VSB TECHNICAL UNIVERSITY OF OSTRAVA | FACULTY OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Advanced Methods of Detection of the Steganography Content

Pokročilé metody detekce steganografického obsahu

Jakub Hendrych

PhD Thesis

Supervisor: prof. Ing. Lačezar Ličev, CSc., prof.h.c.

Ostrava, 2022

Abstrakt a přínos práce

Steganografie může být využita k nelegálním aktivitám. Proto je velmi důležité být připraven. K detekci steganografického obrázku máme k dispozici techniku známou jako stegoanalýza. Existují různé typy stegoanalýzy v závislosti na tom, zda je znám originální nosič nebo zdali víme, jaký byl použit algoritmus pro vložení tajné zprávy. Z hlediska praktického použití jsou nejdůležitější metody "slepé stagoanalýzy", které zle aplikovat na obrazové soubory a jelikož nemáme originální nosič pro srovnání. Tato doktorská práce popisuje metodologii obrazové stegoanalýzy. V této práci je důležité porozumět chování cíleného steganografického algoritmu. Pak můžeme využít jeho slabiny ke zvýšení detekční schopnosti a úspěšnosti kategorizace. Primárně se zaměřujeme na prolomení steganografického algoritmu OutGuess2.0 a sekundárně na algoritmus F5. Analyzujeme schopnost detektoru, který využívá proces kalibrace, výpočtu shlukování a mělkou neuronovou síť k detekci přítomnosti steganografické zprávy na podezřelém snímku. Nový přístup a výsledky jsou sepsány v této doktorské práci.

Hlavní přínosy disertační práce jsou následující:

1. Detekce stegogramů vytvořených steganografickými algoritmy OutGuess2.0 a F5.
2. Vysoká úspěšnost klasifikace - s ohledem na sensitivitu testu.
3. Invariance sensitivity testu vůči velikosti tajné zprávy.
4. Podpora různých rozlišení a barevné hloubky obrázků.
5. Aplikace mělké neuronové sítě.
6. Aplikace filtrování makrobloků JPEG obrázků pro zvýšení úspěšnosti klasifikace.
7. Publikování výzkumu - všechny prezentované přínosy jsou publikovány na mezinárodní konferenci nebo v časopise.

Klíčová slova

steganografie; stegoanalýza; neuronová síť; mělká neuronová síť; ANN; JPEG; DCT; kalibrace, shlukování; OutGuess2.0; F5

Abstract and Contributions

Steganography can be used for illegal activities. It is essential to be prepared. To detect steganography images, we have a counter-technique known as steganalysis. There are different steganalysis types, depending on if the original artifact (cover work) is known or not, or we know which algorithm was used for embedding. In terms of practical use, the most important are "blind steganalysis" methods that can be applied to image files because we do not have the original cover work for comparison. This philosophiæ doctor thesis describes the methodology to the issues of image steganalysis.In this work, it is crucial to understand the behavior of the targeted steganography algorithm. Then we can use it is weaknesses to increase the detection capability and success of categorization. We are primarily focusing on breaking the steganography algorithm OutGuess2.0. and secondary on breaking the F5 algorithm. We are analyzing the detector's ability, which utilizes a calibration process, blockiness calculation, and shallow neural network, to detect the presence of steganography message in the suspected image. The new approach and results are discussed in this Ph.D. thesis.

In particular, the main contributions of the dissertation thesis are as follows:

1. Detecting stegogrammes created by OutGuess2.0 and F5 steganography algorithms.
2. High classification success rate - with regard to the sensitivity of the test.
3. Invariance of the sensitivity of the test against secret message length.
4. Different image resolution and image color depth support.
5. Application of the shallow neural network.
6. Application of the JPEG macroblock filtering for increasing the classification success rate.
7. Publications of our research - all the contributions presented here are published in international conference or journal

Keywords

steganography; steganalysis; neural network; shallow neural network; ANN; JPEG; DCT; calibration; blockiness; OutGuess2.0; F5

Acknowledgement

Firstly, I would like to express my sincere gratitude to my supervisor, prof. Ing. Lačezar Ličev, CSc., prof. h.c. for the continuous support of my Ph.D. study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D. study.

Besides my supervisor, I would like to thank prof. Ing. Ivan Zelinka, Ph.D. for giving me the chance to discover the beauty of steganography and steganalysis and provide me the opportunity (within the grant project Technology Agency of the Czech Republic - TACR Delta "Security of Mobile Devices and Communication" - TF01000091 - Development of the application for steganography of static images for the purposes of mobile communication) that led to the creation of this research.

I thank my fellow Ph.D. student, Ing. Radim Kunčický, for the stimulating discussions, for the sleepless nights we were working together before countless deadlines, and for all the fun we have had in the last several years of our study.

Last but not least, I would like to thank my friends, family, and mainly my wife Lucie for supporting me, calming me down throughout writing this thesis, my Ph.D. study, and my life in general.

Contents

List of symbols and abbreviations

1D	–	One-Dimensional
2D	–	Two-Dimensional
6D	–	Six-Dimensional
AC	–	Alternating Current
ANN	–	Artificial Neural Network
ANOVA	–	Analysis of Variance
ASCII	–	American Standard Code for Information Interchange
B	–	Byte
BC	–	Before Christ
BMP	–	Bitmap
CF	–	Colorful
DC	–	Direct Current
DCT	–	Discrete Cosine Transform
DF	–	Degrees of Freedom
DPCM	–	Differential Pulse-Code Modulation
DWT	–	Discrete Wavelet Transform
ECC	–	Elliptic Curve Cryptography
FFT	–	Fast Fourier Transform
GIF	–	Graphics Interchange Format
GLM	–	Gray Level Modification
GOP	–	Group of Pictures
GUI	–	Graphic User Interface
HVS	–	Human Vision Sensitivity
ISTR	–	International Security Threat Report
IT	–	Information Technologies
JPEG	–	Joint Photographic Experts Group
kB	–	Kilobyte
LSB	–	Least Significant Bit

MBNS	–	Multiple-Based Notational System
MBPIS	–	Multi-Bit Plane Image Steganography
MPEG	–	Moving Picture Experts Group
MS	–	Mean Square
PNG	–	Portable Network Graphics
PRNG	–	Pseudo-Random Number Generator
RS	–	Regular and Singular group
PSNR	–	Peak Signal-to-Noise Ratio
PVD	–	Pixel-Value Differencing
RLE	–	Run-Length Encoding
SPA	–	Sample Pair Analysis
SS	–	Sum of Squares
SVM	–	Support Vector Machine
WS	–	Weighted Steganalysis
YASS	–	Yet Another Steganographic Scheme

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

The term steganography refers to the art of secret communications. By using this art, it is possible for person Alice to send a secret message to person Bob in such a way that the third party does not know that the message even exists. The message is always embedded in another object known as cover work. It is done by manipulation of a cover works properties. The output is a stegogramme, which is very similar to a cover work, but it also carries the hidden message. If Alice sents Bob this stegogramme, anybody who intercepts this communication will obtain only a stegogramme. Then is a difficult task to tell that the stegogramme is not innocent. Moreover, this is the main advantage of steganography, to create an illusion of innocent communication [1].

One of the oldest examples of steganography application dates back to the 5-th century BC of Greek history. Histaeus, the ruler of Miletus, tattooed a secret message on the shaved head of his most trusted servant. After the hair had grown back, the servant was sent to Aristagorus, where his hair was shaved, and the message with commands was revealed [2]. In this example, the servant was used as the carrier for the secret message, and anyone, who saw this servant, had no suspicion that he carried a message.

The development of information technologies has brought new opportunities to apply steganography methods. In modern terms, steganography is usually implemented computationally. It means that cover work can be text files, images, audio, and video files, and a secret message is embedded within them. Imagine a company with employees and internal secret company data. The employee Alice from the customer service department received an offer from Bob. This offer is to steal information about customers from the database. If she sends this secret information to the competitor Bob via email, there is a high possibility that she will be exposed. For example, by some security email policy monitor. Then Alice will be charged for information fraud. See the first scenario in figure 1.1.

However, if Alice uses steganography encode algorithm, she can embed a text file containing confidential internal data into the prepared JPEG image (cover work). After that, she

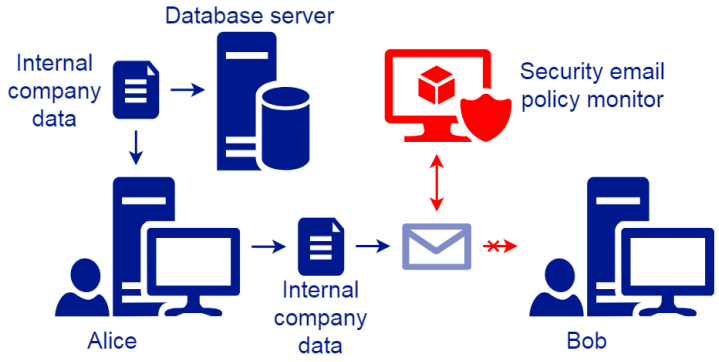


Figure 1.1: First scenario - data transport via e-mail.

sends this image (stegogramme) to Bob. Then he uses a steganography decoding algorithm on the image and retrieves the company data. See the second scenario in figure 1.2. Of course, we can example many other scenarios with a social network, internet forums, and not to mention that terrorists can use steganography [3].

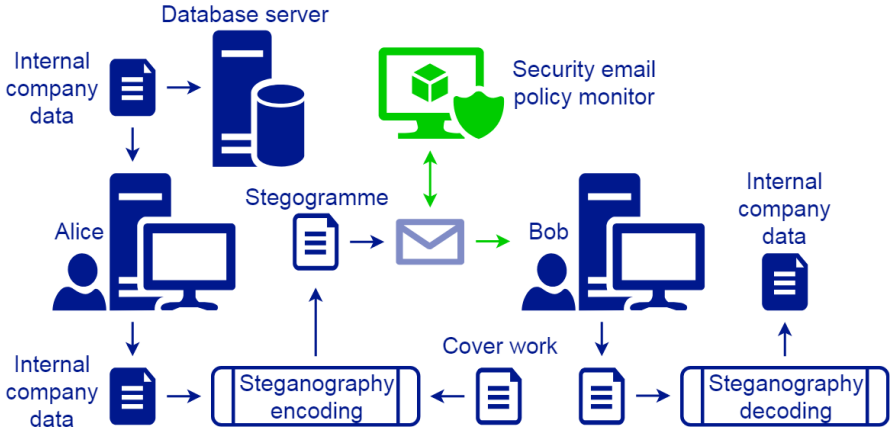


Figure 1.2: Second scenario - image with embedded data transported via e-mail.

The idea of steganography does not necessarily mean an equivalent to some illegal activity. It can also be used, like digital watermarking, to protect our data. Alternatively, it can be used as an additional data layer (alternative to metadata) without changing the content itself. A good example was the story of one Ukrainian who sold his movies. Before he sold his movie to the buyer, he put the buyer's information directly into the file using a steganography algorithm. If the copy of this movie appeared on the internet, he could precisely find out who was responsible for it. However, there is always a chance that the steganography will be misusage and that is why we must be prepared.

In recent years many steganography algorithms have been made. The simplest method is a modification of the LSB of an image pixel. Modification of the bit may be performed sequentially or randomly. However, these methods of inserting information directly into spa-

tial regions of the image are easily detectable. Therefore, new methods were developed. The information was embedded into the transform domain. The JPEG technology uses compression that converts the image into the DCT domain. This DCT domain presents the data as the high and low frequencies. High frequencies are related to the areas of an image with high details, but low frequencies are associated with low details. To reduce the size of the JPEG image, some of the high details are removed – the human eye cannot distinguish these areas. So there is no obstacle to modifying these values (DCT coefficients) to hide the secret information. This methodology is used by the steganography algorithm OutGuess2.0, on which we are primarily focusing. It is an open-source algorithm. Therefore, it will most likely be used for embedding the secret information into the picture cover work. OutGuess2.0 is also implemented with its own GUI, which does not need excellent knowledge of the steganography of the end-user. This fact also increases the likelihood of use in the corporate sector. The secondary goal is to detect the less known steganographic algorithm F5.

The counter technique of steganography is steganalysis, which serves us to detect stegogrammes. The main goal of steganalysis is to identify the presence of the secret message but not the successful extraction of the message. That is a non-trivial task because of the used cryptography method on the secret message. There are different types of steganalysis, depending on whether the cover is known or not during the analysis (known-cover attack). If we know the used steganography algorithm (known-stego attack steganalysis), it is called "targeted steganalysis". Otherwise, it is called "blind steganalysis". In terms of practical use, especially the analysis techniques of "blind steganalysis" applied to image files are essential.

The first part of this thesis is dedicated to the state of the art of steganography a steganalysis. Next, we are focusing on steganography and steganalysis in more detail. We will deal with the theory, the main principle of DCT steganography, and course, by specific algorithms - OutGuess2.0 and F5. The next chapter will describe the contribution to steganalysis, followed by detailed research. Of course, the results of the stegogramme classification and others will not be missed. The last part of this thesis is dedicated to the conclusion, where we summarize the achieved results and possible future development of our research.

In the following figure 1.3, we can practically see the similarity between both images and also the innocent look of the stegogramme, which is generated by OutGuess2.0 steganography algorithm.[1]

[1]The correct answer is at the end of this thesis.

(a) (b)

Figure 1.3: One of these images is a stegogramme with 1 kB message. Guess which one or try to find a difference.

Chapter 2

State of the Art

2.1 Steganography Techniques

The threat by image steganography is very high. Many steganography algorithms were developed, from the simplest one to a modification of image spatial domain (such as the LSB method) to more advanced algorithms that use transform domain for embedding a secret message.

2.1.1 Least Significant Bit

One of the standard techniques is based on manipulating the LSB planes by directly replacing the least significant bits of the cover image with the message bits [4, 5]. LSB methods typically achieve high capacity [6], but unfortunately, LSB insertion is vulnerable to slight image manipulation such as cropping and compression.

In computer science, the least significant bit refers to the smallest (right-most) bit of a binary sequence. The binary structure is such that each integer may only be either a 0 or a 1. Changing the LSB value from 0 to 1 does not significantly impact the final image. If we now think of each 8-bit binary sequence as a means of expressing the color of a pixel for an image, it should be clear to see that changing the LSB value from 0 to 1 will only change the color by +1. This modification cannot be noticed by a human eye [7, 8].

If we talk about the LSB steganography algorithm, we should mention that this algorithm has two different embedding variations - sequential and randomized approaches. Sequential embedding often means that the algorithm starts at the first pixel of the cover image and embeds the bits of the message data in order until there is nothing left to embed. However, the randomized approach scatters the locations of the values that will be modified to contain the bits of the secret message. The main reason for randomizing is to make this method hard to break by a steganalysis that is looking to determine whether the image is a stegogramme

or not. The simplest form of LSB steganography is the method known as Hide & Seek [2]. This algorithm also provides a randomized approach with the use of PRNG.

Various methods about the LSB steganography have been proposed in literature [9, 10, 11]. For example, authors in [12, 13, 14] proposed a new substitution scheme. In some cases, embedding the secret message into the cover works LSB of the cover image may degrade the stegogramme. So this increased the likelihood that the observer could detect that something was going on in the image. The authors proposed the method that uses a genetic algorithm to search for an approximate optimal solution with a very satisfying computation time. Also, the research [15] was made on the requirement for maximizing the embedding capacity. Authors answer the question of determining the maximal embedding capacity for each pixel. An image steganography model is proposed that is based on variable-size LSB insertion to maximize the embedding capacity while maintaining crucial factor - image fidelity [16, 17, 18]. Applying LSB techniques to color images brings new possibilities - modifications of all color components for embedding the secret message [19, 20]. Even though the LSB technique is one of the oldest computational steganography methods, it is developing is not over yet [21, 22, 23].

2.1.2 Other Techniques of Spatial Domain Modification

In 2005, authors in [24] presented an adaptive steganographic scheme with the MBNS based on HVS. The hiding capacity of each image pixel is determined by its local variation. The formula for computing the local variation considers the factor of human visual sensitivity. A tremendous local variation value indicates that the area where the pixel belongs is a high details area (such as edges), which means more confidential data can be hidden. On the contrary, less confidential data will be hidden in the image block when the local variation value is small because it is a low details area. This way, the stegogramme quality degradation is invisible to the human eye.

Authors did other exciting research in [25]. They proposed the MBPIS for the image. This method is resistant to most statistical steganalysis algorithms such as RS and pixel difference histogram analysis. Authors adjust the embedding process to be more adaptive to cover images by considering two parameters - similarity threshold (for selecting high details area in lower bit planes) and size of blocks in embedding bit planes.

Increasing embedding capacity is a crucial task for steganography algorithms. This property is the main goal for the PVD method proposed by authors in [26]. PVD method embeds a secret message into grayscale cover images. A cover image is partitioned into non-overlapping blocks of two consecutive pixels. A difference value is calculated from the values of the two pixels in every block. A more significant difference in the original pixel values allows a more extensive modification. Then the difference value is replaced by a new value of the secret message. Based on the PVD method, various approaches have also been proposed, such as

a new method [27, 28] that uses three-pixel value differencing. These approaches improve embedding capacity and lower the PSNR.

As we mentioned above in chapter 1, steganography is not always equivalent to illegal activity. Authors in [29] present an approach called GLM that can provide security and information protection through steganography with low computational complexity and high embedding capacity. The information is embedded into the gray level values of the grayscale image pixels. GLM technique uses the concept of odd and even numbers to map data within an image. A set of pixels are selected from a given image based on a mathematical function. The gray level values of those pixels are examined and compared with the message bitstream. This bitstream is going to be mapped in the image. This feature leads to a one-to-one mapping between binary data and the selected pixels. Also, authors in [30] deal with image verification techniques to prevent counterfeiting. The authors report that this method can be extended to securing video data to prevent unauthorized video editing or validation.

2.1.3 Techniques of Transform Domain Modification

As we mentioned above in chapter 1, transform domain-based algorithms are used for embedding secret message JPEG compression process. These steganography algorithms widely used transformation functions include FFT, DWT, and DCT.

A typical representative of DCT based algorithms is JSteg, presented by the author [31]. JSteg is very similar to Hide & Seek algorithm discussed in section 2.1.1. It utilizes the LSB embedding technique. JSteg embeds a secret message into a cover image by successively replacing the least significant bits of non-zero quantized DCT coefficients with secret message bits. DCT coefficient used to hide secret message bits is selected at random by a PRNG, which a key can control.

Another example of steganography DCT based algorithm is OutGuess presented by [2] in 2003. Several versions of this algorithm were implemented. When the OutGuess0.1 was developed, it was considered much more secure than the Hide & Seek and JSteg. However, after the release, steganalysts was able to find a fatal flaw in the technique that left statistical artifacts in the stegogrammes, so this algorithm was vulnerable to statistical analysis. After that, author [2] implement a new version - OutGuess2.0. The main goal of this version was to ensure that the statistical properties of the cover image were maintained after embedding, such that stegogramme looks statistically similar to a clear innocent image. The algorithm itself consists of two steps. Firstly, OutGuess2.0 randomly embeds secret message bits into the least significant bits of the quantized DCT coefficients (except 0 and 1). Secondly, corrections are then made to the rest of the coefficients after embedding. These corrections are done because the global DCT histogram of the stegogramme must be similar to the cover image. More about OutGuess2.0 will be discussed later in chapter 3.1.4.

In 2001 was introduce F5 algorithm [32]. Also, this algorithm has several versions (F3, F4, and F5). However, the F5 was designed to improve previous versions by minimizing the disturbance caused by embedding the secret message on the cover image. Instead of replacing the least significant bits of quantized DCT coefficients with the secret message bits, the absolute value of the coefficient is decreased by one if it has to be modified. The F5 algorithm embeds the secret message bits into randomly chosen DCT coefficients and employs matrix embedding that minimizes the necessary changes to hide a certain message length. During the embedding process, the message length and the number of non-zero AC coefficients are used to determine the best matrix embedding that minimizes the number of modifications of the original cover image. More about F5 will be discussed later in chapter 3.1.5.

Another algorithm that belongs to JPEG steganography is YASS presented by authors in [33]. This algorithm is unique because it does not directly embed secret message data into DCT coefficients. Cover work is first divided into large blocks with a fixed size in the spatial domain. Then within each of these blocks, a sub-block is randomly selected for performing DCT. Next, secret message data are embedded into the DCT coefficients of the sub-block. Finally, after performing the inverse DCT to the sub-blocks, an image is compressed and distributed as a JPEG image. Also, a new version was proposed by the author in [34]. This updated schema has enhanced security performance via block randomization.

The authors provide another exciting research [35]. The main challenge is how to increase the payload capacity without the cover image being detected as stegogramme. In this paper, the authors propose a large-capacity Invertible Steganography Network for image steganography. They take steganography and the recovery of hidden images as a pair of inverse problems on image domain transformation. Then introduce a single invertible network's forward and backward propagation operations to leverage the image embedding and extract problems. Also, the capacity of image steganography is significantly improved by naturally increasing the number of channels of the hidden image branch.

Similarly, the authors in [36] propose a new high-capacity image steganography method based on deep learning. DCT is used to transform the secret image, and then the transformed image is encrypted by ECC to improve the anti-detection property of the obtained image. The deep neural network with a set of hiding and extraction networks is applied to improve the steganography payload. These networks enable steganography and extraction of full-size images. Also, the image obtained using this steganography method has a higher PSNR.

As we mentioned at the start of this section, other transformation functions can be used for embedding. For example, DWT steganography is sometimes used [37, 38, 39]. However, these methods still used the standard technique of LSB modification. Except for modifying the actual pixel, the secret message data are stored into wavelet coefficients.

2.2 Steganalysis Techniques

The main goal of steganalysis is to break steganography and detect the potential secret message. Almost all steganalysis algorithms rely on the statistical differences between the cover image and stegogramme. Some steganalysis techniques rely on visual inspection to reveal the presence of the secret message. Next, techniques reveal the slightest alterations in images by observing their statistical behavior. Also, analyzing the data file format structure can lead to discovering the secret message presence.

2.2.1 Breaking the Spatial Domain Techniques

The LSB steganography algorithm has been the first and most important spatial domain technique. This fact was the reason for so many developed steganalysis algorithms. These algorithms have been proved most successful. For example, Chi-square statistical test [40, 41], RS analysis [42], WS [43], SPA [44], and structural steganalysis [45, 46].

Let us speak about breaking the MBNS technique. It is tough to observe some abnormality between the cover image and it is stegogramme through the histogram of pixel values or the histogram of pixel prediction errors. In 2008 authors [47] presented exciting research to break the MBNS technique. In the MBNS, secret data are converted into symbols in a notational system with multiple bases. Authors observed that given any base value, more small symbols are generated than large symbols in converting binary data of the secret message to symbols. They proved that the number of small remainders increases due to the steganography modification. This observation is used for the conditional probability to discriminate between the clear image and stegogramme.

For detecting the stegogrammes created by the MBPIS technique, which embeds the secret message into the multiple Gray code bit-planes, a steganalysis method was proposed by the authors [48] to estimate the embedding ratios based on sample pairs analysis. The proposed method combines appropriate trace sets of a more elegant and pellucid sample pair analysis model to estimate the modification ratio of each natural binary bit-plane. Then the modification ratios in Gray code bit-planes are estimated from the modifications ratios in natural bit-planes. Finally, the embedding ratios are obtained from the modification ratios in Gray code bit-planes. Also, authors [49] present a steganalysis scheme for MBPIS. They show how to adapt RS analysis into a local analysis to design an efficient detector against the MBPIS technique.

