neural network).
Fridrich et al. [67]	2004	Images from a digital camera, downsampled from original 2272×1704 resolution to 800×600 and converted to grayscale.	60	Estimation of hidden message via weighted stego image	Various tests and results.
Ker et al. [68]	2008	1,600 raw digital camera images. 3,000 NRCS images (6). 1040 images supplied by Binghamton University	5640	Improved new weighted stego estimators	Various tests and results depending on image set and statistical measures (IRQ, Mean Error, Mean absolute error)

Authors - Ref	Year	Database	# of images	Method	Accuracy – Detection rate – Error rate
Bhattacharyya et al. [30]	2011	No database used	20	Auto-regressive model and SVM classifier	Various tests and results depending on embedding rate (0.01bpp-1.0bpp)
H.B.Kekre et al. [70]	2011	BMP images (30 color and 30 grayscale) of size 128 x 128	60	Feature vectors derived from GLCM. Euclidean distance as classification metric.	Various tests showing detection accuracy per feature per embedding length
Fillatre [71]	2012	BOSSBase v0.92 (7)	9074	Adaptive statistical test based on the likelihood ratio.	Various statistical tests concerning BOSSBase images and comparison with other methods.
Fridrich et al [73]	2013	BOSSBase v0.92 (7)	9074	Machine learning detector utilizing co-occurrences of neighboring noise residuals as features.	Various tests and results concerning average detection error for different versions of the rich model, dependence on the change rate for two selected quality factors etc.

Table A.2: Synoptic presentation of LSB Matching steganalysis methods.

Authors- Ref	Year	Database	# of images	Method	Accuracy – Detection rate – Error rate
Zou et al [75]	2006	2812 images from Vision Research Lab (8). 1096 images included in the CorelDRAW Version 10.0 software CD#3 (9)	3908	2-D Markov chain of thresholded prediction-error image along with horizontal, vertical and diagonal directions serve as features. SVM with linear and non-linear kernel	52.28% for 0.01 bpp embeding rate – 97.75% for 0.1 bpp
Malekmoham-adi et al [76]	2009	Grayscale images taken from USC-SIPI (4)	200	Gabor filter coefficients and statistics of the gray level co-occurrence matrix of images as features. SVM as classifier	94.50% average detection rate for claean and stego images. Embedding rate 0.141 bpp.
Pevny et al. [48]	2010	9200 raw images from digital camera. BOWS2 (10) (10700 images). NRCS (6) (1576 raw scans of film converted to grayscale sized 2100 × 1500) - JPEG85 (9200 images from camera compressed by JPEG with qf 85). JOINT (images from all four databases above, 30800 images)	30800 at topmost	Local dependences between differences of neighboring cover elements are modeled as a Markov chain, whose empirical probability transition matrix is taken as a feature vector. SVM as classifier	0.08 – 0.057 error rate when Embedding rate is 0.25bpp. 0.02 – 0.026 error rate when Embedding rate is 0.50bpp.

Authors - Ref	Year	Database	# of images	Method	Accuracy – Detection rate – Error rate
Zhang et al. [77]	2010	NRCS image database (6)	3185 TIFF images converted to grayscale.	Statistical modeling of pixel difference distributions.	Embedding rate 50% - 100%: 68.48% - 98.27% True Positive respectively
Fridrich et al. [49]	2011	BOSSBase v0.92 (7)	9074 grayscale images	33963 feature vector, along with the use of ensemble classifiers obtained by fusing decisions of simple detectors implemented using the Fisher linear discriminant.	Embedding rates from 0.1bpp – 0.5bpp. Error rate from 21.0% to 7.3% respectively.
Gul et al. [78]	2011	BOSSBase v 0.92 (7) BOSSRank image set (7)	10074 images	Features extracted by applying a function to the image constructing the k variate PDF estimates, and downsampling it by a suitable downsampling algorithm. Linear & SVM classifier.	Accuracy 85% when using a SVM.
Fridrich et al. [79]	2012	BOSSBase v0.92 (7)	9074 grayscale images	Rich image models combined with ensemble classifiers	Payload from 0.0.5bpp – 0.40bpp. Error estimates on Mean Absolute Deviation from 0.065 – 0.0035.
Pevny et al. [80]	2012	Raw images from digital camera converted to grayscale and to JPEG (qf 80)	9200	Feature vector extracted from the investigated object and the embedding change rate. Support vector regression was utilized then.	Various tests and results depending on embedding rate and comparison to prior art