Researches have been done to break a PVD steganography technique in recent years. For example, in 2010, a new steganalysis method was proposed [50]. Usually, PVD is immune to conventional steganalysis methods. It performs the embedding in the difference of the values of pixel pairs. However, several characteristics are identified in the histogram of the cover

images and stegogrammes. The authors designed five distinct multilayer perceptron neural networks to detect different embedding levels with 88.6% success on the correct categorization. Also, the same authors did another research to break a PVD [51]. Next, exciting research was done by authors [52] which presents the method that exploits a severe design flaw in the data embedding procedure. Also, in 2010 novel steganalysis method was presented by [53] to detect stegogrammes created by modified PVD (MPVD) that corrects the above-mentioned design flaw.

2.2.2 Breaking the Transform Domain Techniques

Many steganalysis methods for breaking JSteg were published in recent years [54, 55]. For example, in [56, 57] authors present novel statistical test or in [58] steganalysis method is based on hypothesis test. The hypothesis is that the steganography algorithm leaves statistical evidence that can be exploited for detection with image quality features and multivariate regression analysis. To this effect, authors identified quality metrics by ANOVA hypothesis test.

If we want to break the newest version of OutGuess, we cannot use similar statistics because OutGuess preserves the shape of the histogram of DCT coefficients. The Authors in [59, 60] present a new methodology on how to detect OutGuess stegogrammes quantitatively by counting the discontinuity along the boundaries of all JPEG macroblocks. This spatial domain feature is called Blockiness, and authors proposed it in [60]. It was observed that the Blockiness value linearly increases with the number of modified DCT coefficients. Together with the calibration process, we use this feature as a base methodology for our research. Improvements and additional features are discussed later. Another interesting methodology for breaking OutGuess is [61] presented in 2004. This method for JPEG images is based on comparing JPEG steganography algorithms and evaluating their embedding mechanisms. The classification method is calculated as the difference between a specific functional calculated from the stegogramme and the same functional obtained from a decompressed, cropped, and recompressed stegogramme. Also, authors in [62] present the method of blind steganalysis based on the classifying feature vectors derived from images. This method is fascinating on the term of universality. It is capable of assigning stegogrammes to six popular steganography algorithms.

For breaking the F5 algorithm, one major flaw was exploited. F5 algorithm preserves some crucial characteristics of the histogram of DCT coefficients, such as the monotonicity and the symmetry. However, it does not modify the shape of the histogram. Authors exploited this flaw in [63] in 2002.

Even the YASS steganography technique can be attacked. Many research was published in recent years [64]. YASS is the algorithm for digital images that hides messages robustly in

a key-dependant transform domain. Authors in [65] demonstrate experimentally that twelve different settings of YASS can be reliably detected even for low embedding rates and in small images. The next exciting research [66] was published in 2011. The authors designed the features of differential neighboring joint density on the absolute array of DCT coefficients between the JPEG images and the calibrated versions. The methodology first identified blocks possibly used for embedding and the non-candidate neighbors that are impossible to use for information hiding. Then, the neighboring joint density difference between candidate block and non-candidate neighbors is obtained. For classification, the SVM is used.

Another interesting research was published in 2019 by the authors [67]. Most of the steganalysis algorithms are not effective for mismatched steganalysis. This paper proposes a method to solve the mismatched steganalysis on the internet images by domain adaptation classifier. It makes the distribution between training and testing sets more similar to obtain better detection performance. Authors integrate joint distribution adaptation and geometric structure as regularization terms to a standard supervised classifier. This method has about 85% (average) success rate in the steganography algorithm classification.

In 2020, authors [68] introduce a shallow "OneHot" CNN. This network encodes DCT coefficients using clipped one-hot encoding into a binary volumetric representation of the DCT plane fed to a convolutional block designed to learn relevant intra-block and inter-block relationships using vanilla and dilated convolutions. Methodology for plugging the "OneHot" network into conventional steganalysis CNNs is also introduced for an end-to-end learnable detector with improved performance.

In the case of OutGuess and F5 algorithms, another interesting analysis was performed by the authors [69]. The authors analyze the change in the file size by embedding a secret message. This feature might be used in steganalysis. The experiments show exciting results. OutGuess and complementary embedding methods increase the file size, while F5 decreases it. All factors are considered, such as secret message length or JPEG quality factors. Also, authors in [70] claim that there are no effective methods to recover the steganography key because it is difficult to statistically distinguish the coefficient sequences selected by true and false keys. Therefore, the author's paper proposes a method for recovering the steganography key of a JPEG image steganography. This process is not a trivial task. The author's target is the F5 algorithm, composing the check matrix and shuffling key. Firstly, the check matrix is recovered based on the embedding ratio estimated by quantitative steganalysis. The shuffling key is then recovered based on the distribution difference between the bit sequences extracted by the true and false shuffling keys. Additionally, the cardinality of the shuffling keyspace is significantly reduced by examining the extracted encoding parameter and message length. Experimental results show that the proposed method can recover the stego key accurately and efficiently. It is essential to note that the authors have pushed the boundaries of steganalysis, which primarily does not seek to extract the secret message itself.

Chapter 3

Insight into Steganography and Steganalysis

3.1 Steganography

In terms of development, steganography consists of two algorithms - encoding and decoding. The first algorithm provides embedding, and the second for extracting. The embedding algorithm is the most carefully constructed process because it ensures that the secret message goes unnoticed if someone intercepts communication and gets the stegogramme. Steganography encoders try to make a minimum distortion of cover work. The lower distortion leads to a better chance of un-detectability. For inputs are required the secret message (usually a text file that contains the transferring message) and cover work (file to construct a stegogramme that contains a secret message). The extracting algorithm, in most cases, inverse the embedding process, and the output is the secret message.

The techniques of steganography are very similar to digital watermarking. However, there is one huge difference. In digital watermarking, the emphasis is that nobody can remove or alter the content of the watermarked data, even if it is obvious they exist. However, steganography is purely focused on hiding, so it is difficult to tell that a secret message exists. There is also a difference against cryptography, which does not hide but encodes a secret message. So nobody knows the content of the secret message. In some cases, steganography systems use cryptography to encode the content of the secret message as the next layer of security. Also, some systems require a key to protect a secret message. Without a key, it would be possible for someone to correctly extract the message if they managed to break the encoding algorithm.

Suppose we want to classify this art for a broader terminological context. In that case, steganography is one of the disciplines of the information hiding techniques, as we can see in

the following figure 3.1. Also, we want to be clear that this thesis focuses only on technical steganography.

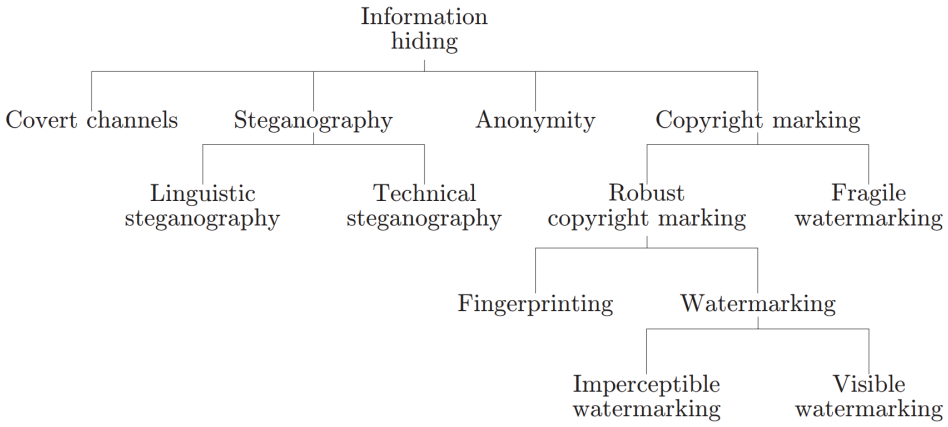


Figure 3.1: Classification of the steganography (the figure was taken from [71]).

3.1.1 Steganography Divided by Cover Work Type

One of the possible divisions of steganography is according to the type used in the cover work. With the development o IT, new possibilities arise:

- **Text file as cover work** - this type is one of the base schemes of steganography that brings several possibilities. Whitespaces can conceal messages in ASCII by appending whitespace to the end of several lines. This approach has a positive effect because spaces and tabulators are generally not visible in text editors, and therefore the message is effectively hidden from observers. Besides, other techniques can hide information, such as word and line shifting coding, syntactic and semantic methods, etc.

  Using steganography that utilizes text files as cover work has many advantages. Information embedded into a forum article or email message will not attract any attention. Also, this kind of email can be stylized as spam email to increase the innocence of transmission due to common occurrence. Symantec provides for the year 2017 annual ISTR [72] where 53% of all email traffic worldwide is spam. Spam messages can be generated by mimic algorithms, an example of steganography propagation. This process can make generated texts look like an average internet article or advertisement to fool spam filters. Also, mimic algorithms emphasized the language's statistical fingerprint to produce the best match. This language fingerprint is based on the frequency analysis of each letter used in a specific language.

- **Audio file as cover work** - it is another scheme of steganography. Simple methods are used for encoding LSB 2.1.1 technique which replaces bit by information in each

sampling point. This technique can be efficiently applied to encode large amounts of hidden data in a given audio signal. It does so at the expense of introducing significant noise at a theoretical upper limit [73]. Another method uses phase coding that utilizes substituting the phase of an initial audio segment with a reference phase, representing the embedded data. The phase of subsequent segments is adjusted in order to preserve the relative phase between segments [74]. Also, this method provides a more complex solution than [73]. Other methods, described by author [74] for steganography, that used an audio file as cover are spread spectrum and echo hiding. In the case of spread spectrum, the method spreads the embedded data across the frequency spectrum as much as possible. In the case of echo hiding, data are embedded into a host signal by introducing an echo. The data are hidden by three echo parameters - decay rate, initial amplitude, and offset. The echo blends as the offset between the original and the echo decreases. At some point, the echo and original sound are not perceived as separate by the human ear.

- **Video file as cover work** - these files are nothing else than many images and sounds representing a video stream. This steganography scheme provides ample space for hiding information, and due to a continuous stream of information, it is difficult for an observer to get any suspicion.

  Imagine MPEG video file format. We have two options for embedding information with this kind of video stream. The first one is the modification of Intra frames (I-Frames). These frames are entirely independent of others, and they are fully transmitted. How they are compressed matches the compression used in JPEG format. I-Frames are key ones. Other types of frames are relative to them. The number of frames between two I-Frames can be referred to as GOP. The number is approximately 14-17. Therefore, embedding information into I-Frames can be possible by some of the techniques mentioned in the chapter 2.1.3. Also, MPEG provides other frames, such as Predicted frames (P-Frames) and Bi-directional frames (B-Frames). B-Frames are similar to P-Frames, but they are related to the previous and next frames in the stream. P-Frame can describe as a backward difference from the previous I-Frame. They are calculated based on the motion vector (see figure 3.2) of the given macroblock. The given P-Frame consists of such motion vectors.

  The second possible option for steganography is a modification of those motion vectors of P-Frames or B-Frames. In the data, the stream of P and B-Frames much more often than I-Frames (see figure 3.3), so the data throughput is more significant or the same as for DCT algorithms modifying I-Frames. The author in [76] shows that the correct change in the quantization ratios of the vectors in the encoder provides approximately two times greater transfer capacities than previously known vector-using methods. However, the

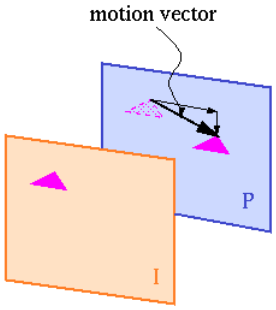


Figure 3.2: Principle of the MPEG motion vector.

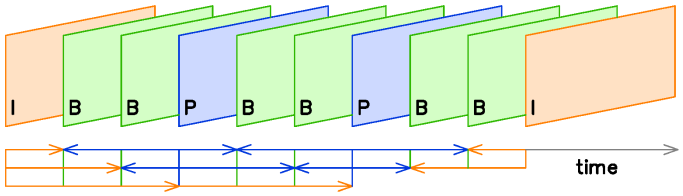


Figure 3.3: The composition of GOP of MPEG file format (the figure was taken from [75]).

method requires the processing of uncompressed data. For many techniques, however, data payload capacity and undetectability depend on the content of the video itself - the motion vectors themselves. For example, authors in [77] show that the changes encoded into the vectors of rapidly moving objects are unrecognizable and almost undetectable. However, this leads to the problem of choosing suitable vectors for embedding data [78]. The best way is to use both frames as the most exciting direction of video stream steganography. However, such a hybrid method has not yet been described.

Many tools for embedding secret messages into video files were developed. However, many do not hide information directly into the video stream, but they use metadata or eventually insert data behind the end of the file flag. These approaches are not real steganography of video files. From the tools, only a few are engaging in using steganography, such as MSU Stegovideo and Steganosaurus. MSU Stegovideo is slightly obsolete, and it is weaknesses known as the methods of detection [79]. Steganosaurus is the newer one that supports insertion into motion vectors of H2.64 format, one of the most modern techniques available today. Unfortunately, this tool contains a single vector selection scheme for insertion, and that is the use of the first motion vector in the image frame. This property makes it virtually non-usable - low cover capacity and easy detection.

- **Image file as cover work** - image files are the most common data file on the Internet

after text information files and for this thesis are the key cover work type. This type of cover work is ideal for hiding information because of the limitation of the human eye. Steganography provides many techniques for different image file formats, such as BMP, GIF, PNG, and mainly JPEG. BMP file format provides a well-suited solution for steganography that we can embed large data without a significant change of the original image information. The most commonly used steganography techniques are described in chapters 2.1.1 and 2.1.2. Usually, the BMP file format is not the best image representation for transmitting over the Internet due to its large size. Therefore, steganography algorithms that use GIF, PNG, and mainly JPEG cover works for embedding information are more interesting. JPEG steganography is one of the most complicated because data are not hidden directly into the spatial domain of the image. Instead, they use frequency domain as we described in chapter 2.1.3. As we mentioned, this thesis described a solution for breaking the steganography algorithms OutGuess2.0 and F5 that uses JPEG compression for embedding the information. Problematics of JPEG compression and mentioned steganography algorithms will be discussed in detail later.

3.1.2 Steganography Divided by Embedding Method

Another possible division of steganography is according to the type of used method for the information insertion. This division is not so frequent. Therefore, we will describe it only briefly:

- **Substitution steganography** - these methods embed the least significant part of cover work to embed information. Therefore, this modification is unrecognizable by human eyes. To this category belongs OutGuess2.0 and F5 algorithms on which we are focusing.

- **Injection steganography** - methods represented by injection steganography usually insert a secret message into a cover work (text, image, etc.) to increase the file size but does not affect a stegogramme presentation by the original application. Therefore, original data are not damaged in a stegogramme.

- **Propagation steganography** - these methods utilize a generation engine which, when fed the payload produces an output file. The content of this file, sometimes referred to as a "mimic", may appear as an image file, audio file, etc. A given payload will yield the same steganography object file with few exceptions when the generation engine reprocesses it. These methods do not use cover work on the input [80].

3.1.3 The main principle of DCT Steganography - JPEG Compression

JPEG compression is a commonly used method for reducing the image file size (see figure 3.4), without reducing the visual quality enough to become noticeable by the human eye. Compression extracts all the information from an image that the human eye is not perceptible.

(a) Quality 80 - 39,4 kB (b) Quality 50 - 33,3 kB (c) Quality 20 - 29,6 kB

Figure 3.4: Example of JPEG compression - Image resolution 250 x 250 px, chroma downsample ratio 4:2:0.

This chapter is dedicated to JPEG compression and the associated DCT used by steganography algorithms such as OutGuess2.0 and F5 for secret message embedding. Generally, DCT is a method that performs the transformation encoding over the sampled signal. The signal is generally a 1D - nD signal. In our methodology, of course, we work with JPEG images that represents a 2D signal. The principle of DCT is to find a correlation between neighboring (even more distant) pixels. The DCT transforms the processed signal from the spatial domain into the frequency domain. Now, we will describe the steps of JPEG compression:

1. **Conversion from RGB space to YC_bC_r space** - the first step of JPEG compression is to convert RGB pixel values of the image into three components YC_bC_r. The Y component represents the luminance (brightness), and the C_b and C_r components represent blue and red chrominance color. The 2D grid determines chrominance components with blue to yellow on one axis and red to green on another axis.

2. **Downsampling of the C_b and C_r chrominance components** - because of the less sensitivity of the human eye to changes in color space, the C_b and C_r chrominance components are downsampled. The downsampling is not used on the Y luminance component because the human eye is more sensitive to changes in brightness.

   By downsampling, it is possible to remove a lot of color information from an image without losing quality. JPEG compression provides several options for downsampling. For example, by taking two adjacent pixels or 4 pixels in the 2x2 grid and averaging

them into one value. As we can see in figures 3.5 and 3.6 that shows each component of the image, most reduced are C_b and C_r chrominance components. It is important to mention that the downsampling is a lossy compression.

(a) Y component (b) C_b component (c) C_r component

Figure 3.5: Decomposition of the image 3.4a to YC_bC_r components - intensity values.

(a) Y component (b) C_b component (c) C_r component

Figure 3.6: Decomposition of the image 3.4c to YC_bC_r components - intensity values.

3. **Discrete Cosine Transform** - the next step is the transformation to the frequency domain by DCT. The DCT is done separately for each YC_bC_r component. Each component plane is divided into macroblocks of size 8x8. If the width or height of the image is not divided by eight, the missing pixels are added by mirroring known pixels to pixels beyond the image boundary. For every macroblock (see figure 3.7a), DCT is performed by equation 3.1 for 2D signal.

$$t(i,j) = c(i,j) \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} s(m,n) cos \frac{\pi(2m+1)i}{2M} cos \frac{\pi(2n+1)j}{2N}, \tag{3.1}$$

where $t(i,j)$ denotes matrix of transformed signal, $c(i,j)$ denotes constant, $s(m,n)$ denotes sampled signal and M and N denotes count of the signal samples - for JPEG macroblock $M, N = 8$. The product of two cosines is also known as the base function.

The result of the DCT is a matrix of DCT coefficients (see figure 3.7b). Therefore, for one macroblock 8x8, the matrix contains a total of 64 DCT coefficients representing the frequencies.

$$\begin{bmatrix} 139 & 144 & 149 & 153 & 155 & 155 & 155 & 155 \\ 144 & 151 & 153 & 156 & 159 & 156 & 156 & 156 \\ 150 & 155 & 160 & 163 & 158 & 156 & 156 & 156 \\ 159 & 161 & 162 & 160 & 160 & 159 & 159 & 159 \\ 159 & 160 & 161 & 162 & 162 & 155 & 155 & 155 \\ 161 & 161 & 161 & 161 & 160 & 157 & 157 & 157 \\ 162 & 162 & 161 & 163 & 162 & 157 & 157 & 157 \\ 162 & 162 & 161 & 161 & 163 & 158 & 158 & 158 \end{bmatrix}$$

(a) Macroblock 8x8 of pixel values

$$\begin{bmatrix} 235.6 & -1.0 & -12.1 & -5.2 & 2.1 & -1.7 & -2.7 & 1.3 \\ -22.6 & -17.5 & -6.2 & -3.2 & -2.9 & -0.1 & 0.4 & -1.2 \\ -10.9 & -9.3 & -11.6 & 1.5 & 0.2 & -0.9 & -0.6 & -0.1 \\ -7.1 & -1.9 & 0.2 & 1.5 & 0.9 & -0.1 & 0.0 & 0.3 \\ -0.6 & -0.8 & 1.5 & 1.6 & -0.1 & -0.7 & 0.6 & 1.3 \\ 1.8 & -0.2 & 1.6 & -0.3 & -0.8 & 1.5 & 1.0 & -1.0 \\ -1.3 & -0.4 & -0.3 & -1.5 & -0.5 & 1.7 & 1.1 & -0.8 \\ -2.6 & 1.6 & -3.8 & -1.8 & 1.9 & 1.2 & -0.6 & -0.4 \end{bmatrix}$$

(b) Matrix of DCT coeficients

Figure 3.7: Matrixes of pixel values and DCT coefficients.

As we can see in figure 3.7b, the largest value is always located in the top-left corner of the matrix. This value represents the DC coefficient. This DC coefficient is the average value of all pixel values on the macroblock. Other higher values are typically situated around the DC coefficient and represent low image frequencies. Note that the coefficients closest to zero are populated around the lower-right corner of the macroblock. These values represent the high frequencies of the image, and also, these coefficients are removed by JPEG compression or used by steganography algorithms for embedding the information. It is essential to mention that every DCT coefficient is also called the AC coefficient, except the DC coefficient.

4. **The quantization of DCT coefficients** - this step is crucial for JPEG compression. The quantization of DCT coefficients causes the loss of information. Each coefficient in the macroblock is divided by the value that is stored in the quantization table (see equation 3.2).

$$N_{i,j} = round\left(\frac{T_{i,j}}{Q_{i,j}}\right), \tag{3.2}$$

where N is the matrix that contains normalized DCT coefficients, T is the matrix that contains the DCT coefficient from the previous step, and Q denotes matrix represent quantization table.

$$\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

(a) Quantization table

$$\begin{bmatrix} 15 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(b) Normalized DCT coefficients

Figure 3.8: Matrixes of quantization table and normalized DCT coefficients.

The quantization tables (see figure 3.8a) are designed to have low values in the position of the DC coefficient, and the areas belong to low-frequency. In contradiction, in high-frequency areas, the quantization tables have high values. Therefore, the result (according to the equation 3.2) of the matrix of normalized DCT coefficients (see figure 3.8b) contains a large number of zeros in the high-frequency areas. This behavior leads to the Huffman coding when long sets of zeros are very well coded.

Many graphics editors have saved only a few quantization tables, for example, only three tables for quality 90, 70, and 50. Suppose the user wants to compress an image with a quality factor of 80. In that case, the editor applies the two closest quantization tables (90 and 70) to determine a new table with a quality factor of 80. Also, the quantization tables are implemented in many libraries. Another option is that the user can design his quantization table, which can focus on compression, for example, of the horizontal, vertical, or diagonal edges.

5. **Zig-Zag ordering** - this step is for the increasing number of zeros placed next to each other. This ordering reorders the values using a zig-zag type motion to group similar frequencies. Following figure 3.9 describe this motion.

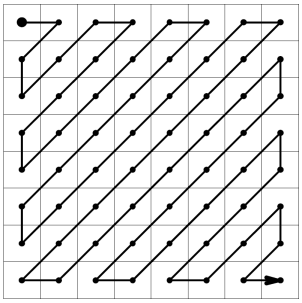


Figure 3.9: Zig-Zag ordering for JPEG compression (the figure was taken from [81]).

6. **Huffman coding** - this step is beyond the scope of this thesis. Briefly, first, RLE (massive sets of zeros) compress the high-frequency coefficients, and DPCM compresses the first low-frequency coefficient. Next, the Huffman algorithm compressed everything and stored it in the JPEG header.

   However, it is essential to mention one research by author [82]. The author introduces a new approach to detect steganography content. The basis is Huffman coding. The research describes that a stegogramme will have a specific structure of Huffman coding than an image without an embedded message. The author also applies a neural network that identifies stegogramme. The detection rate is very high, in some cases up to 99.9%. However, there is a question about whether this methodology identifies the presence of a secret message or only the processing of a specific JPEG encoder for which the neural network has been learned. This property could lead to a false classification of a stegogramme and overloading of a deployed steganalysis detector. However, it is still fascinating research.

   Also, we adapt this methodology in our research [83]. The main idea is to detect which algorithms (Outguess2.0 and StegHide) were used. When the steganography method is applied to JPEG images, the DCT coefficients are modified, and the results of Huffman coding are different. These structural changes can be seen in overall statistics as changes in Huffman coding in the histogram length. These changes contain some hidden patterns found in every steganography file and are unique to each steganography tool. For classification, we also use the neural network. The neural network utilizes topology with 68 input neurons, two hidden layers of 128 neurons, and two output neurons – each for one steganography algorithm with a range of $0 - 1$, where one signs the highest probability for the given algorithm. The input vector is made of 64 Huffman tables coefficients, two quality values, and two values for the image's resolution.

3.1.4 Know Your Enemy - OutGuess2.0

This chapter focuses on the steganography algorithm OutGuess2.0 that we primarily focus on detecting in this thesis.

Algorithm 1 Pseudocode of the encoding procces of the OutGuess2.0.

```
convert image covIMG to DCT domain dom in 8x8 macroblocks
generate randomised sequence C using data dom and seed sd
for j = 1 to length(mess) do
    cVal ← DCT(C_j)
    while cVal = DC or cVal = 0 or cVal = 1 do        ▷ Skip DC, 0 and 1 coefficients
        cVal := next DCT coefficient from C
    end while
    cVal_j ← C_j mod 2 + mess_j
    C_j ← cVal_j
end for
correct coefficients to the looks of the clear image
generate original sequence covIMG using data C and seed sd
convert each 8x8 macroblock back to spatial domain
```

The encoding process of the OutGuess2.0 (see pseudocode 1) is a combination of randomized Hide & Seek [2] and JSteg [31] algorithm. The first step is converting the input cover work image $covIMG$ to the DCT domain. Then the DCT coefficients are shuffled into a random order by a PRNG. Then the secret message $mess$ is embedded by the same technique as JSteg. It is important to mention that the encoding process avoids embedding within DC and any AC coefficient $cVal$ equal to either 0 or 1. Next, the DCT coefficients are shuffled back to the correct position. At this phase, OutGuess2.0 provides corrections to the coefficients in that they appear similar to that of a clear image in terms of frequencies of the values. This correction causes a new compression, and the stegogramme is not changed from the point of view of statistical analysis as was described in [84]. Finally, the image is converted back to the spatial domain, producing the stegogramme.

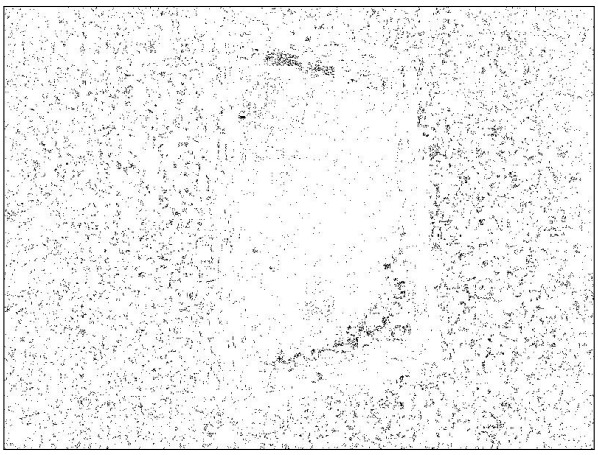
Algorithm 2 Pseudocode of the decoding procces of the OutGuess2.0.

```
convert image stegoIMG to DCT domain dom in 8x8 macroblocks
generate randomised sequence S using data dom and seed sd
for j = 1 to length(stegoIMG) do
    cVal ← DCT(S_j)
    while cVal = DC or cVal = 0 or cVal = 1 do        ▷ Skip DC, 0 and 1 coefficients
        cVal := next DCT coefficient from stegoIMG
    end while
    mess_j ← stegoIMG_j mod 2
end for
```

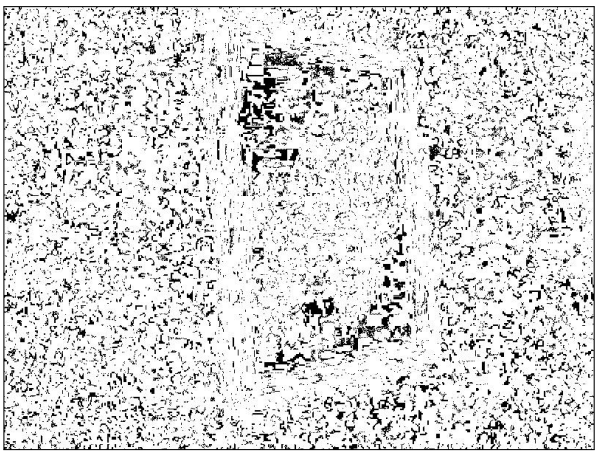
The decoding process (see pseudocode 2) converts the stegogramme $stegoIMG$ back to the DCT domain before being shuffled using the same seed sd that was used in the encoding process. The secret message data $mess$ are extracted by LSB technique from all DCT coefficients, except DC and all AC coefficients equal to 0 or 1.

At this moment, a question arises - Where exactly does OutGuess2.0 embed the secret message data? In other words, which of these three components YC_bC_r are modified? We had thoughts, for example, that the embedding process excludes the Y component (luminance) because human vision is more sensitive to a luminance change, and therefore modification could be visible. On the other hand, DCT coefficients of the Y component have mostly bigger values than the coefficients of the other two components. Therefore, these values provide a better solution for the LSB technique.

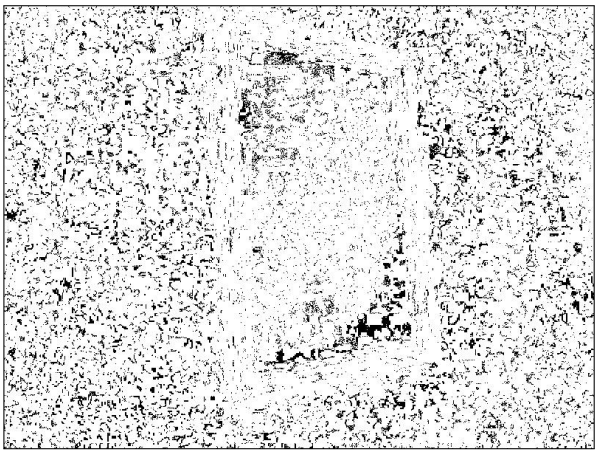
At our previous researches [85, 86, 87, 88], we was not distinguished YC_bC_r as independent separate values. Therefore, this leads to the distortion of the correct stegogramme and clear image classification. This thesis provides new research where we approach these components independently during the entire classification process. As we can see in the following figure 3.10, we used pixel difference between all three components of the cover work and his stegogramme. Of course, as we mentioned before, OutGuess2.0 uses its compression that distorts the results of the pixel difference. Therefore, we can not confidently say that the OutGuess2.0 always modified all three components in the same way. Nevertheless, by using the individual approach to YC_bC_r, we are increasing the likelihood of the correct classification. In other words, we do not need to care about which component is modified.



(a) Y (luminance) component



(b) C_b (blue chroma) component



(c) C_r (red chroma) component

Figure 3.10: Pixel difference between cover work and his stegogramme, where white color indicates non-equal pixel value and black color indicates equal pixel value.

3.1.5 Know Your Enemy - F5

This chapter focuses on the steganography algorithm F5. It is essential to mention that we want to test our methodology primarily intended for OutGuess2.0 on another DCT steganography algorithm. Therefore, this algorithm will be described briefly.

As mentioned before, the F5 algorithm also had previous versions. In essence, each version removed fatal design errors used for detection by some steganalysis methods. The main improvements in the latest version - F5 minimized the disturbances on a cover work. These disturbances were caused during the data embedding. Therefore, matrix encoding has been introduced to reduce the number of changes needed to embed data.

Algorithm 3 Pseudocode of the encoding procces of the F5.

```
1: convert image covIMG to DCT domain dom in 8x8 macroblocks
2: for j = 1 to length(mess) do
3:     cVal ← dom_j
4:     while cVal = DC or cVal = 0 do
5:         cVal := next DCT coefficient from dom
6:     end while
7:     CVal ← abs(cVal_j)
8:     if CVal = mess_j and CVal > 0 then
9:         CVal ← CVal + 1
10:        abs(dom_j) ← CVal
11:    else if CVal ≠ mess_j and CVal < 0 then
12:        CVal ← CVal − 1
13:        abs(dom_j) ← CVal
14:    end if
15:    if dom_j = 0 then
16:        mess_j := next mess_j
17:    end if
18:    C_j ← cVal_j
19: end for
20: convert each 8x8 macroblock back to spatial domain
```

The main principle of the encoding algorithm (see pseudocode 3) is the evaluation of negative DCT coefficients. This principle is also an improvement against the old version F3: odd-negative DCT coefficients are mapped by steganography value equal to 0, even-negative coefficients by value 1, odd-positive coefficients by value 1, and even-positive coefficients by a value equal to 0. For example, the DCT coefficient with value -3 will remain -3. Therefore, the histogram for the stegogramme will not show deviations in the frequency distribution. The encoding algorithm itself works by calculating the embedding potential of encoding $mess$ within $covIMG$ based on $mess_j$. Hamming coding is applied to embed potentially more than one bit per value by making no more than one modification to the DCT coefficients.

Algorithm 4 Pseudocode of the decoding procces of the F5.

```
1: convert image stegoIMG to DCT domain dom in 8x8 macroblocks
2: for j = 1 to length(mess) do
3:     cVal ← dom_j
4:     while cVal = DC or cVal = 0 do
5:         cVal := next DCT coefficient from dom
6:     end while
7:     CVal ← abs(cVal_j)
8:     if CVal = mess_j and CVal > 0 then
9:         mess_j ← abs(cVal_j) − 1
10:    else if CVal ≠ mess_j and CVal < 0 then
11:        mess_j ← abs(cVal_j) + 1
12:    end if
13: end for
```

On the decoding process (see pseudocode 4) we firstly convert stegogramme $stegoIMG$ to obtain DCT coefficients. Again DC values and any coefficient equal to zero are skipped. The most important part of the decoding is addition and subtraction from $CVal$ caused by the inverted encoding process because we need to extract the correct bit for message $mess_j$ - for example, the value is decremented (line 9) because this value was incremented in the encoding algorithm.

3.2 Targeted Steganalysis

Targeted steganalysis is one of two types of steganalysis. As we already mentioned in the introduction, we have both a suspicious picture and mostly an original cover work in the targeted steganalysis. This approach is also called known-cover attacks steganalysis. Also, there is a particular case called known-stego attacks steganalysis. In this case, we do not know the original cover work, but we know used steganography algorithm. We evoke that the main idea of steganalysis is only to identify the presence of a secret message, not it is successful extraction. This chapter deals with three types of attacks on steganography algorithms.

3.2.1 Visual Attack

The visual attack is the first and primary type of targeted steganalysis. Even though it is a type of approach that relies on visual examination of the suspected image, it is clear that the image cannot be compared with the original only by human vision due to the main principle of steganography. Therefore, we need to remove those parts of the digital image under which a secret message is hidden. These parts are most likely not modified by the steganography algorithm and only cover the embedded message. Thus, the key to the successful classification

of the stegogramme will be if we correctly identify those redundant data that can be ignored and those data potentially containing the secret message.

The visual attack is often used to classify Hide & Seek algorithm. This algorithm uses the LSB technique. The main idea for steganalyst will be to reduce digital image into single bit plane. To be more specific, it is the LSB bit plane in the case of the LSB technique. The next step will be an investigation by the human eye for any periodic or other types of suspicious patterns in the image.

Even though it is a primary type of steganalysis, it has several defects. For example, different stegogramme classification conditions are due to the heterogeneous sensitivity of human vision among observers. Also, it is not suited for extensive image data.

3.2.2 Structural Attack

The basic principle of a structured attack is that it detects high-level properties, which are prominent features of a particular steganography algorithm. In other words, steganography algorithm usually leaves behind a characteristic fingerprint structure on data. Therefore, steganalyst can immediately flag the image as suspicious. The next step is to find the before-mentioned fingerprints and classify a suspected image as stegogramme.

3.2.3 Statistical Attack

Thanks to mathematical statistics, we can determine if some event happens on a set of random data. Thus, we can apply this definition to the issue of steganalysis. Statistical tests can reveal an image that has been modified by steganography algorithm by the statistical property that deviates from a norm. For example, some of the older versions of OutGuess show apparent deviations in the histogram of DCT values.

One of the bases statistical attacks is Chi-squared Test. This test makes it possible to compare a suspected image's statistical properties with its counterpart's theoretically expected statistical properties. Then is possible to classify a suspected image as a stegogramme or clear image.

3.3 Blind Steganalysis

Blind steganalysis is the second type of steganalysis. This type includes every case when we do not know the original cover work or the steganography algorithm used. Therefore, this reflects most of the realistic scenarios, such as scenario two on figure 1.2. With the increasingly developed steganography algorithms, these classification methods are fundamental. Same as our method described in this thesis.

Chapter 4

Contribution to the Area

As we mentioned in the chapter 1, steganography can be used for illegal activities. Therefore, steganalysis techniques are fundamental to detecting such stegogrammes. In chapter 2 we present many solutions that have been proposed in recent years, but there are still some needs for further research. We cannot concentrate on breaking all steganography algorithms and related issues in our effort. Instead, we contributed to detecting the stegogrammes created by steganography techniques that are based on the modification of the transform domain of JPEG images.

This chapter would like to summarize our contribution to the area. Many research, for example, [59, 89, 90], were tested and developed on the low-resolution grayscale JPEG images. This type of data is sufficient for the research and testing. However, we want to test our methodology on data that reflect realistic scenarios - high-resolution colorful images. For example, proposing methods [89, 90] were tested on one low-resolution image, and detection capability was from 66% to 94% according to the selected steganography algorithm and secret message length. Also, the authors in [91] propose a new methodology based on the universality to detect several steganography algorithms, but for the cost of detection success rate.

Now we will summarize all of our research assets with the concrete contribution to the area of blind steganalysis. This list is also separated into two sections - primary and secondary contributions. Primary contributions summarize the most critical assets with regards to modern requirements of steganalysis algorithm, such as high detection success rate, etc. We have only one secondary contribution: the possibility of applying our research on another steganography algorithm - F5.

PRIMARY CONTRIBUTION

- Detecting stegogrammes created by OutGuess2.0 steganography algorithm - our methodology can correctly classify the stegogramme created by one of the most available and advanced steganography algorithms - OutGuess2.0. It is also an algorithm available on

the Internet that is free to download and provides the user his own intuitive GUI - these facts increase the likelihood that it will be in the corporate sector to send an image with the embedded secret message. This algorithm has been subjected to an analysis in the chapter 3.1.4.

- High classification success rate - concerning the sensitivity of the test. In other words, evaluation is stricter in classification between a stegogramme and a clear image. Results are discussed in detail in chapter 6.1.

- Invariance of the test's sensitivity against secret message length - this is critical. The secret message can contain anything from short passwords to the long description of the customers. Therefore, it is essential to correctly classify stegogrammes with the same success rate with no relation to a message payload. Our methodology provides this invariance of the length of the embedded message on the test's sensitivity. This statement is also discussed and tested by ANOVA statistical hypothesis test in chapter 6.2.

- Image resolution and color depth - development and testing on several resolutions, including high-resolution colorful JPEG images (up to 10 Mpx). This feature that we bring to our methodology reflects realistic data that can be tested, for example, on the company's internal network. High-resolution and color depth provide better solutions for DCT steganography, and also, these kinds of images are commonly used, so they are not suspicious.

- Application of the ANN (we are using the specific type called shallow neural network) - our methodology implements the ANN to classify stegogramme and clear image. With the ANN, we can replace the trivial classification process, such as comparing two values with the more complex solution, including classification based on several characteristics and the experience of the ANN. We are also extending the issues of ANN. The application of ANN to our methodology is discussed in chapter 5.3.

- JPEG macroblock filtering - filtering of non-modified macroblocks by embedding process. By observing OutGuess2.0, we bring a new feature to the classification process. If we remove macroblocks that are not used to embed the secret message, we can remove the distortion from blockiness calculation and, therefore, increase the classification success rate. More is discussed in chapter 5.1.

- Publications of our research - all the contributions presented here were published in international conference, or journal [92].

SECONDARY CONTRIBUTION

- Application of the research for the classification of the F5 steganography algorithm - as we have already mentioned, we test (see chapter 6) the extensibility of our methodology on another DCT steganography algorithm. This extensibility will generally increase the applicability of our research.

Chapter 5

Methodology

This chapter will describe our methodology, including individual steps in detail. Results of our research are mentioned later on. The following figure 5.1 represents the process of the classification.

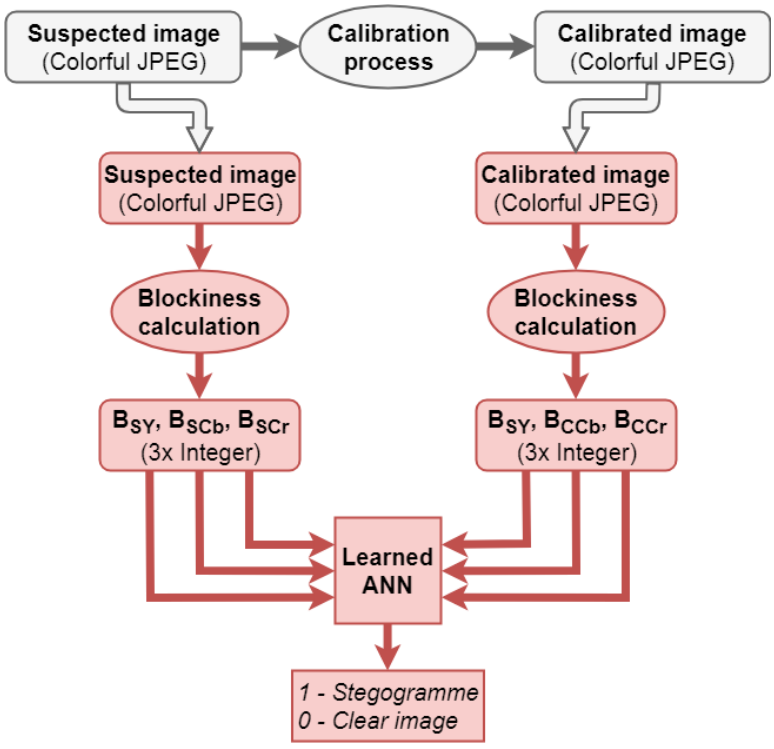


Figure 5.1: Classification process - diagram

5.1 Calibration Process

As we mentioned in chapter 2.2.2, we adopted the method of steganalysis [59], specifically "blind steganalysis". Thus, there is no cover image that we can compare with the suspected image and detect differences. This disadvantage is removed by using a method called calibration.

The input of this method is a suspected image, and the output is a calibrated image. The calibrated image is not a copy of the cover image, but it is very similar. This calibrated image should represent the state of the image after embedding the secret message by OutGuess2.0. As we mentioned in chapter 3.1.4, the OutGuess2.0 use it is own compression with quality factor Q_S. In other words, the stegogramme passed double compression, and it is necessary to compress the suspected image by factor Q_C (quality factor of an image before OutGuess2.0 compression) and consequently by factor Q_S. This process will simulate embedding messages by using the OutGuess2.0 system. The algorithm for creating the calibrated image consists of the following steps:

1. Cropping the suspected image by 4px from every side.
2. Compress the suspected image by using Q_C.
3. Compress the suspected image by using Q_S.

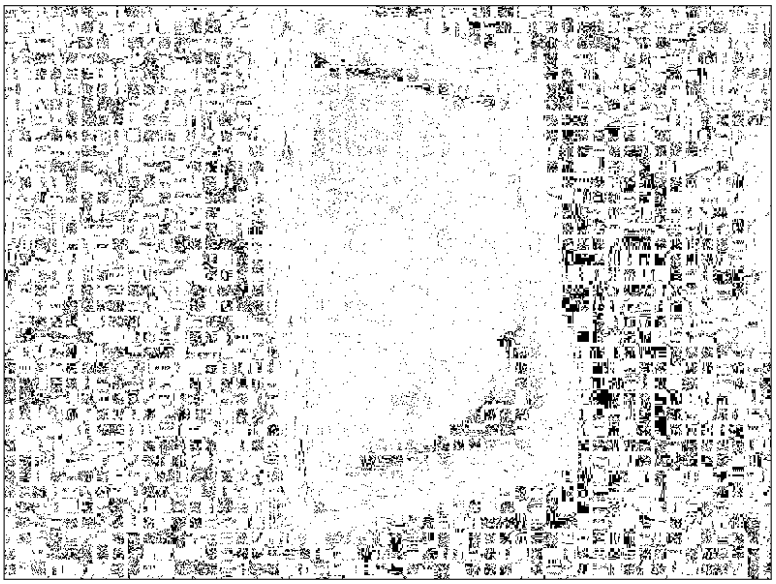
Cropping the suspected image will have the important effect that we will describe in detail later in chapter 5.2. The question is how to get quality factor Q_C when the state of the image before OutGuess2.0 compression is unknown. The advantage is that the OutGuess2.0 preserves the histogram. For this purpose the authors [59] developed the following formula 5.1:

$$Q_C = \arg\min_{Q} \sum_{(i,j)} \sum_{d} |h_d(i,j) - h_d(i,j,Q)|^2, \tag{5.1}$$

where $h_d(i,j)$ is a histogram of values (i,j) -th DCT mode of the suspected image, $h_d(i,j,Q)$ is the same histogram of the cropped image with using of quality factor Q, that is subsequently compressed by quality factor Q_S. So Q_C is calculated as the quality factor that minimizes the difference between $h_d(i,j,Q)$ and $h_d(i,j)$ for those DCT modes (i,j) that correspond to the lowest-frequencies. Those are coefficients with coordinates (1, 2), (2, 1), and (2, 2) in the DCT table. The DC term with coordinate (1, 1) is excluded. Also, it is important to mention that this is done for each individual YC_bC_r components, then these values are sum together. Next, Q_C is calculated as was mentioned above.

For example, from the following figure 5.2, we can see how the calibration process works for stegogramme (with 1000 B embedded message) and the clear image. For the stegogramme,

we made pixel difference between stegogramme and his calibrated image - figure 5.2a. Then equally for the clear image - figure 5.2b. Of course, the comparison of each pixel between the input image and the calibrated image cannot be the same, but we can see the difference between the stegogramme and clear image.