Authors - Ref	Year	Database	# of images	Method	Accuracy – Detection rate – Error rate
Cogranne et al. [81]	2013	BOSSBase v0.92 (7) BOWS Database (10)	9074 raw images 10000 images	Generalized likelihood ratio test.	Various tests and results
Holub et al. [85]	2013	BOSSBase v1.01 (7)	10000 raw images	Projection of neighboring residual samples onto a set of random vectors. Histogram of the projections was used as feature vector.	Various test along with the detection error for different embedding rate and three different content adaptive steganographic algorithms in the spatial domain
Xia et al. [86]	2014	NRCS (6) 3161 images each of them was split to four other in order and converted to grayscale BOSSBase v0.92 (7) - 9074 raw images	12644	Co-occurrence matrix was used to model the differences with the small absolute value to extract features. SVM as classifier.	Various test concerning the detection of HUGO evaluated by “detection reliability” p (p=2A-1, where A is the area below the ROC curve)
Xia et al. [87]	2016	NRCS (6) 3161 images each of them was split to four other in order and converted to grayscale BOSSBase v0.92 (7) - 9074 raw images	12644	Calculation of the center of mass (COM) of the characteristic function of difference histogram (DHCF). SVM as classifier.	Various test on different embedding rate (0.10bpp to 1.0bpp) to two different datasets, with minimized classification error as metric.

Authors - Ref	Year	Database	# of images	Method	Accuracy – Detection rate – Error rate
Goljan et al. [50]	2015	BOSSBase v1.01 (7)	10000 raw images	Method discussed in [41] along with additional features extracted by three-dimensional co-occurrences of residuals computed from all three-color channels.	Various tests for different embedding rates (0.05bpp to 0.5bpp) with average detection error as metric, on variations of BOSSBase dataset and its grayscale versions.
Chen et al. [89]	2016	BOSSBase (7) – 10000 images NRCS (6) – 3161 images	13161	Calculation of the difference histogram characteristic function (DHCF) and the moment of DHCFs and used them as features. Features were calibrated by decreasing the influence of image content on them. SVM classifier	Various test for embedding rate 0.25bpp. Results in papers figures
Lerch-Hostalot et al. [90]	2016	BOSSBase (7)	10000	Unsupervised steganalysis method combined with artificial training sets and supervised classification.	Various tests for different embedding rates (0.1bpp, 0.2bpp, 0.25bpp, 0.4bpp) for three different steganographic algorithms. Comparison results with other methods
Sandoval et al. [91]	2017	BOWS (10) – 10000 images UCID (11) – 1338 images	11338	12 relevant features based on the probability density function (PDF) of difference of adjacent pixels and the co-occurrence matrix of the image. SVM as classifier	Various tests for different embedding rates (100%, 75%, 50%, 25%). 87.2% average detection accuracy.

Table A.3: Synoptic presentation of Spread Spectrum Image Steganography (SSIS) steganalysis methods.

Authors-Ref	Year	Database	# of images	Method	Accuracy – Detection rate – Error rate
Harmsen et al. [94]	2003	Images from Kodak PhotoCD PCD0992 (12)	24	Histogram Characteristic Function (HFC) – Center of Mass (COM). Mahalanobis distance as metric	95% accuracy at embedding rate of 1bpp
Chandramouli et al. [95]	2003	2D DCT coefficients of Lena image	1	!st technique deploys regression. 2nd technique exploits higher order statistics	45% (approx.) estimation of message bits for the first technique. 70% (approx.) estimation of message bits for the second technique.
Wang et al. [96]	2003	Lena, Jet and Baboon	3	Histograms of pixel differences. Kolmogorov–Smirnov (KS) binary hypodissertation test for classification.	Authors don't provide experimental results on large number of images.
Rongrong et al. [97]	2006	No reference by the authors.	300	Calculation of the scatter difference in both cover and it's "possible" stego image. Difference between the two scatters for classification.	Accuracy over 90%.
Sullivan et al. [98]	2005	1. digital camera images, partitioned into smaller sub-images 2. scanned photographs 3. scanned, downsampled, cropped photos 4. Images from Corel volume Scenic Sites, converted to PNG. Color images were converted to grayscale.	No reference by the authors	Markov random chain for modeling the correlation between pixels. SVm for classification.	95% accuracy
Li et al. [99]	2013	Variations of Baboon image	1	Multicarrier iterative generalized least-squares core algorithm	Authors compare their method with other ones. No experiments on a large database.