(a) Pixel difference - stegogramme and his calibrated image

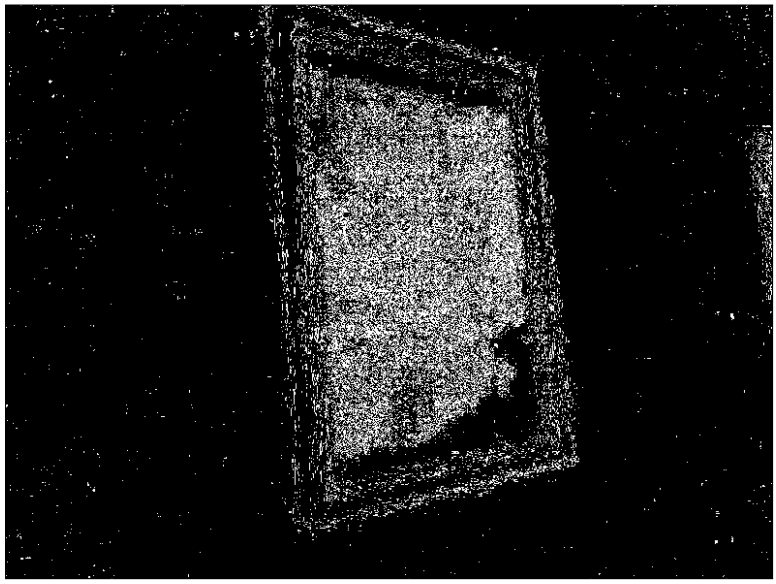
(b) Pixel difference - clear image and his calibrated image

Figure 5.2: Difference between stegogramme and clear image, where white color indicates non-equal pixel value and black color indicates equal pixel value.

From the initial testing and observing the steganography algorithm OutGuess2.0's behavior, we improved the calibration process for colorful JPEG images. How is it done? Please

follow the next paragraphs.

In the following figure 5.3a, we can see the high-frequency domains (high details, white pixels) are used for embedding the secret message. This figure was produced by pixel difference between the stegogramme in figure 5.3b and original cover work.



(a) Pixel difference - stegogramme and cover work

(b) Stegogramme

Figure 5.3: Embedding to the high-frequency domains of the stegogramme

Also, some tolerance was added to neglect the effect of OutGuess2.0 compression. Therefore, we can see where the secret message is embedded. If we know that the OutGuess2.0

is focused on modifying the high-frequency domains, we use this fact to improve detection capability. For the blockiness calculation, we exclude (based on the standard deviation of every macroblock) the low-frequency domains of the suspected image because this area distorts the result. From the practical point of view, the classification process should not mark some of the stegogrammes as a clear image. Therefore, this enhancement has a positive effect on the behavior of the detector in the way of the correct classification of the suspect image - it improves the sensitivity of the test mentioned in chapter 4. Some results of the macroblock filtering are listed in the chapter 6.3.

5.2 Blockiness Calculation

We also find a property that detects the presence of a hidden message in the image - it is called Blockiness. This property responds to the variable lengths of the messages due to the invariance of the classification success rate against the secret message length, as we mentioned in chapter 4.

As we mentioned in chapter 2.2.2, authors [60] present blockiness values. It is a statistical property that is different for the calibrated and the suspected image, so we can determine whether it is stegogramme. A blockiness value is defined as the sum of spatial discontinuities along the boundary of all 8x8 macroblocks of JPEG image. In other words, stegogrammes will contain different coefficients along the borders of 8x8 macroblocks than the clear image. Again, the input of this calculation is an image (suspected and calibrated image), and the output is the blockiness value for each individual $\mathrm{YC_bC_r}$ component. The formula 5.2 for blockiness calculation is as follows:

$$B_X = \sum_{i=1}^{[\frac{M-1}{8}]} \sum_{j=1}^{N} |p_{8i,j} - p_{8i+1,j}| + \sum_{j=1}^{[\frac{N-1}{8}]} \sum_{i=1}^{M} |p_{i,8j} - p_{i,8j+1}|, \tag{5.2}$$

where X indicates one of $\mathrm{YC_bC_r}$ components, p_i, j denotes coordinates of the pixel values on the $M \times N$ image.

As we can see from the formula 5.2, the calculation operates with individual columns and rows. First, the sum of all differences between the values of the eighth and ninth row (generally the row below) is calculated. The sum is continuously accumulated for the others pair of the rows (generally the rows 8i and 8i+1). Similarly, the second sum for the columns is solved, which accumulates the pair of columns (generally the columns 8j and 8j+1). Finally, the two sums are added together, and we get the blockiness value of one $\mathrm{YC_bC_r}$ components.

As we mentioned in chapter 5.1, one step of the process is cropping the suspected image from every side by 4 pixels. This modification ensures that the entire block structure is removed, 8x8 macroblocks are shifted from both directions, and thus a more accurate estima-

tion of the calibrated image is derived. Blockiness calculation takes advantage of the fact that JPEG steganography algorithms embed the secret message in the same 8x8 macroblocks used for compression. Now, we can calculate three blockiness values B_{SX} for the suspected image and three blockiness values B_{CX} for the calibrated image, where X denotes one of YC_bC_r components. We can get YC_bC_r values of the pixel by calculation via RGB pixel values.

5.3 Artificial Neural Network

Neural networks are inspired by biological neural networks. Due to this fact, we can simulate a simple function of the human mind. Like the human model, the artificial neural network (ANN) needs to gain experiences by learning. According to these experiences, it will then decide – it is a set of data (inputs), for which we know the correct result (output).

In our neural network, we used the Perceptron. It is one of the most used models, whose potential P is defined as the weighted sum of the incoming signals (inputs). If the threshold is exceeded, it leads to excitation of the neuron (to 1). Otherwise, it leads to inhibition (to 0). In order to properly recognize the input values of the ANN, every neuron's weight must be correctly set. This process is done by learning from the training set. Our ANN consists of seven neurons – 6 inputs and one output neuron - figure 5.4.

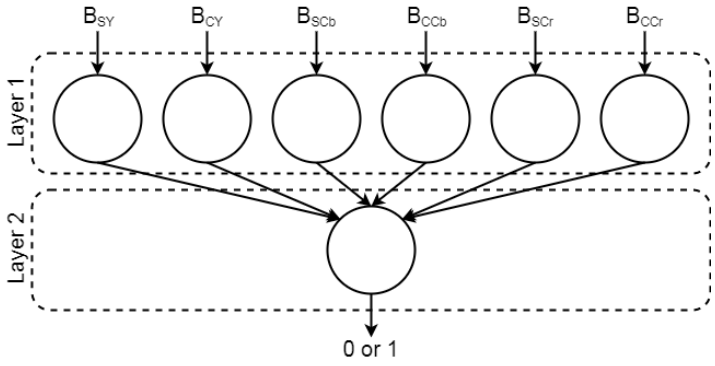


Figure 5.4: Topology of the Artificial Neural Network

To the input, we sent six inputs representing six blockiness values of YC_bC_r for the suspected and the calibrated image. The potential P of the neuron is defined by the equation 5.3. Therefore, the border between the classification of the stegogramme and the clear image will be defined as the poly-plane in 6D space. The output then generates a value of 1 or 0. Value 1 indicates stegogramme, and value 0 indicates a clear image without any secret message. We must emphasize that we use our simple ANN for finding the best possible border between the classification of the stegogramme and the clear image.

$$P = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + w_5x_5 + w_6x_6 \tag{5.3}$$

where w indicates weights of the neurons adjusted by training, x_1 indicates blockiness value B_{SY} of the Y component of the suspected image, x_2 indicates blockiness value B_{CY} of the Y component of the calibrated image, x_3 indicates blockiness value B_{SCb} of the C_b component of the suspected image, x_4 indicates blockiness value B_{CCb} of the C_b component of the calibrated image, x_5 indicates blockiness value B_{SCr} of the C_r component of the suspected image and x_6 indicates blockiness value B_{CCr} of the C_r component of the calibrated image.

It is hard to imagine poly-plane in 6D space represented by our ANN topology. Therefore, we can simplify this topology to only two input neurons and one output neuron. The potential equation of such topology will be modified to (see equation 5.4):

$$P = w_0 + w_1 x_1 + w_2 x_2 \tag{5.4}$$

where w indicates weights of the neurons, x_1 and x_2 some general input values.

Furthermore, this equation 5.4 is nothing else than an equation for the line in 2D space. Therefore, we can analogously modify this formula to equation 5.5:

$$ax + by + c = 0 \tag{5.5}$$

Then, the border for the classification between two general cases (for us, stegogramme vs. clear image) only represents the line in 2D space.

Also, we try to test other topologies. For example, we include some hidden layers with different numbers of neurons or change neuron output function. However, because we do not have any empirical rule about ANN topologies and no thoughts about it, we left this idea behind. However, for this topology (see figure 5.4), we still get the best results.

In one of our previous research [88], we also use the ANN for improvement of the classification capability. This improvement of the classification capability was also tested and verified.

Chapter 6

Results

In this chapter, we will summarize all results. Also, we mentioned our testing and training database.

We create a database of color JPEG images taken with two devices - Huawei P7 and LG-D605 cameras for development and testing purposes. The database consists of two parts – the test and the train section. As the name suggests, the test section is designated for testing, and the train section is prepared for learning the ANN. Each section contains clear images and stegogrammes created by steganography algorithms OutGuess2.0 and F5. Individual images are available in five different resolutions – 800 x 449, 1024 x 575, 1440 x 809, 2560 x 1438 and 4200 x 2358. For every resolution, we prepared 320 images with a message length of 10 B, 320 images with a message length of 50 B, the same amount of the images for message lengths 200 B, 500 B, 800 B, and 1000 B. Count of images in the database is about 20 500.

For testing purposes, we prepared several testing sets (one for each resolution with one message length). These sets contained 640 vectors – 320 stegogrammes and 320 clear images. These vectors consist of seven values representing blockiness value B_{SY} of the Y component of the suspected image, B_{CY} of the Y component of the calibrated image, B_{SCb} of the C_b component of the suspected image, B_{CCb} of the C_b component of the calibrated image, B_{SCr} of the C_r component of the suspected image and B_{CCr} of the C_r component of the calibrated image. The seventh vector value is 1 or 0 - 0 indicates clear image, and 1 indicates stegogramme.

To learn the ANN, we prepared the set of 2552 vectors consisting of clear images and stegogrammes. As the stegogrammes, we used images with a message length of 10 B for every five resolutions.

6.1 Classification Success Rate

In the figure 6.1, we can see the chart of classification success rates of our steganography detector compared to our previous results. We can confidently say that we improve the classification

success rate for the low-resolution images, but for the cost of reducing the high-resolution images success rate. If this methodology were used to monitor the corporate network, images with a lower resolution would likely be used than high-resolution images that are unsuitable due to their size for network transmission.

The classification success rate is computed as the accuracy rate by the following equation 6.1:

$$Acc = \left(\frac{a+d}{a+b+c+d} \right) * 100, \tag{6.1}$$

where a is denotes as TP - true positive result, d as TN - true negative result, b as FN - false positive result and c as FP - false negative result. We can also compute the error rate Err by subtracting Acc from 100.

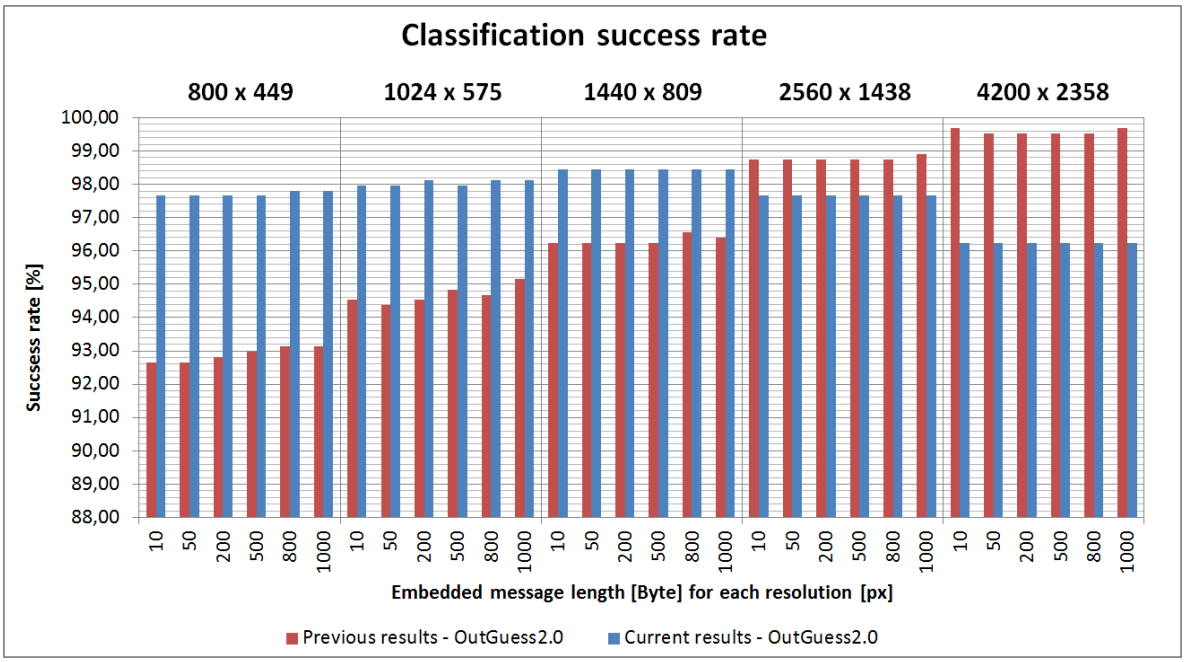


Figure 6.1: Chart of the classification success rate with the comparison of the current results and results of [83, 85].

Next, we present specific results for each resolution and the secret message length. Please note that only the results for the secret message length equal to 10 B and 1000 B are shown. The rest of the results are attached in appendix A at the end of this thesis. Results tables also include statistical values for the sensitivity and specificity of the test. Especially the high sensitivity of the test is essential for us because of the reasons described in the chapter 4. The sensitivity is defined as the probability that a test result will be positive when the stegogramme is present. Sensitivity is calculated by the following equation 6.2:

$$Sensitivity = \left(\frac{TP}{TP + FN}\right) * 100, \tag{6.2}$$

where TP is true positive, and FN is false negative.

And the specificity of the test is defined as the probability that a test result will be negative when the stegogramme is not present. The following equation calculates specificity 6.3:

$$Specificity = \left(\frac{TN}{FP + TN}\right) * 100. \tag{6.3}$$

where TN is true negative, and FP is false positive.

Following tables of test[1] results:

Table 6.1: Test results - OutGuess2.0, resolution 800 x 449, 10 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 318	FP = 13
Test outcome NEGATIVE (clear image)	FN = 2	TN = 307
Sensitivity [%]	**99,38**	
Specificity [%]	**95,94**	
Accuracy [%]	**97,66**	
Error [%]	**2,34**	

Table 6.2: Test results - OutGuess2.0, resolution 800 x 449, 1000 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 319	FP = 13
Test outcome NEGATIVE (clear image)	FN = 1	TN = 307
Sensitivity [%]	**99,69**	
Specificity [%]	**95,94**	
Accuracy [%]	**97,81**	
Error [%]	**2,19**	

[1]All testing was performed on Intel Xeon Gold 6138 32 cores 2.00GHz, RAM 100GB, parallelized.

Table 6.3: Test results - OutGuess2.0, resolution 1024 x 575, 10 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 318	FP = 11
Test outcome NEGATIVE (clear image)	FN = 2	TN = 309
Sensitivity [%]	**99,38**	
Specificity [%]	**96,56**	
Accuracy [%]	**97,97**	
Error [%]	**2,03**	

Table 6.4: Test results - OutGuess2.0, resolution 1024 x 575, 1000 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 319	FP = 11
Test outcome NEGATIVE (clear image)	FN = 1	TN = 309
Sensitivity [%]	**99,69**	
Specificity [%]	**96,56**	
Accuracy [%]	**98,13**	
Error [%]	**1,88**	

Table 6.5: Test results - OutGuess2.0, resolution 1440 x 809, 10 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 318	FP = 8
Test outcome NEGATIVE (clear image)	FN = 2	TN = 312
Sensitivity [%]	**99,39**	
Specificity [%]	**97,50**	
Accuracy [%]	**98,44**	
Error [%]	**1,56**	

Table 6.6: Test results - OutGuess2.0, resolution 1440 x 809, 1000 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 318	FP = 8
Test outcome NEGATIVE (clear image)	FN = 2	TN = 312
Sensitivity [%]	**99,39**	
Specificity [%]	**97,50**	
Accuracy [%]	**98,44**	
Error [%]	**1,56**	

Table 6.7: Test results - OutGuess2.0, resolution 2560 x 1438, 10 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 315	FP = 10
Test outcome NEGATIVE (clear image)	FN = 5	TN = 310
Sensitivity [%]	**98,44**	
Specificity [%]	**96,88**	
Accuracy [%]	**97,66**	
Error [%]	**2,34**	

Table 6.8: Test results - OutGuess2.0, resolution 2560 x 1438, 1000 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 315	FP = 10
Test outcome NEGATIVE (clear image)	FN = 5	TN = 310
Sensitivity [%]	**98,44**	
Specificity [%]	**96,88**	
Accuracy [%]	**97,66**	
Error [%]	**2,34**	

Table 6.9: Test results - OutGuess2.0, resolution 4200 x 2358, 10 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 311	FP = 15
Test outcome NEGATIVE (clear image)	FN = 9	TN = 305
Sensitivity [%]	**97,19**	
Specificity [%]	**95,31**	
Accuracy [%]	**96,25**	
Error [%]	**3,75**	

Table 6.10: Test results - OutGuess2.0, resolution 4200 x 2358, 1000 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 311	FP = 15
Test outcome NEGATIVE (clear image)	FN = 9	TN = 305
Sensitivity [%]	**97,19**	
Specificity [%]	**95,31**	
Accuracy [%]	**96,25**	
Error [%]	**3,75**	

It is clear from the previous tables 6.1 - 6.10 that for the correct classification of the stegogramme created by the OutGuess2.0 algorithm and the clear image, we achieved an accuracy of **96,25% - 98,44%** (with an error of 3,75% - 1,56%). However, the sensitivity of the test that was emphasized was **97,19% - 99,69%**.

Possible loss of classification capabilities for high-resolution images may be due to the character of the ANN training set. The training set was learned only at resolutions up to 2560 x 1438. Therefore, the lack of "ANN experience" could lead to this drop in classification capabilities. The calculation of vectors representing 4200 x 2358 images was not included in the training set because of the time-consuming calculation on different and lower PC specs. Another possible reason is modifying the maximum length of a secret message. The OutGuess2.0 algorithm provides this feature. It allows the user to insert a message to a specific length only. In other words, OutGuess2.0, based on the input cover image, allows the user to insert the message with concrete length so that it is not so easily detected in the image. This feature was most prominent in high-resolution images (e.g., 4200 x 2358), especially for 1000 B message length. The number of such "reduced" stegogrammes was in the range from about ten images (for small resolutions) to half of the stegogramme series

(for high resolution). This feature can confuse the learned ANN that determines whether it is a clear image or a stegogramme based on individual blockiness values. This feature also demonstrates how interesting the OutGuess2.0 algorithm is. However, as we mentioned above, low-resolution images are likely to be used in the corporate network.

As we mentioned in the chapter 4, our secondary contribution is the applicability of our methodology to attack another steganography algorithm - the F5 algorithm. Let it be clear that this methodology is primarily focused on the classification of the stegogrammes created by the OutGuess2.0 algorithm. Therefore, we only consider the possible application to another algorithm as an additional feature. Again, we only present the secret message length equal to 10 B and 1000 B results. The rest of the results are attached in appendix A:

Table 6.11: Test results - F5, resolution 800 x 449, 10 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 299	FP = 13
Test outcome NEGATIVE (clear image)	FN = 21	TN = 307
Sensitivity [%]	**93,44**	
Specificity [%]	**95,94**	
Accuracy [%]	**94,69**	
Error [%]	**5,31**	

Table 6.12: Test results - F5, resolution 800 x 449, 1000 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 307	FP = 13
Test outcome NEGATIVE (clear image)	FN = 13	TN = 307
Sensitivity [%]	**95,94**	
Specificity [%]	**95,94**	
Accuracy [%]	**95,94**	
Error [%]	**4,06**	

Table 6.13: Test results - F5, resolution 1024 x 575, 10 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 285	FP = 11
Test outcome NEGATIVE (clear image)	FN = 35	TN = 309
Sensitivity [%]	**89,06**	
Specificity [%]	**96,56**	
Accuracy [%]	**92,81**	
Error [%]	**7,19**	

Table 6.14: Test results - F5, resolution 1024 x 575, 1000 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 297	FP = 11
Test outcome NEGATIVE (clear image)	FN = 23	TN = 309
Sensitivity [%]	**92,81**	
Specificity [%]	**96,56**	
Accuracy [%]	**94,69**	
Error [%]	**5,31**	

Table 6.15: Test results - F5, resolution 1440 x 809, 10 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 263	FP = 8
Test outcome NEGATIVE (clear image)	FN = 57	TN = 312
Sensitivity [%]	**82,19**	
Specificity [%]	**97,50**	
Accuracy [%]	**89,84**	
Error [%]	**10,16**	

Table 6.16: Test results - F5, resolution 1440 x 809, 1000 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 264	FP = 8
Test outcome NEGATIVE (clear image)	FN = 56	TN = 312
Sensitivity [%]	**82,50**	
Specificity [%]	**97,50**	
Accuracy [%]	**90,00**	
Error [%]	**10,00**	

Table 6.17: Test results - F5, resolution 2560 x 1438, 10 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 286	FP = 10
Test outcome NEGATIVE (clear image)	FN = 34	TN = 310
Sensitivity [%]	**89,38**	
Specificity [%]	**96,88**	
Accuracy [%]	**93,13**	
Error [%]	**6,88**	

Table 6.18: Test results - F5, resolution 2560 x 1438, 1000 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 286	FP = 10
Test outcome NEGATIVE (clear image)	FN = 34	TN = 310
Sensitivity [%]	**89,38**	
Specificity [%]	**96,88**	
Accuracy [%]	**93,13**	
Error [%]	**6,88**	

It is clear from the previous tables 6.11 - 6.18 that for the correct classification of the stegogramme created by the F5 algorithm and the clear image, we achieved an accuracy of **89,84% - 95,94%** (with an error of 10,16% - 4,06%). The sensitivity of the test was **82,19% - 95,94%**.

Even though these results are not as good as the classification of OutGuess2.0 stegogrammes, they are still decent. It is essential to mention that the ANN was not learned on F5 stegogrammes. However, the sensitivity was very low, and therefore it should not be used in the corporate sector. On the contrary, the steganography algorithm F5 is less accessible than OutGuess2.0. Also, we can not create stegogrammes for resolution 4200 x 2358 - with this resolution, F5 creates empty stegogrammes with file size 0 B.

6.2 Sensitivity of the Test vs. Secret Message Length

As we mentioned, our methodology provides invariance between the test's sensitivity and secret message length. The reason is in the chapter 4. To prove this statement, we can perform ANOVA statistical hypothesis test.

The ANOVA procedure is designed to construct a statistical model describing the impact of a single categorical factor X (secret message length) on a dependent variable Y (sensitivity of the test). This test can determine whether or not there are significant differences between the means of Y at the different levels of X. In our case, we will prove that there is no significant difference between our groups of secret message lengths.

First, we determine the null hypothesis H_0: *"the mean of each population will be the same"*. Next, we determine the hypothesis H_1: *"H_0 does not apply"*. The population, also sometimes referred to as a group, is, in our case, it is the lengths of the secret message. In the following text, groups for each length of a secret message will be referred to as A (for 10 B message), B (for a 50 B message), C (for a 200 B message), D (for a 500 B message), E (for an 800 B message) and F (for a 1000 B message).

The following table 6.19 represents all the sensitivities of the test of OutGuess2.0 algorithm for each group (secret message length):

Table 6.19: Sensitivity of the test summary - OutGuess2.0.

Groups	**A**	**B**	**C**	**D**	**E**	**F**
	99,38	99,38	99,38	99,38	99,69	99,69
	99,38	99,38	99,69	99,38	99,69	99,69
Sensitivity [%]	99,38	99,38	99,38	99,38	99,38	99,38
	98,44	98,44	98,44	98,44	98,44	98,44
	97,19	97,19	97,19	97,19	97,19	97,19

The following table 6.20 presents summary statistics for the previous table 6.19:

Table 6.20: Summary statistics for OutGuess2.0.

Groups	**A**	**B**	**C**	**D**	**E**	**F**
Count	5	5	5	5	5	5
Sum	493,75	493,75	494,0625	493,75	494,375	494,375
Average	98,75	98,75	98,8125	98,75	98,875	98,875
Variance	0,927734	0,927734	1,044922	0,927734	1,152344	1,152344

The last table 6.21 is a summary for ANOVA statistical hypothesis test:

Table 6.21: ANOVA statistical hypothesis test - OutGuess2.0.

Source of Variation	**SS**	**DF**	**MS**	**F-ratio**	**P-value**
Between Groups	0,094401	5	0,01888	0,018471	0,999846
Within Groups	24,53125	24	1,022135		
Total	24,62565	29			

The conclusion from ANOVA - The P-value (0,999846) corresponding to the F-ratio of ANOVA is higher than $\alpha = 0,05$, therefore we can not reject the hypothesis H_0. In other words, the mean response for all groups by the same way, and therefore there is no dependency between sensitivity and secret message length.

In the same way, we will process results from the F5 algorithm by ANOVA. Again, we define the same hypothesis as before. The following table 6.22 represents all the sensitivities of the test of F5, for each group (secret message length):

Table 6.22: Sensitivity of the test summary - F5.

Groups	**A**	**B**	**C**	**D**	**E**	**F**
	93,44	93,44	94,38	94,69	95,31	95,94
Sensitivity [%]	89,06	89,38	90,94	90,94	91,88	92,81
	82,19	82,19	82,19	82,50	82,81	82,50
	89,38	89,38	89,38	89,38	89,38	89,38

The following table 6.23 presents summary statistics for the previous table 6.22:

Table 6.23: Summary statistics for F5.

Groups	**A**	**B**	**C**	**D**	**E**	**F**
Count	4	4	4	4	4	4
Sum	354,0625	354,375	356,875	357,5	359,375	360,625
Average	88,51563	88,59375	89,21875	89,375	89,84375	90,15625
Variance	21,76921	21,90755	26,33464	25,97656	27,89714	33,23568

Again, the last table 6.24 is a summary for ANOVA statistical hypothesis test:

Table 6.24: ANOVA statistical hypothesis test - F5.

Source of Variation	SS	DF	MS	F-ratio	P-value
Between Groups	8,614095	5	1,722819	0,06579	0,996543
Within Groups	471,3623	18	26,18679		
Total	479,9764	23			

The conclusion from ANOVA - Again, the P-value (0,996543) corresponding to the F-ratio of ANOVA is higher than $\alpha = 0,05$, therefore we can not reject the hypothesis H_0. Therefore there is no dependency between sensitivity and secret message length.

Even though it is possible to estimate these results directly from a table or graph, it is always necessary to do a statistical survey to support such a statement. That was the purpose of this chapter. Another finding is that for further testing, we may not have to distinguish sets based on the secret message length, and we can merge these testing sets into one set for each of the five resolutions.

6.3 Results of the Macroblock Filtering

In this chapter, we will present differences in the classification with and without macroblock filtering. This functionality has been described in the chapter 5.1.

Again, we will only present tables for the secret message's length 10 B and 1000 B for each resolution. The rest will be placed in appendix B at the end of this thesis.

The following tables are for the OutGuess2.0 algorithm:

Table 6.25: Application of the macroblock filtering - OutGuess2.0, resolution 800 x 449, 10 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	99,38	94,06	**5,31**
Specificity [%]	95,94	95,94	**0,00**
Accuracy [%]	97,66	95,00	**2,66**
Error [%]	2,34	5,00	**-2,66**

Table 6.26: Application of the macroblock filtering - OutGuess2.0, resolution 800 x 449, 1000 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	99,69	98,13	**1,56**
Specificity [%]	95,94	95,94	**0,00**
Accuracy [%]	97,81	97,03	**0.78**
Error [%]	2,19	2,97	**-0.78**

Table 6.27: Application of the macroblock filtering - OutGuess2.0, resolution 1024 x 575, 10 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	99,38	95,31	**4,06**
Specificity [%]	96,54	96,54	**0,00**
Accuracy [%]	97,97	95,94	**2,03**
Error [%]	2,03	4,06	**-2,03**

Table 6.28: Application of the macroblock filtering - OutGuess2.0, resolution 1024 x 575, 1000 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	99,69	97,81	**1,88**
Specificity [%]	96,56	96,56	**0,00**
Accuracy [%]	98,13	97,19	**0,94**
Error [%]	1,88	2,81	**-0,94**

Table 6.29: Application of the macroblock filtering - OutGuess2.0, resolution 1440 x 809, 10 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	99,38	94,69	**4,69**
Specificity [%]	97,50	98,13	*-0,63*
Accuracy [%]	98,44	96,41	**2,03**
Error [%]	1,56	3,59	**-2,03**

Table 6.30: Application of the macroblock filtering - OutGuess2.0, resolution 1440 x 809, 1000 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	99,38	96,25	**3,13**
Specificity [%]	97,50	98,13	*-0,63*
Accuracy [%]	98,44	97,19	**1,25**
Error [%]	1,56	2,81	**-1,25**

Table 6.31: Application of the macroblock filtering - OutGuess2.0, resolution 2560 x 1438, 10 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	98,44	96,56	**1,88**
Specificity [%]	96,88	98,13	*-1,25*
Accuracy [%]	97,66	97,34	**0,31**
Error [%]	2,34	2,66	**-0,31**

Table 6.32: Application of the macroblock filtering - OutGuess2.0, resolution 2560 x 1438, 1000 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	98,44	98,13	**0,31**
Specificity [%]	96,88	98,13	*-1,25*
Accuracy [%]	97,66	98,13	*-0,47*
Error [%]	2,34	1,88	*0,47*

Table 6.33: Application of the macroblock filtering - OutGuess2.0, resolution 4200 x 2358, 10 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	97,19	78,13	**19,06**
Specificity [%]	95,31	100,00	*-4,69*
Accuracy [%]	96,25	89,06	**7,19**
Error [%]	3,75	10,94	**-7,19**

Table 6.34: Application of the macroblock filtering - OutGuess2.0, resolution 4200 x 2358, 1000 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	97,19	79,69	**17,50**
Specificity [%]	95,31	100,00	*-4,69*
Accuracy [%]	96,25	89,84	**6,41**
Error [%]	3,75	10,16	**-6,41**

It is clear from the previous tables 6.25 - 6.34 that using the macroblock filtering function has a positive effect on the sensitivity of the test in each of the tests performed. Improvements in sensitivity were in the range from **0,31% - 19,06%**. This function also improved the overall test accuracy by up to **7,19%**. However, in some cases, the specificity of the test was reduced to **-4,69%**. However, as mentioned before, priority is given to the test's sensitivity. Deviations to the desired results are indicated by the *italics* font in the Difference column in the previous tables.

We also provide the following tables for the F5 algorithm:

Table 6.35: Application of the macroblock filtering - F5, resolution 800 x 449, 10 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	93,44	82,19	**11,25**
Specificity [%]	95,94	95,94	**0,00**
Accuracy [%]	94,69	89,06	**5,63**
Error [%]	5,31	10,94	**-5,63**

Table 6.36: Application of the macroblock filtering - F5, resolution 800 x 449, 1000 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	95,94	95,67	**0,27**
Specificity [%]	95,94	95,94	**0,00**
Accuracy [%]	95,94	95,81	**0,13**
Error [%]	4,06	4,19	**-0,13**

Table 6.37: Application of the macroblock filtering - F5, resolution 1024 x 575, 10 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	89,06	76,88	**12,19**
Specificity [%]	96,56	96,56	**0,00**
Accuracy [%]	92,81	86,72	**6,09**
Error [%]	7,19	13,28	**-6,09**

Table 6.38: Application of the macroblock filtering - F5, resolution 1024 x 575, 1000 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	92,81	84,38	**8,44**
Specificity [%]	96,56	96,56	**0,00**
Accuracy [%]	94,69	90,47	**4,22**
Error [%]	5,31	9,53	**-4,22**

Table 6.39: Application of the macroblock filtering - F5, resolution 1440 x 809, 10 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	82,19	59,69	**22,50**
Specificity [%]	97,50	98,13	*-0,63*
Accuracy [%]	89,84	78,91	**10,94**
Error [%]	10,16	21,09	**-10,94**

Table 6.40: Application of the macroblock filtering - F5, resolution 1440 x 809, 1000 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	82,50	63,13	**19,38**
Specificity [%]	97,50	98,13	*-0,63*
Accuracy [%]	90,00	80,63	**9,38**
Error [%]	10,00	19,38	**-9,38**

Table 6.41: Application of the macroblock filtering - F5, resolution 2560 x 1438, 10 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	89,38	80,00	**9,38**
Specificity [%]	96,88	98,13	*-1,25*
Accuracy [%]	93,13	89,06	**4,06**
Error [%]	6,88	10,94	**-4,06**

Table 6.42: Application of the macroblock filtering - F5, resolution 2560 x 1438, 1000 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	89,38	81,25	**8,13**
Specificity [%]	96,88	98,13	*-1,25*
Accuracy [%]	93,13	89,69	**3,44**
Error [%]	6,88	10,31	**-3,44**

Now we summarize the results presented in the tables 6.35 - 6.42 for the application of macroblock filtering function. Again, this function had a positive effect on the sensitivity of the test that was improved in the range of **0,27% - 22,50%**. This function also improved the accuracy by up to **10,94%**. Same as the OutGuess2.0 algorithm, there was a small drop in the specificity of the test by up to **-1,25%**. Deviations to the desired results are indicated by the *italics* font in the Difference column in the previous tables.

If we compare the results of both algorithms, the application of the macroblock filtering function was successful, especially the F5 algorithm.

6.4 Comparison to the Other Existing Methods

As mentioned in chapter 4, it is not easy to compare the results of different methods. Each steganalytical method processes a different type of image, image format, image resolution, embedded message length, or used steganography algorithm. Therefore, the following tables

compare the results of different methodologies based on their classification success rate - accuracy.

Table 6.43: Accuracy comparison between [89] and our method.

	[89]		Our Method	
Algorithm	OutGuess2.0	F5	OutGuess2.0	F5
Accuracy [%]	**67,9 - 97,4**	**52,2 - 91,6**	**97,66 - 97,81**	**95,16 - 95,94**
Error [%]	**32,1 - 2,6**	**47,8 - 8,4**	**2,34 - 2,19**	**4,84 - 4,06**
Conditions	NaN JPEG; Message length: 200 B - 1000 B; Resolution: 512 x 512		CF. JPEG; Message length: 200 B - 1000 B; Resolution: 800 x 449	

Table 6.44: Accuracy comparison between [93] and our method.

	[93]		Our Method	
Algorithm	OutGuess2.0	F5	OutGuess2.0	F5
Accuracy [%]	**69,4**	**69,4**	**97,66 - 97,81**	**94,69 - 95,94**
Error [%]	**30,6**	**30,6**	**2,34 - 2,19**	**5,31 - 4,06**
Conditions	CF JPEG; Message length: BPNC 0,05 - 0,2; Resolution: 640 x 480		CF JPEG; Message length: 10 B - 1000 B; Resolution: 800 x 449	

As we can see in the previous table 6.44, the authors used a different method of classification than the length of the embedded message. In this case, they utilized a BPNC - Bit Per Non-Zero Coefficient in the range 0,05 - 0,2. It is challenging to estimate equivalent message length. Therefore, we used the whole range of embedded message length, from 10 B - 1000 B, for the comparison.

Table 6.45: Accuracy comparison between [94] and our method.

	[94]		Our Method	
Algorithm	OutGuess2.0	F5	OutGuess2.0	F5
Accuracy [%]	**89,6**	**95,6**	**97,66 - 97,81**	**94,69 - 95,94**
Error [%]	**10,4**	**4,4**	**2,34 - 2,19**	**5,31 - 4,06**
Conditions	CF JPEG; Message length: BPNC 0,03 - 0,2; Resolution: 256 x 256		CF JPEG; Message length: 10 B - 1000 B; Resolution: 800 x 449	

Table 6.46: Accuracy comparison between [95] and our method.

	[95]		Our Method	
Algorithm	OutGuess2.0	F5	OutGuess2.0	F5
Accuracy [%]	**98,1**	**75,1**	**97,66 - 98,44**	**89,84 - 94,69**
Error [%]	**1,9**	**24,9**	**2,34 - 1,56**	**10,16 - 5,31**
Conditions	CF JPEG; Message length: BPNC 0,25; Resolution: 320 x 240 - 1920 x 1080		CF JPEG; Message length: 10 B; Resolution: 800 x 449 - 1440 x 809	

Table 6.47: Accuracy comparison between [67] and our method.

	[67]		Our Method	
Algorithm	OutGuess2.0	F5	OutGuess2.0	F5
Accuracy [%]	**80,2 - 92,2**	**75,0 - 94,7**	**96,25 - 97,66**	**93,13 - 94,69**
Error [%]	**18,8 - 7,8**	**25,0 - 5,3**	**3,75 - 2,34**	**6,87 - 5,31**
Conditions	CF JPEG; Message length: BPNC 0,2; Resolution: 500 x 500 - 4752 x 3168		CF JPEG; Message length: 10 B; Resolution: 800 x 449 - 4200 x 2358 (for F5, up to 2560 x 1438)	

Authors [67] are primarily focusing on mismatched steganalysis. The authors propose a method for minimalizing errors in the classification process of the steganography algorithm.

6.5 Stegosaurus Software

All testing, analysis, and results were obtained from our own software Stegosaurus v1.0, developed for these purposes. This application was implemented in Java with the support of neuroph and Jython libraries. Neuroph library provides final classification by using neural network and design of the topology itself. The Jython library provides a wrapper for the Python script to obtain the quality factor Q of JPEG images. The software supports parallel image processing. However, high hardware requirements are required to process high-resolution images with higher levels of parallelism. Application screenshots are located in Appendix C.

Also, the previous version of this software was a part of the results of the project TACR Delta "Security of Mobile Devices and Communication" - TF01000091 - Development of the application for steganography of static images for the purposes of mobile communication.

Chapter 7

Conclusion

In this thesis, we present our research about detecting the steganography content. We propose a new methodology that can detect stegogrammes primarily created by the DCT steganography algorithm OutGuess2.0 and the DCT algorithm F5. Such research can be used in the corporate sector to secure communication in the internal network and protect company data. These algorithms are readily available for download. Also, they do not require a lot of user knowledge in IT. Therefore, there is a significant possibility that they will be used in the corporate sector.

The second chapter provided a deeper study of steganography and steganalysis techniques. First, we are analyzing the current state of steganography. For these methods, we mentioned steganalysis methods to attack them. This state-of-the-art provides us with the main idea of attacking such a steganography algorithm. We were also able to identify the pros and cons of other steganalysis methods. As for cons, testing of low-resolution images, testing of grayscale images, or in some cases, low classification capability, regardless of the sensitivity of the test. As pros, we consider the applicability of steganalysis methods to more steganographic algorithms. However, sometimes it leads to a drop in classification capability.

The next chapter deals with the theoretical aspects of steganography and steganalysis. In detail, the types of cover work that could be used to embed a secret message were discussed. We also suggested a possible way to use MPEG for steganography purposes. Then we mainly dealt with images that primarily serve as cover work. In our methodology, we deal with the classification of stegogrammes that are represented by JPEG images. An essential part of this chapter is analyzing the main principle of DCT steganography algorithms - JPEG compression. Here, we have explained which part of the compression is used to embed the secret message. Finally, we analyzed the two steganography algorithms we are focusing on to attack. The result was the discovery of the "weakness" that we subsequently used in our methodology. This weakness has inspired us to implement a macroblock filtering function that filters these macroblocks that are not used for embedding the secret messages. These

macroblocks distort the result and reduce detection capability.

In the following chapter, we summarize the contribution to the area. We briefly describe the individual characteristics of our methodology and why we dealt with them. The main advantages of this research are the very high classification capabilities of stegogrammes created by the OutGuess2.0 algorithm. Furthermore, the invariance of the embedded message length on the classification capability. Support for color JPEG images in different resolutions and many more. We cannot concentrate on breaking all steganography algorithms and related issues in our effort. Instead, we contributed to the area of detection of the stegogrammes created by those algorithms that are easily accessible - OutGuess2.0 and F5.

An important part is the next chapter devoted to our methodology. Here we are described individual parts. The most important parts include the calibration process, which creates a calibrated image that simulates a stegogramme. Further, we describe a blockiness calculation that serves us as the statistical property that responds to the presence of the embedded message. We also describe the application of neural networks introduced to this issue.

In the last chapter, we present the results. In the first part of this chapter, we compare current results with our previous research. Our methodology achieves excellent results against stegogrammes created by the OutGuess2.0 algorithm. We achieved an accuracy of 96,25% - 98,44% (with an error of 3,75% - 1,56%). Mainly, the sensitivity of the emphasized test was 97,19% - 99,69%. For higher resolution images, we observe a small drop in classification capability. That may be due to the character of the ANN training set. The training set was learned only at resolutions up to 2560 x 1438. Therefore, the lack of "ANN experience" could lead to this drop in classification capabilities for higher resolutions. The next reason could be a feature of OutGuess2.0 that allows the user to insert a message to a specific length only based on the composition of the cover image. This issue has been described in detail. In this chapter, we also provide results of classification capability on the F5 algorithm. We achieved an satisfying accuracy of 89,84% - 95,94% (with an error of 10,16% - 4,06%). The sensitivity of the test was 82,06% - 95,94%. Even though these results are not as good as the classification of OutGuess2.0 stegogrammes, they are still decent. It is important to mention that this methodology is not primarily intended for F5 classification. In the second part of this chapter, we check our statement that the secret message length is insignificant against the test's sensitivity for both steganography algorithms.

By ANOVA statistical hypothesis test, we found that our statement was correct. In the last part of this chapter, we present the positive effect of the macroblock filtering function that we also proposed. For OutGuess2.0, the test's sensitivity was increased by a range from 0,31% - 19,06%. This feature also improved the overall test accuracy by up to 7,19%. For F5, the test's sensitivity was increased by a range from 0,27% - 22,50% with the improvement of the accuracy by up to 10,94%. We compared existing steganalytical methods with our solution at the end of this chapter. There was also a brief discussion about implemented

steganalysis software that was part of this research.

Finally, we want to mention possible ideas for future work. Very interesting would be to design a complete framework that would classify stegogrammes from different steganography algorithms. We have already done and published some research. Czech company AutoCont a.s. have liked this idea of a complete framework for monitoring the corporate network. On the other hand, the different Czech company, K2 atmitec s.r.o., liked the idea to secure their internal documents by some steganography mark. Since the terms of steganography and steganalysis are relatively unknown, most companies have no idea how easily they can lose their internal secret data. Another exciting direction that development could take would be to calculate the length of an embedded secret message. This feature could be the first step in extracting the contents of a secret message itself. However, this is out of the scope of steganalysis. On top of that, as we mentioned, the secret message itself is protected by encryption.

As we mentioned before, the very idea of steganography does not necessarily mean an equivalent to some illegal activity, but we must be prepared.

Bibliography

1. JOHNSON, Neil F.; JAJODIA, Sushil. Exploring steganography: Seeing the unseen. *Computer* [online]. 1998, vol. 31, no. 2, pp. 26–34 [visited on 2022-02-05]. ISSN 0018-9162. Available from DOI: 10.1109/MC.1998.4655281.

2. PROVOS, N.; HONEYMAN, P. Hide and seek: an introduction to steganography. *IEEE Security & Privacy* [online]. 2003, vol. 1, no. 3, pp. 32–44 [visited on 2022-02-05]. ISSN 1540-7993. Available from DOI: 10.1109/MSECP.2003.1203220.

3. CONWAY, Maura. Code wars: Steganography, signals intelligence, and terrorism. *Knowledge, Technology & Policy* [online]. 2003, vol. 16, no. 2, pp. 45–62 [visited on 2022-02-05]. ISSN 0897-1986. Available from DOI: 10.1007/s12130-003-1026-4.

4. SHAILENDER GUPTA, Ankur Goyal; BHUSHAN, Bharat. Information Hiding Using Least Significant Bit Steganography and Cryptography. *International Journal of Modern Education and Computer Science* [online]. 2012-06-25, vol. 4, no. 6, pp. 27–34 [visited on 2022-02-05]. ISSN 20750161. Available from DOI: 10.5815/ijmecs.2012.06.04.

5. XIA, Zhihua; WANG, Xinhui; SUN, Xingming; WANG, Baowei. Steganalysis of least significant bit matching using multi-order differences. *Security and Communication Networks* [online]. 2014, vol. 7, no. 8, pp. 1283–1291 [visited on 2022-02-05]. ISSN 19390114. Available from DOI: 10.1002/sec.864.

6. KUO, Wen-Chung; CHEN, Yi-Heng; CHUANG, Chen-Tsun. High-capacity steganographic method based on division arithmetic and generalized exploiting modification direction. *Journal of Information Hiding and Multimedia Signal Processing*. 2014, vol. 5, no. 2, pp. 213–222. ISSN 2073-4212.

7. MARVEL, L.M.; BONCELET, C.G.; RETTER, C.T. Spread spectrum image steganography. *IEEE Transactions on Image Processing* [online]. 1999, vol. 8, no. 8, pp. 1075–1083 [visited on 2022-02-05]. ISSN 10577149. Available from DOI: 10.1109/83.777088.

8. BAILEY, Karen; CURRAN, Kevin. An evaluation of image based steganography methods. *Multimedia Tools and Applications* [online]. 2006, vol. 30, no. 1, pp. 55–88 [visited on 2022-02-05]. ISSN 1380-7501. Available from DOI: 10.1007/s11042-006-0008-4.

9. YADAV, Rajkumar; RAVI, Saini; GAURAV, Chawla. A Novel Approach For Image Steganography In Spatial Do-Main Using Last Two Bits of Pixel Value. *International Journal of Security (IJS)*. 2011, vol. 5, no. 2, p. 51.

10. ALAM, Fahim Irfan; FATHENA, Khanam Bappee; FARID, Uddin Ahmed Khondker. An investigation into encrypted message hiding through images using LSB. *International Journal of Engineering Science and Technology (IJEST)*. 2011, vol. 3, no. 2, pp. 948–960. ISSN 0975-5462.