Table A.4: Synoptic presentation of Transform Domain Steganalysis methods.

Authors-Ref	Year	Database	# of images	Method	Accuracy – Detection rate – Error rate
Liu et al. [31]	2003	No reference by the authors.	125	Extract features through DFT, DCT, DWT transform. Neural network as classifier.	Average accuracy 80.2%
Liu et al. [100]	2004	Part of USC-SIPI database (4) and images acquired from digital camera and the internet.	3056	Spectrum analysis and energy differences score differences in the histograms of clean and stego images. A threshold determines whether the image was stego or clean.	Successful detection rate of 99%
Liu et al. [102]	2004	First image set includes images as Lena, Peppers etc., digital photography taken by digital camera. Second image set is from corel image database (9)	183	Statistical analysis of the texture of the image. Neural network as classifier.	Successful detection rate of 84%
Sullivan et al. [103]	2004	Digital orthophoto quarter-quadrangle aerial images, Corel PhotoCD (9) images, and images taken with a Canon digital camera.	3000	Histogram as an empirical probability mass factor (PMF) for feature extraction. Supervised learning for classification.	Various tests on each image dataset. Error rate varies from 0.001 – 0.083 when depending on images quality factor.
Shi et al. [104]	2006	No reference by the authors.	7560	Second order statistics along with threshold utilization for dimensionality reduction. SVM	Various tests depending on steganographic algorithm.

Authors - Ref	Year	Database	# of images	Method	Accuracy – Detection rate – Error rate
Kodovsky et al. [106]	2010	JPEG images acquired by 22 different digital cameras at full resolution in a raw format and then converted to grayscale.	6500	First method of estimation of change rate using the maximum likelihood principle. Second method based on minimizing a penalty function on cover images while increasing it on stego images.	Various tests and results on different estimators using median absolute error, median bias and interquartile range (IQR) as performance measures.
Liu et al. [107]	2010	Images from (13)	17051	Extended the method discussed in [60] proposing a new approach for feature extraction. SVM as classifier	Various tests and results
Sheikhan et al. [108]	2010	UCID (11) Images were converted from TIFF to JPEG.	1338	Contourlet coefficients and cooccurrence metrics of sub-band images for features extraction. SVM as classifier	Average accuracy 96.29%
Kodovsky et al. [109]	2012	Images taken from camera	6500	Ensemble classifier	Various tests and results. Median (MED) testing error over ten different splits of the CAMERA database into a training and testing set, as well as the median absolute deviation (MAD) values.
Cho et al. [113]	2013	UCID (11) INRIA Holidays dataset (14)	2829	Decomposed image blocks	Various tests and results depending on method and classifier.
Lakshmi et al. [114]	2014	No reference by the authors.	20	Authors exploited a 3-Level DWT and calculated the energy value for both training and testing dataset. SVM as classifier	Accuracy 90%

Authors - Ref	Year	Database	# of images	Method	Accuracy – Detection rate – Error rate
Holub et al. [115]	2015	BOSSBase v1.01 (7)	10000	Features derived first-order statistics of quantized noise residuals obtained from the decompressed JPEG image using 64 kernels of the DCT. Tests on selected state-of-the-art JPEG steganographic schemes.	Various tests and results depending on quality factor and steganographic scheme.

Table A.5: Synoptic presentation of Universal or Blind Steganalysis methods.