11. BANDYOPADHYAY, Debiprasad; DASGUPTA, Kousik; K, Mandal J.; DUTTA, Paramartha. A Novel Secure Image Steganography Method Based on Chaos Theory in Spatial Domain. *International Journal of Security, Privacy and Trust Management* [online]. 2014-02-28, vol. 3, no. 1, pp. 11–22 [visited on 2022-02-05]. ISSN 23194103. Available from DOI: 10.5121/ijsptm.2014.3102.

12. CHANG, Chin-Chen; HSIAO, Ju-Yuan; CHAN, Chi-Shiang. Finding optimal least-significant-bit substitution in image hiding by dynamic programming strategy. *Pattern Recognition* [online]. 2003, vol. 36, no. 7, pp. 1583–1595 [visited on 2022-02-05]. ISSN 00313203. Available from DOI: 10.1016/S0031-3203(02)00289-3.

13. WANG, Ran-Zan; LIN, Chi-Fang; LIN, Ja-Chen. Image hiding by optimal LSB substitution and genetic algorithm. *Pattern Recognition* [online]. 2001, vol. 34, no. 3, pp. 671–683 [visited on 2022-02-05]. ISSN 00313203. Available from DOI: 10.1016/S0031-3203(00)00015-7.

14. CHAN, Chi-Kwong; CHENG, L.M. Hiding data in images by simple LSB substitution. *Pattern Recognition* [online]. 2004, vol. 37, no. 3, pp. 469–474 [visited on 2022-02-05]. ISSN 00313203. Available from DOI: 10.1016/j.patcog.2003.08.007.

15. LEE, Y.K.; CHEN, L.H. High capacity image steganographic model. *IEE Proceedings - Vision, Image, and Signal Processing* [online]. 2000, vol. 147, no. 3, pp. 288–294 [visited on 2022-02-05]. ISSN 1350245X. Available from DOI: 10.1049/ip-vis:20000341.

16. VYAS, Krati; PAL, B. L. A proposed method in image steganography to improve image quality with LSB technique. *International Journal of Advanced Research in Computer and Communication Engineering*. 2014, vol. 3, no. 2, pp. 5246–5251.

17. AL-SHATNAWI, Atallah M. A new method in image steganography with improved image quality. *Applied Mathematical Sciences*. 2012, vol. 6, no. 79, pp. 3907–3915.

18. PRIYA, S. Shanmuga; MAHESH, K.; KUPPUSAMY, K. Efficient Steganography Method to Implement Selected Lease Significant Bits in Spatial Domain (SLSB–SD). *International Journal of Engineering Research and Applications (IJERA)*. 2012, vol. 2, no. 3, pp. 2632–2637. ISSN 2248-9622.

19. JUNEJA, Mamta; SANDHU, Parvinder Singh. A New Approach for Information Security using an Improved Steganography Technique. *Journal of Information Processing Systems* [online]. 2013-09-30, vol. 9, no. 3, pp. 405–424 [visited on 2022-02-05]. ISSN 1976-913X. Available from DOI: 10.3745/JIPS.2013.9.3.405.

20. LASKAR, Shamim Ahmed; HEMACHANDRAN, Kattamanchi. Steganography based on random pixel selection for efficient data hiding. *International Journal of Computer Engineering and Technology*. 2013, vol. 4, no. 2, pp. 31–44.

21. ASAD, Muhammad; GILANI, Junaid; KHALID, Adnan. An enhanced least significant bit modification technique for audio steganography. In: *International Conference on Computer Networks and Information Technology* [online]. Abbottabad, Pakistan: IEEE, 2011, pp. 143–147 [visited on 2022-02-05]. ISBN 978-1-61284-940-9. ISSN 2223-6317. Available from DOI: 10.1109/ICCNIT.2011.6020921.

22. HINDI, Amjad Y.; DWAIRI, Majed O.; ALQADI, Yiad A. A novel technique for data steganography. *Engineering, Technology & Applied Science Research*. 2019, vol. 9, no. 6, pp. 4942–4945. Available also from: https://www.researchgate.net/profile/Majed-Dwairi/publication/337739962_A_Novel_Technique_for_Data_Steganography/links/5e60b348299bf1bdb8544034/A-Novel-Technique-for-Data-Steganography.pdf.

23. WAZIRALI, Ranyiah; ALASMARY, Waleed; MAHMOUD, Mohamed M. E. A.; ALHINDI, Ahmad. An Optimized Steganography Hiding Capacity and Imperceptibly Using Genetic Algorithms. *IEEE Access* [online]. 2019, vol. 7, no. 1, pp. 133496–133508 [visited on 2022-02-05]. ISSN 2169-3536. Available from DOI: 10.1109/ACCESS.2019.2941440.

24. ZHANG, Xinpeng; WANG, Shuozhong. Steganography using multiple-base notational system and human vision sensitivity. *IEEE Signal Processing Letters* [online]. 2005, vol. 12, no. 1, pp. 67–70 [visited on 2022-02-05]. ISSN 1070-9908. Available from DOI: 10.1109/LSP.2004.838214.

25. NGUYEN, Bui Cong; YOON, Sang Moon; LEE, Heung-Kyu. Multi Bit Plane Image Steganography. In: *Digital Watermarking* [online]. 1st ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 61–70 [visited on 2022-02-05]. ISBN 978-3-540-48825-5. Available from DOI: 10.1007/11922841_6.

26. WU, Da-Chun; TSAI, Wen-Hsiang. A steganographic method for images by pixel-value differencing. *Pattern Recognition Letters* [online]. 2003, vol. 24, no. 9-10, pp. 1613–1626 [visited on 2022-02-05]. ISSN 01678655. Available from DOI: 10.1016/S0167-8655(02)00402-6.

27. CHANG, Ko-Chin. A Novel Image Steganographic Method Using Tri-way Pixel-Value Differencing. *Journal of multimedia*. 2008, vol. 3, no. 2, pp. 37–44.

28. LEE, Yen-Po; LEE, Jen-Chun; CHEN, Wei-Kuei; CHANG, Ko-Chin; SU, Ing-Jiunn; CHANG, Chien-Ping. High-payload image hiding with quality recovery using tri-way pixel-value differencing. *Information Sciences* [online]. 2012, vol. 191, no. 1, pp. 214–225 [visited on 2022-02-05]. ISSN 00200255. Available from DOI: 10.1016/j.ins.2012.01.002.

29. POTDAR, V.M.; CHANG, E. Grey level modification sleganography for secret communication. In: *2nd IEEE International Conference on Industrial Informatics, 2004. INDIN '04. 2004* [online]. Berlin, Germany: IEEE, 2004, pp. 223–228 [visited on 2022-02-05]. ISBN 0-7803-8513-6. ISSN 0-7803-8513-6. Available from DOI: 10.1109/INDIN.2004.1417333.

30. LADWANI, Vandana M.; MURTHY, K. Srikanta. A New Approach to Securing Image. *International Journal of Advance Research in Computer and Communication Engineering*. 2015, vol. 4, no. 4, pp. 2319–5940.

31. UPHAM, D. *Steganographic algorithm JSteg* [online]. -: -, 1993 [visited on 2022-02-05]. Available from: http://zooid.%20org/~%20paul/crypto/jsteg.

32. WESTFELD, Andreas. F5—a steganographic algorithm. *International workshop on information hiding*. 2001, vol. 2001, no. 1, pp. 289–302.

33. SOLANKI, K.; SARKAR, A.; MANJUNATH, B. S. Yet another steganographic scheme that resists blind steganalysis. *Proceedings of 9th Information Hiding Workshop*. 2007, vol. 2007, no. 1, p. 1.

34. YU, Lifang; ZHAO, Yao; NI, Rongrong; SHI, Yun Q. A high-performance YASS-like scheme using randomized big-blocks. In: *2010 IEEE International Conference on Multimedia and Expo* [online]. Singapore: IEEE, 2010, pp. 474–479 [visited on 2022-02-05]. ISBN 978-1-4244-7491-2. ISSN 1945-7871. Available from DOI: 10.1109/ICME.2010.5582542.

35. LU, Shao-Ping. Large-capacity image steganography based on invertible neural networks. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, vol. 2021, no. 1, pp. 10816–10825.

36. DUAN, Xintao; GUO, Daidou; LIU, Nao; LI, Baoxia; GOU, Mengxiao; QIN, Chuan. A New High Capacity Image Steganography Method Combined With Image Elliptic Curve Cryptography and Deep Neural Network. *IEEE Access* [online]. 2020, vol. 8, no. 1, pp. 25777–25788 [visited on 2022-02-05]. ISSN 2169-3536. Available from DOI: 10.1109/ACCESS.2020.2971528.

37. AL-ATABY, Ali; AL-NAIMA, Fawzi. A modified high capacity image steganography technique based on wavelet transform. *Changes*. 2008, vol. 2008, no. 4, p. 6.

38. TALELE, Gandhe; KESKAR, A. G. Steganography security for copyright protection of digital images using dwt. *International Journal of Computer and Network Security*. 2010, vol. 2, no. 4, p. 1.

39. KUMAR, Vijay; KUMAR, Dinesh. Performance evaluation of DWT based image steganography. In: *2010 IEEE 2nd International Advance Computing Conference (IACC)* [online]. Patiala, India: IEEE, 2010, pp. 223–228 [visited on 2022-02-05]. ISBN 978-1-4244-4790-9. ISSN 978-1-4244-4790-9. Available from DOI: 10.1109/IADCC.2010.5423005.

40. WESTFELD, Andreas; PFITZMANN, Andreas. Attacks on Steganographic Systems. In: *Information Hiding* [online]. 1st ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 61–76 [visited on 2022-02-05]. ISBN 978-3-540-67182-4. Available from DOI: 10.1007/10719724_5.

41. PROVOS, Neils; HONEYMAN, Peter. Detecting steganographic content on the internet. *Center for Information Technology Integration*. 2001, vol. 2001, no. 1, p. 1.

42. FRIDRICH, J.; GOLJAN, M.; DU, Rui. Detecting LSB steganography in color, and gray-scale images. *IEEE Multimedia* [online]. 2001, vol. 8, no. 4, pp. 22–28 [visited on 2022-02-05]. ISSN 1070986X. Available from DOI: 10.1109/93.959097.

43. FRIDRICH, Jessica; III, Edward J. Delp; WONG, Ping W.; GOLJAN, Miroslav. On estimation of secret message length in LSB steganography in spatial domain. *Security, steganography, and watermarking of multimedia contents VI*. 2004, vol. 2004, no. 5306, pp. 23–34. Available from DOI: 10.1117/12.521350.

44. DUMITRESCU, Sorina; WU, Xiaolin; WANG, Zhe. Detection of LSB Steganography via Sample Pair Analysis. In: *Information Hiding* [online]. 2578th ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003-12-18, pp. 355–372 [visited on 2022-02-05]. ISBN 978-3-540-00421-9. Available from DOI: 10.1007/3-540-36415-3_23.

45. KER, Andrew D. A General Framework for Structural Steganalysis of LSB Replacement. In: *Information Hiding* [online]. 3727th ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 296–311 [visited on 2022-02-05]. ISBN 978-3-540-29039-1. Available from DOI: 10.1007/11558859_22.

46. KER, Andrew D.; III, Edward J. Delp; WONG, Ping Wah. Fourth-order structural steganalysis and analysis of cover assumptions. *Security, Steganography, and Watermarking of Multimedia Contents VIII*. 2006, vol. 2006, no. 6072, pp. 25–38. Available from DOI: 10.1117/12.642920.

47. LI, Bin; FANG, Yanmei; HUANG, Jiwu. Steganalysis of Multiple-Base Notational System Steganography. *IEEE Signal Processing Letters* [online]. 2008, vol. 15, no. 1, pp. 493–496 [visited on 2022-02-05]. ISSN 1070-9908. Available from DOI: 10.1109/LSP.2008.924000.

48. LUO, Xiangyang; LIU, Fenlin; YANG, Chunfang; LIAN, Shiguo; ZENG, Ying. Steganalysis of adaptive image steganography in multiple gray code bit-planes. *Multimedia Tools and Applications* [online]. 2012, vol. 57, no. 3, pp. 651–667 [visited on 2022-02-05]. ISSN 1380-7501. Available from DOI: 10.1007/s11042-010-0663-3.

49. BARBIER, Johann; MAYOURA, Kichenakoumar. Steganalysis of Multi Bit Plane Image Steganography. In: *Digital Watermarking* [online]. 1st ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 99–111 [visited on 2022-02-05]. ISBN 978-3-540-92237-7. Available from DOI: 10.1007/978-3-540-92238-4_9.

50. SABETI, Vajiheh; SAMAVI, Shadrokh; MAHDAVI, Mojtaba; SHIRANI, Shahram. Steganalysis and payload estimation of embedding in pixel differences using neural networks. *Pattern Recognition* [online]. 2010, vol. 43, no. 1, pp. 405–415 [visited on 2022-02-05]. ISSN 00313203. Available from DOI: 10.1016/j.patcog.2009.06.006.

51. SABETI, Vajiheh; SAMAVI, Shadrokh; MAHDAVI, Mojtaba; SHIRANI, Shahram. Steganalysis of Pixel-Value Differencing Steganographic Method. In: *2007 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing* [online]. Victoria, BC, Canada: IEEE, 2007, pp. 292–295 [visited on 2022-02-05]. ISBN 1-4244-1190-4. ISSN 1555-5798. Available from DOI: 10.1109/PACRIM.2007.4313232.

52. LI, Xiaolong; LI, Bin; LUO, Xiangyang; YANG, Bin; ZHU, Ruihui. Steganalysis of a PVD-based content adaptive image steganography. *Signal Processing* [online]. 2013, vol. 93, no. 9, pp. 2529–2538 [visited on 2022-02-05]. ISSN 01651684. Available from DOI: 10.1016/j.sigpro.2013.03.029.

53. BUI, Cong Nguyen. Steganalysis method defeating the modified pixel-value differencing steganography. *International Journal of Innovative Computing Information and Control*. 2010, vol. 6, no. 1, pp. 3193–3203.

54. KODOVSKY, Jan; FRIDRICH, Jessica. Quantitative Structural Steganalysis of Jsteg. *IEEE Transactions on Information Forensics and Security* [online]. 2010, vol. 5, no. 4, pp. 681–693 [visited on 2022-02-05]. ISSN 1556-6013. Available from DOI: 10.1109/TIFS.2010.2056684.

55. LEE, Yeuan-Kuen; HWANG, Shih-Yu; OU, Zhan-He. A novel quantity based on clipping statistics for Jsteg steganalysis. *8th IASTED Int. Con. on Signal & Image Processing (SIP 2006)*. 2006, vol. 2006, no. 1, pp. 98–117.

56. THAI, Thanh Hai; COGRANNE, Remi; RETRAINT, Florent. Statistical Model of Quantized DCT Coefficients: Application in the Steganalysis of Jsteg Algorithm. *IEEE Transactions on Image Processing* [online]. 2014, vol. 23, no. 5, pp. 1980–1993 [visited on 2022-02-05]. ISSN 1057-7149. Available from DOI: 10.1109/TIP.2014.2310126.

57. LYU, Siwei; III, Edward J. Delp; WONG, Ping W.; FARID, Hany. Steganalysis using color wavelet statistics and one-class support vector machines. *Security, steganography, and watermarking of multimedia contents VI*. 2004, vol. 6, no. 5306, pp. 35–45. Available from DOI: 10.1117/12.526012.

58. AVCIBAS, I.; MEMON, N.; SANKUR, B. Steganalysis using image quality metrics. *IEEE Transactions on Image Processing* [online]. 2003, vol. 12, no. 2, pp. 221–229 [visited on 2022-02-05]. ISSN 1057-7149. Available from DOI: 10.1109/TIP.2002.807363.

59. FRIDRICH, Jessica; GOLJAN, Miroslav; HOGEA, Dorin. Attacking the outguess. *Proceedings of the ACM Workshop on Multimedia and Security*. 2002, vol. 2002, no. 2002, p. 4.

60. FU, Dongdong; SHI, Yun; ZOU, Dekun; XUAN, Guorong. JPEG Steganalysis Using Empirical Transition Matrix in Block DCT Domain. In: *2006 IEEE Workshop on Multimedia Signal Processing* [online]. Victoria, BC, Canada: IEEE, 2006, pp. 310–313 [visited on 2022-02-06]. ISBN 0-7803-9752-5. ISSN 0-7803-9751-7. Available from DOI: 10.1109/MMSP.2006.285320.

61. FRIDRICH, Jessica. Feature-Based Steganalysis for JPEG Images and Its Implications for Future Design of Steganographic Schemes. In: *Information Hiding* [online]. 3200th ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 67–81 [visited on 2022-02-06]. ISBN 978-3-540-24207-9. Available from DOI: 10.1007/978-3-540-30114-1_6.

62. PEVNY, Tomas; FRIDRICH, Jessica. Merging Markov and DCT features for multi-class JPEG steganalysis. *Security, steganography, and watermarking of multimedia contents*. 2007, vol. 9, no. 6505, p. 13. Available from DOI: 10.1117/12.696774.

63. FRIDRICH, Jessica; GOLJAN, Miroslav; HOGEA, Dorin. Steganalysis of JPEG Images: Breaking the F5 Algorithm. In: *Information Hiding* [online]. 2578th ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003-12-18, pp. 310–323 [visited on 2022-02-06]. ISBN 978-3-540-00421-9. Available from DOI: 10.1007/3-540-36415-3_20.

64. LI, Bin; HUANG, Jiwu; SHI, Yun Qing. Steganalysis of YASS. *IEEE Transactions on Information Forensics and Security* [online]. 2009, vol. 4, no. 3, pp. 369–382 [visited on 2022-02-06]. ISSN 1556-6013. Available from DOI: 10.1109/TIFS.2009.2025841.

65. MEMON, Nasir D.; KODOVSKÝ, Jan; PEVNÝ, Tomáš; DITTMANN, Jana; ALATTAR, Adnan M.; FRIDRICH, Jessica; III, Edward J. Delp. Modern steganalysis can detect YASS. *Media Forensics and Security*. 2010, vol. 2, no. 7541, p. 11. Available from DOI: 10.1117/12.838768.

66. LIU, Qingzhong. Steganalysis of DCT-embedding based adaptive steganography and YASS. In: *Proceedings of the thirteenth ACM multimedia workshop on Multimedia and security - MM&Sec '11* [online]. New York, New York, USA: ACM Press, 2011, pp. 77– [visited on 2022-02-06]. ISBN 9781450308069. ISSN 9781450308069. Available from DOI: 10.1145/2037252.2037267.

67. YANG, Yong; KONG, Xiangwei; WANG, Bo; REN, Ke; GUO, Yanqing. Steganalysis on Internet images via domain adaptive classifier. *Neurocomputing* [online]. 2019, vol. 351, no. 1, pp. 205–216 [visited on 2022-02-06]. ISSN 09252312. Available from DOI: 10.1016/j.neucom.2019.04.025.

68. YOUSFI, Yassine; FRIDRICH, Jessica. An Intriguing Struggle of CNNs in JPEG Steganalysis and the OneHot Solution. *IEEE Signal Processing Letters* [online]. 2020, vol. 27, no. 1, pp. 830–834 [visited on 2022-02-06]. ISSN 1070-9908. Available from DOI: 10.1109/LSP.2020.2993959.

69. REZAEI, Mohammad; MOGHADAM, Saeed Montazeri. Impact of Steganography on JPEG File Size. In: *2019 27th Iranian Conference on Electrical Engineering (ICEE)* [online]. Yazd, Iran: IEEE, 2019, pp. 1869–1873 [visited on 2022-02-06]. ISBN 978-1-7281-1508-5. ISSN 2642-9527. Available from DOI: 10.1109/IranianCEE.2019.8786506.

70. LIU, Jiufen; YANG, Chunfang; WANG, Junchao; SHI, Yanan. Stego key recovery method for F5 steganography with matrix encoding. *EURASIP Journal on Image and Video Processing* [online]. 2020, vol. 2020, no. 1, p. 17 [visited on 2022-02-06]. ISSN 1687-5281. Available from DOI: 10.1186/s13640-020-00526-2.

71. PETITCOLAS, F.A.P.; ANDERSON, R.J.; KUHN, M.G. Information hiding-a survey. *Proceedings of the IEEE* [online]. 1999, vol. 87, no. 7, pp. 1062–1078 [visited on 2022-02-06]. ISSN 00189219. Available from DOI: 10.1109/5.771065.

72. *INTERNET SECURITY THREAT REPORT - ISTR* [online]. Symantec: Symantec, 2017 [visited on 2022-02-06]. Available from: https://www.symantec.com/content/dam/symantec/docs/reports/istr-22-2017-en.pdf.

73. NOTO, Mark. MP3Stego: Hiding text in MP3 files. *Sans Institute*. 2001, vol. 2001, no. 1, p. 5.

74. SHAKARANARAYANAN, M. *Audio file steganography* [online] [visited on 2022-02-06]. Available from: http://www.cise.ufl.edu/~smanamal/steganography.htm.

75. [Online]. San Francisco (CA): Wikimedia Foundation, 2001- [visited on 2022-02-06]. Available from: https://commons.wikimedia.org/wiki/File:IPB%5C_images%5C_sequence.png.

76. SHANABLEH, Tamer. Matrix encoding for data hiding using multilayer video coding and transcoding solutions. *Signal Processing: Image Communication* [online]. 2012, vol. 27, no. 9, pp. 1025–1034 [visited on 2022-02-06]. ISSN 09235965. Available from DOI: 10.1016/j.image.2012.06.003.

77. FANG, Ding-Yu; CHANG, Long-Wen. Data hiding for digital video with phase of motion vector. In: *2006 IEEE International Symposium on Circuits and Systems* [online]. Kos, Greece: IEEE, 2006, pp. 4– [visited on 2022-02-06]. ISBN 0-7803-9389-9. ISSN 0271-4302. Available from DOI: 10.1109/ISCAS.2006.1692862.

78. ALY, Hussein A. Data Hiding in Motion Vectors of Compressed Video Based on Their Associated Prediction Error. *IEEE Transactions on Information Forensics and Security* [online]. 2011, vol. 6, no. 1, pp. 14–18 [visited on 2022-02-06]. ISSN 1556-6013. Available from DOI: 10.1109/TIFS.2010.2090520.

79. KANCHERLA, K.; MUKKAMALA, S. Video steganalysis using motion estimation. In: *2009 International Joint Conference on Neural Networks* [online]. Atlanta, GA, USA: IEEE, 2009, pp. 1510–1515 [visited on 2022-02-06]. ISBN 978-1-4244-3548-7. ISSN 2161-4393. Available from DOI: 10.1109/IJCNN.2009.5179032.

80. *STEGANOGRAPHY - Information Technologies for IPR Protection* [online] [visited on 2022-02-06]. Available from: http://www.cmlab.csie.ntu.edu.tw/~ipr/ipr2006/data/lecture/Lecture11%5C%20-%5C%20Steganography.pdf.

81. SOJKA, Eduard. *Digitální zpracování a analýza obrazů*. 1. vyd. Ostrava: VŠB-Technická univerzita, 2000. ISBN 80-7078-746-5.

82. HOLOŠKA, Jiří. *Artificial Intelligence Applied on Cryptoanalysis Aimed on Revealing Weaknesses of Modern Cryptology and Computer Security*. Univerzita Tomáše Bati ve Zlíně, 2011. PHD Thesis. Univerzita Tomáše Bati ve Zlíně.

83. HENDRYCH, Jakub; KUNČICKÝ, Radim; LIČEV, Lačezar. New Approach to Steganography Detection via Steganalysis Framework. In: *Proceedings of the Second International Scientific Conference "Intelligent Information Technologies for Industry" (IITI'17)* [online]. 679th ed. Cham: Springer International Publishing, 2018, pp. 496–503 [visited on 2022-02-06]. ISBN 978-3-319-68320-1. Available from DOI: 10.1007/978-3-319-68321-8_51.

84. PROVOS, Niels. Defending against statistical steganalysis. *10th USENIX Security Symposium (USENIX Security 01)*. 2001, vol. 2001, no. 1, p. 1.

85. HENDRYCH, Jakub; KUNČICKÝ, Radim. Detector of the steganography images with the application of artificial neural network. In: *International Multidisciplinary Scientific GeoConference: SGEM 17*. 17th ed. Surveying Geology & Mining Ecology Management (SGEM), 2017, pp. 255–261. Available from DOI: 10.5593/sgem2017/21.

86. KUNČICKÝ, Radim; HENDRYCH, Jakub. Statistical Analysis of Steganalytical Method for Steghide Detection. In: *International Multidisciplinary Scientific GeoConference: SGEM 17*. Surveying Geology & Mining Ecology Management (SGEM), 2017, pp. 611–616. Available from DOI: 10.5593/sgem2017/21.

87. HENDRYCH, Jakub. Catch the Stegogramme – Detection of the Steganogpraphy Content with the Application of ANN. In: *Proceedings of the Ph.D. Workshop of Faculty of Electrical Engineering and Computer Science, WOFEX 2017*. Ostrava: VŠB-TUO, 2017, pp. 201–206. ISBN 978-80-248-4056-7. ISSN 978-80-248-4056-7.

88. LICEV, Lacezar; HENDRYCH, Jakub; KUNCICKY, Radim. Neural Stegoclassifier. In: *2016 6th International Conference on IT Convergence and Security (ICITCS)* [online]. Prague, Czech Republic: IEEE, 2016, pp. 1–3 [visited on 2022-02-06]. ISBN 978-1-5090-3765-0. ISSN 978-1-5090-3765-0. Available from DOI: 10.1109/ICITCS.2016.7740355.

89. CHEN, C.L. Philip; CHEN, Mei-Ching; AGAIAN, Sos; ZHOU, Yicong; ROY, Anuradha; RODRIGUEZ, Benjamin M. A pattern recognition system for JPEG steganography detection. *Optics Communications* [online]. 2012, vol. 285, no. 21-22, pp. 4252–4261 [visited on 2022-02-06]. ISSN 00304018. Available from DOI: 10.1016/j.optcom.2012.06.049.

90. LIU, Qingzhong; SUNG, Andrew H.; QIAO, Mengyu; CHEN, Zhongxue; RIBEIRO, Bernardete. An improved approach to steganalysis of JPEG images. *Information Sciences* [online]. 2010, vol. 180, no. 9, pp. 1643–1655 [visited on 2022-02-06]. ISSN 00200255. Available from DOI: 10.1016/j.ins.2010.01.001.

91. GUL, Gokhan; KURUGOLLU, Fatih. A New Methodology in Steganalysis: Breaking Highly Undetectable Steganograpy (HUGO). In: *Information Hiding* [online]. 6958th ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 71–84 [visited on 2022-02-06]. ISBN 978-3-642-24177-2. Available from DOI: 10.1007/978-3-642-24178-9_6.

92. HENDRYCH, Jakub; LIČEV, Lačezar. Advanced Methods of Detection of the Steganography Content. In: *AETA 2018 - Recent Advances in Electrical Engineering and Related Sciences: Theory and Application* [online]. 554th ed. Cham: Springer International Publishing, 2020, pp. 484–493 [visited on 2022-02-06]. ISBN 978-3-030-14906-2. Available from DOI: 10.1007/978-3-030-14907-9_47.

93. BERA, Swagota; SHARMA, Monisha; SIKHAR, S. Subramanya; DWIVEDI, Atul. An efficient blind steganalysis using higher order statistics for the neighborhood difference matrix. In: *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)* [online]. Chennai, India: IEEE, 2016, pp. 211–215 [visited on 2022-02-06]. ISBN 978-1-4673-9338-6. ISSN 978-1-4673-9338-6. Available from DOI: 10.1109/WiSPNET.2016.7566122.

94. BHASIN, Veenu; BEDI, Punam. Steganalysis of colored JPEG images using ensemble of extreme learning machines. *Nternational Journal on Recent Trends in Engineering & Technology*. 2014, vol. 11, no. 2, p. 63.

95. BRODA, Martin; HAJDUK, Vladimír; LEVICKÝ, Dušan. Universal statistical steganalytic method. *Journal of Electrical Engineering*. 2017, vol. 68, no. 2, pp. 117–124. Available from DOI: 10.1515/jee-2017âĂŞ0016.

List of own publication activities and other outcomes

Publications and Outcomes Related to Thesis

1. HENDRYCH, Jakub; LIČEV, Lačezar. Advanced Methods of Detection of the Steganography Content. In: *AETA 2018 - Recent Advances in Electrical Engineering and Related Sciences: Theory and Application* [online]. 554th ed. Cham: Springer International Publishing, 2020, pp. 484–493 [visited on 2022-02-06]. ISBN 978-3-030-14906-2. ISSN 18761100. Available from DOI: 10.1007/978-3-030-14907-9_47. [SCOPUS indexed].
2. HENDRYCH, Jakub; KUNČICKÝ, Radim; LIČEV, Lačezar. New Approach to Steganography Detection via Steganalysis Framework. In: *Proceedings of the Second International Scientific Conference "Intelligent Information Technologies for Industry" (IITI'17)* [online]. 679th ed. Cham: Springer International Publishing, 2018, pp. 496–503 [visited on 2022-02-06]. ISBN 978-3-319-68320-1. ISSN 21945357. Available from DOI: 10.1007/978-3-319-68321-8_51. [SCOPUS and WOS indexed].
3. HENDRYCH, Jakub; KUNČICKÝ, Radim. Detector of the steganography images with the application of artificial neural network. In: *International Multidisciplinary Scientific GeoConference: SGEM 17*. 17th ed. Surveying Geology & Mining Ecology Management (SGEM), 2017, pp. 255–261. ISBN 978-619740826-3. ISSN 13142704. Available from DOI: 10.5593/sgem2017/21/S07.033. [SCOPUS indexed].
4. KUNČICKÝ, Radim; HENDRYCH, Jakub. Statistical Analysis of Steganalytical Method for Steghide Detection. In: *International Multidisciplinary Scientific GeoConference: SGEM 17*. 17th ed. Surveying Geology & Mining Ecology Management (SGEM), 2017, pp. 611–616. ISBN 978-619740826-3. ISSN 13142704. Available from DOI: 10.5593/sgem2017/21/S07.078. [SCOPUS indexed].
5. HENDRYCH, Jakub. Catch the Stegogramme – Detection of the Steganogpraphy Content with the Application of ANN. In: *Proceedings of the Ph.D. Workshop of Faculty of*

Electrical Engineering and Computer Science, WOFEX 2017. Ostrava: VŠB-TUO, 2017, pp. 201–206. ISBN 978-80-248-4056-7. ISSN 978-80-248-4056-7.

6. LICEV, Lacezar; HENDRYCH, Jakub; KUNCICKY, Radim. Neural Stegoclassifier. In: *2016 6th International Conference on IT Convergence and Security (ICITCS)* [online]. Prague, Czech Republic: IEEE, 2016, pp. 1–3 [visited on 2022-02-06]. ISBN 978-150903764-3. Available from DOI: 10.1109/ICITCS.2016.7740355. [SCOPUS indexed].

Publications and Outcomes Not Related to Thesis

1. LIČEV, Lačezar; HENDRYCH, Jakub; TOMEČEK, Jan; ČAJKA, Radim; KREJSA, Martin. Monitoring of Excessive Deformation of Steel Structure Extra-High Voltage Pylons. *Periodica Polytechnica Civil Engineering* [online]. 2018, vol. 62, no. 2, pp. 323–329 [visited on 2022-02-06]. ISSN 05536626. Available from DOI: 10.3311/PPci.11253. [SCOPUS and WOS indexed; SJR 0,36].
2. KROUPOVÁ, Ivana; LICHÝ, Petr; LIČEV, Lačezar; HENDRYCH, Jakub; SOUČEK, Kamil. Evaluation of properties of cast metal foams with irregular inner structure. *Archives of Metallurgy and Materials 63*. 2018, vol. 63, no. 4, pp. 1847–1851. ISSN 17333490. Available from DOI: 10.24425/amm.2018.125114. [SCOPUS and WOS indexed; SJR 0,254].
3. LIČEV, Lačezar; HENDRYCH, Jakub; KUNČICKÝ, Radim; KOVÁŘOVÁ, Kateřina; KUMPOVÁ, Ivana. Evaluation of Sandstone Internal Structure with Application of Micro-CT and FOTOM System. In: *Proceedings of the Second International Scientific Conference "Intelligent Information Technologies for Industry" (IITI'17)* [online]. 680th ed. Cham: Springer International Publishing, 2018, pp. 332–339 [visited on 2022-02-06]. ISBN 978-3-319-68323-2. Available from DOI: 10.1007/978-3-319-68324-9_36. [SCOPUS and WOS indexed].
4. LIČEV, Lačezar; HENDRYCH, Jakub; KROUPOVÁ, Ivana; SOUČEK, Kamil. Evaluation Of Materials Internal Structure With The Application Of Micro-Ct And Fotom System. In: *XXIV. konference SDMG*. Ostrava: VŠB-TUO, 2017, p. 9. ISBN 978-80-248-4114-4. ISSN 978-80-248-4112-0.
5. KUNCICKY, Radim; LICEV, Lacezar; KRUMNIKL, Michal; FEBEROVA, Karolina; HENDRYCH, Jakub. Sine Inverter Controller with 8 Bit Microcontroller. *Advances in Electrical and Electronic Engineering* [online]. 2016-09-27, vol. 14, no. 3, pp. 287–294 [visited on 2022-02-06]. ISSN 13361376. Available from DOI: 10.15598/aeee.v14i3.1715. [SCOPUS and WOS indexed; SJR 0,226].

6. LICEV, Lacezar; FEBEROVA, Karolina; TOMECEK, Jan; HENDRYCH, Jakub. An Enhanced Method for Automatic Detection and Segmentation of Carotid Artery in Ultrasound Images. In: *Proceedings of the 17th International Conference on Computer Systems and Technologies 2016* [online]. New York, NY, USA: ACM, 2016-06-23, pp. 206–213 [visited on 2022-02-06]. ISBN 978-145034182-0. ISSN 9781450341820. Available from DOI: 10.1145/2983468.2983483. [SCOPUS and WOS indexed].

7. LICHEV, Lachezar; JANEROVA, K.; TOMECEK, J.; HENDRYCH, Jakub. A NEW PHOTOGRAMMETRIC TOOL FOR THE AUTOMATIC DETECTION AND SEGMENTATION OF CIRCULAR OBJECTS IN LOW-QUALITY IMAGES. In: *15th International Multidisciplinary Scientific GeoConference SGEM2015: Informatics, Geoinformatics and Remote Sensing* [online]. Sofia, Bulgaria: STEF92 Technologies, 2015, pp. 3–10 [visited on 2022-02-06]. ISBN 43019948. ISSN 13142704. Available from DOI: 10.5593/SGEM2015/B21/S7.001. [SCOPUS and WOS indexed].

8. LICHEV, Lachezar; TOMECEK, J.; HENDRYCH, Jakub; CHEJKA, R.; KREJSA, M. NEW METHODS OF EVALUATION OF DEFORMATION STRUCTURE EXTRA-HIGH VOLTAGE PYLONS. In: *15th International Multidisciplinary Scientific GeoConference SGEM2015: Informatics, Geoinformatics and Remote Sensing* [online]. Sofia, Bulgaria: STEF92 Technologies, 2015, pp. 215–224 [visited on 2022-02-06]. ISBN 43019968. ISSN 13142704. Available from DOI: 10.5593/SGEM2015/B21/S7.028. [SCOPUS and WOS indexed].

9. LICHEV, Lachezar; HENDRYCH, Jakub; KUNCHICKY, Radim; FEBEROVÁ, Karolina; GOTSEVA, Daniela. A new method of defining objects of interest and 3D visualization in FOTOM-NG system. *International Scientific Conferense Computer Science'2015*. 2015, vol. 2015, no. 1, p. 6.

10. LIČEV, Lačezar; TOMEČEK, Jan; HENDRYCH, Jakub. Nová metoda definování zájmových objektů a 3D vizualizace v systému FOTOMNG. In: *Mezinárodní konference Geodesie a Důlní měřictví 2015, XXII. konference SDMG a Zasedání odborných komisí ISM*. Praha: -, 2015, p. 6. ISBN 978-20-248-3767-3. ISSN 978-20-248-3767-3.

11. LIČEV, Lačezar; TOMEČEK, Jan; HENDRYCH, Jakub; LIS, Dalibor; ČEJKA, Radim; KREJSA, Martin. Monitorování nadměrných deformací ocelové konstrukce stožáru velmi vysokého napětí. In: *Mezinárodní konference Geodesie a Důlní měřictví 2015, XXII. konference SDMG a Zasedání odborných komisí ISM*. Praha: -, 2015, p. 6. ISBN 978-20-248-3767-3. ISSN 978-20-248-3767-3.

12. HENDRYCH, Jakub; KUNČICKÝ, Radim. Analysis of Medical Images, Using Active Contours Optimized by Self-Organizing Migration Algorithm. In: *Proceedings of the Ph.D. Workshop of Faculty of Electrical Engineering and Computer Science, WOFEX*

2015. Ostrava: VŠB-TUO, 2015, pp. 295–300. ISBN 978-80-248-3787-1. ISSN 978-80-248-3787-1.

13. LICHEV, Lachezar; BABIUCH, Marek; HENDRYCH, Jakub; SKANDEROVÁ, Lenka. FOTOMNG SYSTEM AND 2D ANIMATION OF MEASUREMENT PROCESS. In: *13th International Multidisciplinary Scientific GeoConference SGEM2013: Informatics, Geoinformatics and Remote Sensing* [online]. Sofia, Bulgaria: STEF92 Technologies, 2013, pp. 115–122 [visited on 2022-02-06]. ISBN 44563966. ISSN 13142704. Available from DOI: 10.5593/SGEM2013/BB2.V1/S07.015. [SCOPUS and WOS indexed].

14. LIČEV, Lačezar; HENDRYCH, Jakub. 2D animace procesu měřen. In: *GIS Ostrava 2013 Geoinformatics for City Transformations - Symposium GIS Ostrava 2013*. Ostrava: VŠB-TUO, 2013, p. 7. ISBN 978-80-248-2558-8. ISSN 1213-239X.

15. LIČEV, Lačezar; HENDRYCH, Jakub. Systém FOTOMNG a 2D animace procesu měření. In: *Sborník referátů XIX. konference SDMG a IGDM*. 1st ed. Ostrava: VŠB-TUO, 2012, pp. 124–131. ISBN 978-80-248-2824-4. ISSN 978-80-248-2824-4.

List of Citations

These non-self citations are known:

- LIČEV, Lačezar; HENDRYCH, Jakub; TOMEČEK, Jan; ČAJKA, Radim; KREJSA, Martin. Monitoring of Excessive Deformation of Steel Structure Extra-High Voltage Pylons. *Periodica Polytechnica Civil Engineering* [online]. 2018, vol. 62, no. 2, pp. 323–329 [visited on 2022-02-06]. ISSN 05536626. Available from DOI: 10.3311/PPci.11253. [SCOPUS and WOS indexed; SJR 0,36]

    1. LI, Changping; MA, Xinteng; LIU, Yang. Reinforcement Design of Three -Valves Towers and Effect Evaluation Based on Monitoring Data. *IOP Conference Series: Earth and Environmental Science* [online]. 2021-04-01, vol. 719, no. 2, pp. - [visited on 2022-02-06]. ISSN 1755-1307. Available from DOI: 10.1088/1755-1315/719/2/022026
    2. JURASZEK, Janusz. Fiber Bragg Sensors on Strain Analysis of Power Transmission Lines. *Materials* [online]. 2020, vol. 13, no. 7, pp. - [visited on 2022-02-06]. ISSN 1996-1944. Available from DOI: 10.3390/ma13071559
    3. ZHAO, Long; HUANG, Xinbo; ZHANG, Ye; TIAN, Yi; ZHAO, Yu. A Vibration-Based Structural Health Monitoring System for Transmission Line Towers. *Electronics* [online]. 2019, vol. 8, no. 5, pp. - [visited on 2022-02-06]. ISSN 2079-9292. Available from DOI: 10.3390/electronics8050515
    4. WANG, Qi; ZHAO, Zhangyan. An Accurate and Stable Pose Estimation Method Based on Geometry for Port Hoisting Machinery. *IEEE Access* [online]. 2019, vol. 7, no. 1, pp. 39117–39128 [visited on 2022-02-06]. ISSN 2169-3536. Available from DOI: 10.1109/ACCESS.2019.2907222

- KROUPOVÁ, Ivana; LICHÝ, Petr; LIČEV, Lačezar; HENDRYCH, Jakub; SOUČEK, Kamil. Evaluation of properties of cast metal foams with irregular inner structure. *Archives of Metallurgy and Materials 63.* 2018, vol. 63, no. 4, pp. 1847–1851. ISSN 17333490. Available from DOI: 10.24425/amm.2018.125114. [SCOPUS and WOS indexed; SJR 0,254]

1. RADKOVSKÝ, F.; MERTA, V.; OBZINA, T. Design of proven technology of metal foam and porous metal casting production. *Archives of Foundry Engineering*. 2021, vol. 21, no. 1, pp. 125–131. ISSN 18973310. Available from DOI: 10.24425/afe.2021.136088
2. MERTA, V.; LÁNA, I. Manufacturing of cast-metal sponges from copper alloys. *Materiali in tehnologije* [online]. 2020-02-20, vol. 54, no. 1, pp. 117–119 [visited on 2022-02-06]. ISSN 15802949. Available from DOI: 10.17222/mit.2019.159
3. MERTA, V. Possibilities of cleaning of metal sponge castings. In: *METAL 2019 - 28th International Conference on Metallurgy and Materials, Conference Proceedings*. Brano: TANGER Ltd., 2019, pp. 1595–1599. ISBN 978-808729492-5. ISSN -

- HENDRYCH, Jakub; KUNČICKÝ, Radim; LIČEV, Lačezar. New Approach to Steganography Detection via Steganalysis Framework. In: *Proceedings of the Second International Scientific Conference "Intelligent Information Technologies for Industry" (IITI'17)* [online]. 679th ed. Cham: Springer International Publishing, 2018, pp. 496–503 [visited on 2022-02-06]. ISBN 978-3-319-68320-1. ISSN 21945357. Available from DOI: 10.1007/978-3-319-68321-8_51. [SCOPUS and WOS indexed]

  1. SHNIPEROV, A. N.; PROKOFIEVA, A. V. Steganalysis Method of Static JPEG Images Based on Artificial Immune System. *Automatic Control and Computer Sciences* [online]. 2020, vol. 54, no. 5, pp. 423–431 [visited on 2022-02-06]. ISSN 0146-4116. Available from DOI: 10.3103/S0146411620050077
  2. SHNIPEROV, Alexey Nikolaevich; PROKOFIEVA, Aleksandra Vladimirovna. Steganalysis method of static JPEG images based on artificial immune system. In: *Proceedings of the 12th International Conference on Security of Information and Networks - SIN '19* [online]. New York, New York, USA: ACM Press, 2019, pp. 1–7 [visited on 2022-02-06]. ISBN 9781450372428. ISSN -. Available from DOI: 10.1145/3357613.3357617

- LIČEV, Lačezar; HENDRYCH, Jakub; KUNČICKÝ, Radim; KOVÁŘOVÁ, Kateřina; KUMPOVÁ, Ivana. Evaluation of Sandstone Internal Structure with Application of Micro-CT and FOTOM System. In: *Proceedings of the Second International Scientific Conference "Intelligent Information Technologies for Industry" (IITI'17)* [online]. 680th ed. Cham: Springer International Publishing, 2018, pp. 332–339 [visited on 2022-02-06]. ISBN 978-3-319-68323-2. Available from DOI: 10.1007/978-3-319-68324-9_36. [SCOPUS and WOS indexed]

  1. YANG, Fayong; LI, Haibin; ZHAO, Guijuan; GUO, Ping; LI, Wenbo. Mechanical Performance and Durability Evaluation of Sandstone Concrete. *Advances in Mate-*

rials Science and Engineering [online]. 2020-09-01, vol. 2020, no. 1, pp. 1–10 [visited on 2022-02-06]. ISSN 1687-8434. Available from DOI: 10.1155/2020/2417496

- LICEV, Lacezar; FEBEROVA, Karolina; TOMECEK, Jan; HENDRYCH, Jakub. An Enhanced Method for Automatic Detection and Segmentation of Carotid Artery in Ultrasound Images. In: *Proceedings of the 17th International Conference on Computer Systems and Technologies 2016* [online]. New York, NY, USA: ACM, 2016-06-23, pp. 206–213 [visited on 2022-02-06]. ISBN 978-145034182-0. ISSN 9781450341820. Available from DOI: 10.1145/2983468.2983483. [SCOPUS and WOS indexed]

    1. ALZOUBI, Omar; AWAD, Mohammad Abu; ABDALLA, Ayman M. Automatic Segmentation and Detection System for Varicocele in Supine Position. *IEEE Access* [online]. 2021, vol. 9, no. 1, pp. 125393–125402 [visited on 2022-02-06]. ISSN 2169-3536. Available from DOI: 10.1109/ACCESS.2021.3111021

- LICHEV, Lachezar; TOMECEK, J.; HENDRYCH, Jakub; CHEJKA, R.; KREJSA, M. NEW METHODS OF EVALUATION OF DEFORMATION STRUCTURE EXTRA-HIGH VOLTAGE PYLONS. in: *15th International Multidisciplinary Scientific GeoConference SGEM2015: Informatics, Geoinformatics and Remote Sensing* [online]. Sofia, Bulgaria: STEF92 Technologies, 2015, pp. 215–224 [visited on 2022-02-06]. ISBN 43019968. ISSN 13142704. Available from DOI: 10.5593/SGEM2015/B21/S7.028. [SCOPUS and WOS indexed]

    1. JURASZEK, Janusz. Fiber Bragg Sensors on Strain Analysis of Power Transmission Lines. *Materials* [online]. 2020, vol. 13, no. 7, pp. - [visited on 2022-02-06]. ISSN 1996-1944. Available from DOI: 10.3390/ma13071559

List of Projects

During my doctoral studies, I participated in these projects:

1. The grant project Technology Agency of the Czech Republic - TACR Epsilon "OrthoSoft - Computer 3D Planning of operations in orthopedics and oncology orthopedics" - TH04010188 - Development of the web application for planning operations based on computed tomography images and an embedded SQL database of orthopedic implants.