Authors-Ref	Year	Database	# of images	Method	Accuracy – Detection rate – Error rate
Farid [116]	2002	Images were obtained from (1)	1800	Wavelet-like decomposition to build higher order statistical models of natural images. Fisher linear discriminant analysis for discrimination of images	Accuracy varies from 1.3% (LSB – message length 32x32) to 94% (Jsteg – message length 256x256).
Lyu et al. [117]	2004	Natural images downloaded from www.freefoto.com	40000	Extended their work in [11] by applying their method to color images. SVM as classifier.	Various results depending on image (grayscale or color), embedded message length (from 10x10 to 80x80) and steganographic algorithm.
Lafferty et al. [119]	2004	No reference by the authors	2000	Local binary pattern texture operator as feature extractor. ANN classifier.	Various results depending on embedded message length (60 bytes to 100 bytes) and steganographic algorithm.
Xuan et al. [120]	2005	CorelDraw image database (9)	1096	Feature vector formed from the first three moments of characteristic function of wavelet sub-bands with the 3-level Haar wavelet decomposition. Bayes classifier.	Various results depending on embedded message length (10x10 to 80x80) and steganographic algorithm.
Shi et al. [121]	2005	CorelDraw image database (9)	1096	Features derived from the statistical moments of characteristic functions of the prediction-error image, the test image, and their wavelet sub-bands. ANN as classifier.	Detection rate 99.5%

Authors - Ref	Year	Database	# of images	Method	Accuracy – Detection rate – Error rate
Lie et al. [122]	2005	Variations of 132 images such as Lena, Baboon, Barbara etc.	2088	Features extracted from spatial and DCT domains. Nonlinear neural classifier.	Detection rates approx. 90%.
Farid et al. [123]	2006	Natural images downloaded from www.freefoto.com	40000	Extended their work in [69] by including phase statistics in addition to first and higher order magnitude statistics. SVM classifier.	Various results depending on quality factor (70-90, jpeg images) and steganographic algorithm used.
Chen et al. [124]	2006	CorelDraw version 11 CD#4 (9)	1349	Features extracted from projection histogram of Empirical Matrix and from prediction-error image. SVM classifier.	Detection rate 98.1%
Sun et al. [125]	2008	Grayscale images in raw format downloaded from the website of vision research lab, University of California	600	Features from co-occurrence matrices of thresholded differential images. SVM classifier.	Various tests and results depending on payload (0.1bpp – 0.3bpp) and steganographic method used (LSB, ±1). Combined Detection rate 72.2%.
Zhao et al. [126]	2011	UCID (11)	1388	Features from generalized difference images and color correlogram.	Detection rates from 61.85% to 100% depending on steganographic scheme and payload.
Zong et al. [127]	2012	NRCS (6) plus some common standard images.	2056	Method based on the correlation of inter- and intra-wavelet sub-bands in the wavelet domain and feature extraction from the co-occurrence matrix. SVM classifier	Various tests and detection rates concerning feature combination, embedding method and image size.

Authors - Ref	Year	Database	# of images	Method	Accuracy – Detection rate – Error rate
Ghanbari et al. [128]	2012	USC-SIPI (4) BSDS (15) Images from internet	800	Features extracted from the GLCM of the original image and stego image. MLP as classifier.	Accuracy 80%
Zhang et al. [129]	2013	Images were obtained from (1)	5000	Method based on sparse representation. Sparse Representation Classification algorithm. SVM classifier	Various results depending on embedding rate (25%-100%), steganographic scheme and classification method (SRC – SVM)
Devi et al. [130]	2013	No reference made by authors	5931	Method based on minimizing image-to image variations.	Various results depending on embedding rate (0%-100%).
Verma [131]	2014	Gray scale BMP images of size 256x256	60	Features extracted by GLCM. MPL with Pre-processed Vectors Diagonal Back Propagation Algorithm (PVDBPA) as classifier.	Various results depending on version of algorithm used.
Lu et al. [132]	2014	BOSSBase v1.01 (7)	5000	Feature selection method based on the Fisher criterion.	Various results depending on embedding ratio (bpp) and embedding method.
Tang et al. [133]	2016	BOSSBase v1.02 (7)	10000	Feature selection method based on the Fisher criterion, in which the separability of single-dimension and multiple dimension features, combined with measurement of the Euclidean distance, is analyzed.	Various results depending on embedding ratio (bpp) and embedding method.

Authors - Ref	Year	Database	# of images	Method	Accuracy – Detection rate – Error rate
Qian et al. [134]	2015	BOSSBase v1.01 (7) ImageNet (15) - 100000 randomly selected images)	120000	Deep Learning with convolutional neural networks (CNN)	Various tests depending on image database, embedding ratio (0.3bpp – 0.5bpp) and embedding method. Error rate as metric.
Desai et al. [135]	2016	CorelDraw (9) BSDS500 (17)	1400	A reduced dimensional merged feature set for universal image steganalysis using Fisher Criterion and ANOVA techniques was used. SVM with RBF kernel as classifier	97% detection accuracy in various steganographic methods.
Couchot et al. [138]	2016	BosBase v1.01 (7) Raise database (18)	18156	Deep Learning with convolutional neural networks (CNN)	Various tests depending on embedding method and theor different versions. Accuracy as metric.
Sajedi [139]	2016	Washington University image database (19) 3959 images were taken with six cameras with different resolutions	4959	Feature extraction via fuzzy if–then rules. SVM as classifier.	Embedding rate 0.05bpp - 0.4bpp. Average accuracy on different steganographic methods from 79% - 91%.
Rostami et al. [140]	2016	BOSSBase (7)	10000	Feature selection method based on based on optimization process of Particle Swarm Optimization (PSO) and AUC as fitness function. SVM as classifier.	Embedding rate 0.4bpp, detection accuracy 82.62%.
Wu et al. [141]	2017	BOSSBase (7)	10000	Deep residual network (DRN).	Average error rate 6.48%
Ye et al. [142]	2017	BOSSBase (7) BOWS (10)	20000	Deep Learning with convolutional neural networks	Various embedding rates. Low detection error on various steganographic algorithms.

Authors - Ref	Year	Database	# of images	Method	Accuracy – Detection ra Error rate
Nouri et al. [143]	2017	UCID (11)	506	Alteration of singular value curve was used to construct the steganalysis feature vector.	Embedding rates of 0.05, 0.1, 0.2 and 0.4 bpp. Various results on different steganographic algorithms. Comparison with other relevant feature extraction methods.

Table A.6: Dataset links.

1	Philip Greenspun	http://philip.greenspun.com/
2	KODAK	ftp://ftp.kodak.com/www/images/pcd/
3	Noname	http://www.petitcolas.net/fabien/watermarking/benchmark/image_database.html
4	USC-SIP1 Image database	http://sipi.usc.edu/services/database/Database.html
5	CBlR Image Database	http://www.cs.washington.edu/research/imagedatabase/groundtruth/
6	NRCS	http://photogallery.nrcs.usda.gov/
7	BOSSBase	http://agents.fel.cvut.cz/stegodata/
8	Noname	http://vision.ece.ucsb.edu/~sullivak/Research_imgs/
9	Corel	http://www.corel.com
10	BOWS 2	http://bows2.ec-lille.fr/
11	UCID	http://jasoncantarella.com/downloads/ http://vision.doc.ntu.ac.uk/
12	Kodak photo cd	(http://sqez.home.att.net/thumbs/Thumbnails.html)
13	Noname	http://www.cs.nmt.edu/~IA/steganalysis.html)
14	INRIA	http://lear.inrialpes.fr/~jegou/data.php
15	BSDS	http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/fg
16	ImageNet	http://www.image-net.org/
17	BSDS500	https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/
18	Raise	http://mmlab.science.unitn.it/RAISE/
19	Washington University image database	http://imagedatabase.cs.washington.edu/

Konstantinos Karampidis was born in Athens Greece in 1971. He received his B.Sc. degree in Electrical Engineering from the Technological Institute of Crete in 1994 and his master's degree in Informatics & Multimedia from the same institute in 2015. From 1999 until 2016 he was a computer science teacher to various secondary education teaching institutes. From 2015 he is a laboratory assistant (Artificial Neural Networks, Pattern Recognition, Digital Signal Processing) at the Department of Electrical and Computer Engineering in the Hellenic Mediterranean University. From 2016 he belongs to the administrative staff of the Hellenic Mediterranean University in the IT Department. His current research interests include Intelligent System applications, Deep Learning, Machine Learning, Computer Vision and Digital Forensics. He is the author of one book chapter, three Journal and seven International Conference publications. He has participated in various European Research Programs as a technician/researcher. He is a member of the Intelligent Systems and Computer Architecture Laboratory at the Department of Electrical and Computer Engineering in the Hellenic Mediterranean University. He is also a member of the Hellenic Artificial Intelligence Society (EETN) and member of the Hellenic Society of Scientists and Professional of Informatics and Communications (EPY).