2. The grant project Technology Agency of the Czech Republic - TACR "RODOS" - TE0102155 - Consultation in the field of software design. Responsibility for design for efficiently retrieving searched information in large-scale data sources and distributed data processing using HPC. Parallelization and implementation of partial program parts.

3. The grant project Technology Agency of the Czech Republic - TACR Delta "Security of Mobile Devices and Communication" - TF01000091 - Development of the application for steganography of static images for the purposes of mobile communication.

Appendix A

Test Results - Outguess2.0 and F5

Table A.1: Test results - OutGuess2.0, resolution 800 x 449, 50 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 318	FP = 13
Test outcome NEGATIVE (clear image)	FN = 2	TN = 307
Sensitivity [%]	**99,38**	
Specificity [%]	**95,94**	
Accuracy [%]	**97,66**	
Error [%]	**2,34**	

Table A.2: Test results - OutGuess2.0, resolution 800 x 449, 200 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 318	FP = 13
Test outcome NEGATIVE (clear image)	FN = 2	TN = 307
Sensitivity [%]	**99,38**	
Specificity [%]	**95,94**	
Accuracy [%]	**97,66**	
Error [%]	**2,34**	

Table A.3: Test results - OutGuess2.0, resolution 800 x 449, 500 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 318	FP = 13
Test outcome NEGATIVE (clear image)	FN = 2	TN = 307
Sensitivity [%]	**99,38**	
Specificity [%]	**95,94**	
Accuracy [%]	**97,66**	
Error [%]	**2,34**	

Table A.4: Test results - OutGuess2.0, resolution 800 x 449, 800 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 319	FP = 13
Test outcome NEGATIVE (clear image)	FN = 1	TN = 307
Sensitivity [%]	**99,69**	
Specificity [%]	**95,94**	
Accuracy [%]	**97,81**	
Error [%]	**2,19**	

Table A.5: Test results - OutGuess2.0, resolution 1024 x 575, 50 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 318	FP = 11
Test outcome NEGATIVE (clear image)	FN = 2	TN = 309
Sensitivity [%]	**99,38**	
Specificity [%]	**96,56**	
Accuracy [%]	**97,97**	
Error [%]	**2,03**	

Table A.6: Test results - OutGuess2.0, resolution 1024 x 575, 200 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 319	FP = 11
Test outcome NEGATIVE (clear image)	FN = 1	TN = 309
Sensitivity [%]	**99,69**	
Specificity [%]	**96,56**	
Accuracy [%]	**98,13**	
Error [%]	**1,88**	

Table A.7: Test results - OutGuess2.0, resolution 1024 x 575, 500 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 318	FP = 11
Test outcome NEGATIVE (clear image)	FN = 2	TN = 309
Sensitivity [%]	**99,38**	
Specificity [%]	**96,56**	
Accuracy [%]	**97,97**	
Error [%]	**2,03**	

Table A.8: Test results - OutGuess2.0, resolution 1024 x 575, 800 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 319	FP = 11
Test outcome NEGATIVE (clear image)	FN = 1	TN = 309
Sensitivity [%]	**99,69**	
Specificity [%]	**96,56**	
Accuracy [%]	**98,13**	
Error [%]	**1,88**	

Table A.9: Test results - OutGuess2.0, resolution 1440 x 809, 50 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 318	FP = 8
Test outcome NEGATIVE (clear image)	FN = 2	TN = 312
Sensitivity [%]	**99,38**	
Specificity [%]	**97,50**	
Accuracy [%]	**98,44**	
Error [%]	**1,56**	

Table A.10: Test results - OutGuess2.0, resolution 1440 x 809, 200 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 318	FP = 8
Test outcome NEGATIVE (clear image)	FN = 2	TN = 312
Sensitivity [%]	**99,38**	
Specificity [%]	**97,50**	
Accuracy [%]	**98,44**	
Error [%]	**1,56**	

Table A.11: Test results - OutGuess2.0, resolution 1440 x 809, 500 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 318	FP = 8
Test outcome NEGATIVE (clear image)	FN = 2	TN = 312
Sensitivity [%]	**99,38**	
Specificity [%]	**97,50**	
Accuracy [%]	**98,44**	
Error [%]	**1,56**	

Table A.12: Test results - OutGuess2.0, resolution 1440 x 809, 800 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 318	FP = 8
Test outcome NEGATIVE (clear image)	FN = 2	TN = 312
Sensitivity [%]	**99,38**	
Specificity [%]	**97,50**	
Accuracy [%]	**98,44**	
Error [%]	**1,56**	

Table A.13: Test results - OutGuess2.0, resolution 2560 x 1438, 50 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 315	FP = 10
Test outcome NEGATIVE (clear image)	FN = 5	TN = 310
Sensitivity [%]	**98,44**	
Specificity [%]	**96,88**	
Accuracy [%]	**97,66**	
Error [%]	**2,34**	

Table A.14: Test results - OutGuess2.0, resolution 2560 x 1438, 200 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 315	FP = 10
Test outcome NEGATIVE (clear image)	FN = 5	TN = 310
Sensitivity [%]	**98,44**	
Specificity [%]	**96,88**	
Accuracy [%]	**97,66**	
Error [%]	**2,34**	

Table A.15: Test results - OutGuess2.0, resolution 2560 x 1438, 500 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 315	FP = 10
Test outcome NEGATIVE (clear image)	FN = 5	TN = 310
Sensitivity [%]	**98,44**	
Specificity [%]	**96,88**	
Accuracy [%]	**97,66**	
Error [%]	**2,34**	

Table A.16: Test results - OutGuess2.0, resolution 2560 x 1438, 800 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 315	FP = 10
Test outcome NEGATIVE (clear image)	FN = 5	TN = 310
Sensitivity [%]	**98,44**	
Specificity [%]	**96,88**	
Accuracy [%]	**97,66**	
Error [%]	**2,34**	

Table A.17: Test results - OutGuess2.0, resolution 4200 x 2358, 50 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 311	FP = 15
Test outcome NEGATIVE (clear image)	FN = 9	TN = 305
Sensitivity [%]	**97,19**	
Specificity [%]	**95,31**	
Accuracy [%]	**96,25**	
Error [%]	**3,75**	

Table A.18: Test results - OutGuess2.0, resolution 4200 x 2358, 200 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 311	FP = 15
Test outcome NEGATIVE (clear image)	FN = 9	TN = 305
Sensitivity [%]	**97,19**	
Specificity [%]	**95,31**	
Accuracy [%]	**96,25**	
Error [%]	**3,75**	

Table A.19: Test results - OutGuess2.0, resolution 4200 x 2358, 500 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 311	FP = 15
Test outcome NEGATIVE (clear image)	FN = 9	TN = 305
Sensitivity [%]	**97,19**	
Specificity [%]	**95,31**	
Accuracy [%]	**96,25**	
Error [%]	**3,75**	

Table A.20: Test results - OutGuess2.0, resolution 4200 x 2358, 800 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 311	FP = 15
Test outcome NEGATIVE (clear image)	FN = 9	TN = 305
Sensitivity [%]	**97,19**	
Specificity [%]	**95,31**	
Accuracy [%]	**96,25**	
Error [%]	**3,75**	

Table A.21: Test results - F5, resolution 800 x 449, 50 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 299	FP = 13
Test outcome NEGATIVE (clear image)	FN = 21	TN = 307
Sensitivity [%]	**93,44**	
Specificity [%]	**95,94**	
Accuracy [%]	**94,69**	
Error [%]	**5,31**	

Table A.22: Test results - F5, resolution 800 x 449, 200 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 302	FP = 13
Test outcome NEGATIVE (clear image)	FN = 18	TN = 307
Sensitivity [%]	**94,38**	
Specificity [%]	**95,94**	
Accuracy [%]	**95,16**	
Error [%]	**4,84**	

Table A.23: Test results - F5, resolution 800 x 449, 500 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 303	FP = 13
Test outcome NEGATIVE (clear image)	FN = 17	TN = 307
Sensitivity [%]	**94,69**	
Specificity [%]	**95,94**	
Accuracy [%]	**95,31**	
Error [%]	**4,69**	

Table A.24: Test results - F5, resolution 800 x 449, 800 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 305	FP = 13
Test outcome NEGATIVE (clear image)	FN = 15	TN = 307
Sensitivity [%]	**95,31**	
Specificity [%]	**95,94**	
Accuracy [%]	**95,63**	
Error [%]	**4,38**	

Table A.25: Test results - F5, resolution 1024 x 575, 50 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 286	FP = 11
Test outcome NEGATIVE (clear image)	FN = 34	TN = 309
Sensitivity [%]	**89,38**	
Specificity [%]	**96,56**	
Accuracy [%]	**92,97**	
Error [%]	**7,03**	

Table A.26: Test results - F5, resolution 1024 x 575, 200 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 291	FP = 11
Test outcome NEGATIVE (clear image)	FN = 29	TN = 309
Sensitivity [%]	**90,94**	
Specificity [%]	**96,56**	
Accuracy [%]	**93,75**	
Error [%]	**6,25**	

Table A.27: Test results - F5, resolution 1024 x 575, 500 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 291	FP = 11
Test outcome NEGATIVE (clear image)	FN = 29	TN = 309
Sensitivity [%]	**90,94**	
Specificity [%]	**96,56**	
Accuracy [%]	**93,75**	
Error [%]	**6,25**	

Table A.28: Test results - F5, resolution 1024 x 575, 800 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 294	FP = 11
Test outcome NEGATIVE (clear image)	FN = 26	TN = 309
Sensitivity [%]	**91,88**	
Specificity [%]	**96,56**	
Accuracy [%]	**94,22**	
Error [%]	**5,78**	

Table A.29: Test results - F5, resolution 1440 x 809, 50 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 263	FP = 8
Test outcome NEGATIVE (clear image)	FN = 57	TN = 312
Sensitivity [%]	**82,19**	
Specificity [%]	**97,50**	
Accuracy [%]	**89,84**	
Error [%]	**10,16**	

Table A.30: Test results - F5, resolution 1440 x 809, 200 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 263	FP = 8
Test outcome NEGATIVE (clear image)	FN = 57	TN = 312
Sensitivity [%]	**82,19**	
Specificity [%]	**97,50**	
Accuracy [%]	**89,84**	
Error [%]	**10,16**	

Table A.31: Test results - F5, resolution 1440 x 809, 500 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 264	FP = 8
Test outcome NEGATIVE (clear image)	FN = 56	TN = 312
Sensitivity [%]	**82,50**	
Specificity [%]	**97,50**	
Accuracy [%]	**90,00**	
Error [%]	**10,00**	

Table A.32: Test results - F5, resolution 1440 x 809, 800 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 265	FP = 8
Test outcome NEGATIVE (clear image)	FN = 55	TN = 312
Sensitivity [%]	**82,81**	
Specificity [%]	**97,50**	
Accuracy [%]	**90,16**	
Error [%]	**9,84**	

Table A.33: Test results - F5, resolution 2560 x 1438, 50 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 286	FP = 10
Test outcome NEGATIVE (clear image)	FN = 34	TN = 310
Sensitivity [%]	**89,38**	
Specificity [%]	**96,88**	
Accuracy [%]	**93,13**	
Error [%]	**6,88**	

Table A.34: Test results - F5, resolution 2560 x 1438, 200 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 286	FP = 10
Test outcome NEGATIVE (clear image)	FN = 34	TN = 310
Sensitivity [%]	**89,38**	
Specificity [%]	**96,88**	
Accuracy [%]	**93,13**	
Error [%]	**6,88**	

Table A.35: Test results - F5, resolution 2560 x 1438, 500 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 286	FP = 10
Test outcome NEGATIVE (clear image)	FN = 34	TN = 310
Sensitivity [%]	**89,38**	
Specificity [%]	**96,88**	
Accuracy [%]	**93,13**	
Error [%]	**6,88**	

Table A.36: Test results - F5, resolution 2560 x 1438, 800 B secret message length.

	Condition - Images contain a secret message	
	Condition positive	*Condition negative*
Test outcome POSITIVE (stegogramme)	TP = 286	FP = 10
Test outcome NEGATIVE (clear image)	FN = 34	TN = 310
Sensitivity [%]	**89,38**	
Specificity [%]	**96,88**	
Accuracy [%]	**93,13**	
Error [%]	**6,88**	

End of Appendix A.

Appendix B

Test Results - Application of the Macroblock Filtering - OutGuess2.0 and F5

Table B.1: Application of the macroblock filtering - OutGuess2.0, resolution 800 x 449, 50 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	99,38	94,38	**5,00**
Specificity [%]	95,94	95,94	**0,00**
Accuracy [%]	97,66	95,16	**2,50**
Error [%]	2,34	4,84	**-2,50**

Table B.2: Application of the macroblock filtering - OutGuess2.0, resolution 800 x 449, 200 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	99,38	95,31	**4,06**
Specificity [%]	95,94	95,94	**0,00**
Accuracy [%]	97,66	95,63	**2,03**
Error [%]	2,34	4,38	**-2,03**

Table B.3: Application of the macroblock filtering - OutGuess2.0, resolution 800 x 449, 500 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	99,38	95,94	**3,44**
Specificity [%]	95,94	95,94	**0,00**
Accuracy [%]	97,66	95,94	**1,75**
Error [%]	2,34	4,06	**-1,75**

Table B.4: Application of the macroblock filtering - OutGuess2.0, resolution 800 x 449, 800 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	99,69	97,19	**2,50**
Specificity [%]	95,94	95,94	**0,00**
Accuracy [%]	97,81	96,56	**1,25**
Error [%]	2,19	3,44	**-1,25**

Table B.5: Application of the macroblock filtering - OutGuess2.0, resolution 1024 x 575, 50 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	99,38	95,31	**4,06**
Specificity [%]	96,86	96,56	**0,00**
Accuracy [%]	97,97	95,94	**2,03**
Error [%]	2,03	4,06	**-2,03**

Table B.6: Application of the macroblock filtering - OutGuess2.0, resolution 1024 x 575, 200 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	99,69	96,25	**3,44**
Specificity [%]	96,56	96,56	**0,00**
Accuracy [%]	98,13	96,41	**1,75**
Error [%]	1,88	3,56	**-1,75**

Table B.7: Application of the macroblock filtering - OutGuess2.0, resolution 1024 x 575, 500 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	99,38	96,25	**3,13**
Specificity [%]	96,56	96,56	**0,00**
Accuracy [%]	97,97	96,41	**1,56**
Error [%]	2,03	3,59	**-1,56**

Table B.8: Application of the macroblock filtering - OutGuess2.0, resolution 1024 x 575, 800 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	99,69	98,13	**1,56**
Specificity [%]	96,56	96,56	**0,00**
Accuracy [%]	98,13	97,34	**0,78**
Error [%]	1,88	2,66	**-0,78**

Table B.9: Application of the macroblock filtering - OutGuess2.0, resolution 1440 x 809, 50 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	99,38	94,69	**4,69**
Specificity [%]	97,50	98,13	*-0,63*
Accuracy [%]	98,44	96,41	**2,03**
Error [%]	1,56	3,59	**-2,03**

Table B.10: Application of the macroblock filtering - OutGuess2.0, resolution 1440 x 809, 200 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	99,38	94,69	**4,69**
Specificity [%]	97,50	98,13	*-0,63*
Accuracy [%]	98,44	96,41	**2,03**
Error [%]	1,56	3,59	**-2,03**

Table B.11: Application of the macroblock filtering - OutGuess2.0, resolution 1440 x 809, 500 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	99,38	95,63	**3,75**
Specificity [%]	97,50	98,13	*-0,63*
Accuracy [%]	98,44	96,88	**1,56**
Error [%]	1,56	3,13	**-1,56**

Table B.12: Application of the macroblock filtering - OutGuess2.0, resolution 1440 x 809, 800 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	99,38	95,63	**3,75**
Specificity [%]	97,50	98,13	*-0,63*
Accuracy [%]	98,44	96,88	**1,56**
Error [%]	1,56	3,13	**-1,56**

Table B.13: Application of the macroblock filtering - OutGuess2.0, resolution 2560 x 1438, 50 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	98,44	96,56	**1,88**
Specificity [%]	96,88	98,13	**1,25**
Accuracy [%]	97,66	97,34	**0,31**
Error [%]	2,34	2,66	**-0,31**

Table B.14: Application of the macroblock filtering - OutGuess2.0, resolution 2560 x 1438, 200 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	98,44	96,88	**1,56**
Specificity [%]	96,88	98,13	*-1,25*
Accuracy [%]	97,66	97,50	**0,16**
Error [%]	2,34	2,50	**-0,16**

Table B.15: Application of the macroblock filtering - OutGuess2.0, resolution 2560 x 1438, 500 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	98,44	96,88	**1,56**
Specificity [%]	96,88	98,13	*-1,25*
Accuracy [%]	97,66	97,50	**0,16**
Error [%]	2,34	2,50	**-0,16**

Table B.16: Application of the macroblock filtering - OutGuess2.0, resolution 2560 x 1438, 800 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	98,44	97,81	**0,63**
Specificity [%]	96,88	98,13	*-1,25*
Accuracy [%]	97,66	97,97	*-0,31*
Error [%]	2,34	2,03	*0,31*

Table B.17: Application of the macroblock filtering - OutGuess2.0, resolution 4200 x 2358, 50 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	97,19	78,13	**19,06**
Specificity [%]	95,31	100,00	*-4,69*
Accuracy [%]	96,25	89,06	**7,19**
Error [%]	3,75	10,94	**-7,19**

Table B.18: Application of the macroblock filtering - OutGuess2.0, resolution 4200 x 2358, 200 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	97,19	78,44	**18,75**
Specificity [%]	95,31	100,00	*-4,69*
Accuracy [%]	96,25	89,22	**7,03**
Error [%]	3,75	10,78	**-7,03**

Table B.19: Application of the macroblock filtering - OutGuess2.0, resolution 4200 x 2358, 500 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	97,19	78,44	**18,75**
Specificity [%]	95,31	100,00	*-4,69*
Accuracy [%]	96,25	89,22	**7,03**
Error [%]	3,75	10,78	**-7,03**

Table B.20: Application of the macroblock filtering - OutGuess2.0, resolution 4200 x 2358, 800 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	97,19	79,38	**17,81**
Specificity [%]	95,31	100,00	*-4,69*
Accuracy [%]	96,25	89,69	**6,56**
Error [%]	3,75	10,31	**-6,56**

Table B.21: Application of the macroblock filtering - F5, resolution 800 x 449, 50 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	93,44	82,19	**11,25**
Specificity [%]	95,94	95,94	**0,00**
Accuracy [%]	94,69	89,06	**5,63**
Error [%]	5,31	10,94	**-5,63**

Table B.22: Application of the macroblock filtering - F5, resolution 800 x 449, 200 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	94,38	83,13	**11,25**
Specificity [%]	95,94	95,94	**0,00**
Accuracy [%]	95,16	89,53	**5,63**
Error [%]	4,84	10,47	**-5,63**

Table B.23: Application of the macroblock filtering - F5, resolution 800 x 449, 500 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	94,69	86,25	**8,44**
Specificity [%]	95,94	95,94	**0,00**
Accuracy [%]	95,31	91,09	**4,22**
Error [%]	4,69	8,91	**-4,22**

Table B.24: Application of the macroblock filtering - F5, resolution 800 x 449, 800 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	95,31	86,88	**8,44**
Specificity [%]	95,94	95,94	**0,00**
Accuracy [%]	95,63	91,41	**4,22**
Error [%]	4,38	8,59	**-4,22**

Table B.25: Application of the macroblock filtering - F5, resolution 1024 x 575, 50 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	89,38	77,19	**12,19**
Specificity [%]	96,56	96,56	**0,00**
Accuracy [%]	92,97	86,88	**6,09**
Error [%]	7,03	13,13	**-6,09**

Table B.26: Application of the macroblock filtering - F5, resolution 1024 x 575, 200 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	90,94	79,38	**11,56**
Specificity [%]	96,56	96,56	**0,00**
Accuracy [%]	93,75	87,97	**5,78**
Error [%]	6,25	12,03	**-5,78**

Table B.27: Application of the macroblock filtering - F5, resolution 1024 x 575, 500 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	90,94	82,19	**8,75**
Specificity [%]	96,56	96,56	**0,00**
Accuracy [%]	93,75	89,38	**4,38**
Error [%]	6,25	10,63	**-4,38**

Table B.28: Application of the macroblock filtering - F5, resolution 1024 x 575, 800 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	91,88	83,13	**8,75**
Specificity [%]	96,56	96,56	**0,00**
Accuracy [%]	94,22	89,84	**4,38**
Error [%]	5,78	10,16	**-4,38**

Table B.29: Application of the macroblock filtering - F5, resolution 1440 x 809, 50 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	82,19	59,69	**22,50**
Specificity [%]	97,50	98,13	*-0,63*
Accuracy [%]	89,84	78,91	**10,94**
Error [%]	10,16	21,90	**-10,94**

Table B.30: Application of the macroblock filtering - F5, resolution 1440 x 809, 200 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	82,19	60,00	**22,19**
Specificity [%]	97,50	98,13	*-0,63*
Accuracy [%]	89,84	79,06	**10,78**
Error [%]	10,16	20,94	**-10,78**

Table B.31: Application of the macroblock filtering - F5, resolution 1440 x 809, 500 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	82,50	60,94	**21,56**
Specificity [%]	97,50	98,13	*-0,63*
Accuracy [%]	90,00	79,53	**10,47**
Error [%]	10,00	20,74	**-10,47**

Table B.32: Application of the macroblock filtering - F5, resolution 1440 x 809, 800 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	82,81	62,19	**20,63**
Specificity [%]	97,50	98,13	*-0,63*
Accuracy [%]	90,16	80,16	**10,00**
Error [%]	9,84	19,84	**-10,00**

Table B.33: Application of the macroblock filtering - F5, resolution 2560 x 1438, 50 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	89,38	80,63	**8,75**
Specificity [%]	96,88	98,13	*-1,25*
Accuracy [%]	93,13	89,38	**3,75**
Error [%]	6,88	10,63	**-3,75**

Table B.34: Application of the macroblock filtering - F5, resolution 2560 x 1438, 200 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	89,38	80,63	**8,75**
Specificity [%]	96,88	98,13	*-1,25*
Accuracy [%]	93,13	89,38	**3,75**
Error [%]	6,88	10,63	**-3,75**

Table B.35: Application of the macroblock filtering - F5, resolution 2560 x 1438, 500 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	89,38	81,25	**8,13**
Specificity [%]	96,88	98,13	*-1,25*
Accuracy [%]	93,13	89,69	**3,44**
Error [%]	6,88	10,31	**-3,44**

Table B.36: Application of the macroblock filtering - F5, resolution 2560 x 1438, 800 B secret message length.

	With the use of MF	**Without the use of MF**	*Difference*
Sensitivity [%]	89,38	81,25	**8,13**
Specificity [%]	96,88	98,13	*-1,25*
Accuracy [%]	93,13	89,69	**3,44**
Error [%]	6,88	10,31	**-3,44**

End of Appendix B.

Appendix C

Stegosaurus Software v1.0 - Screenshots

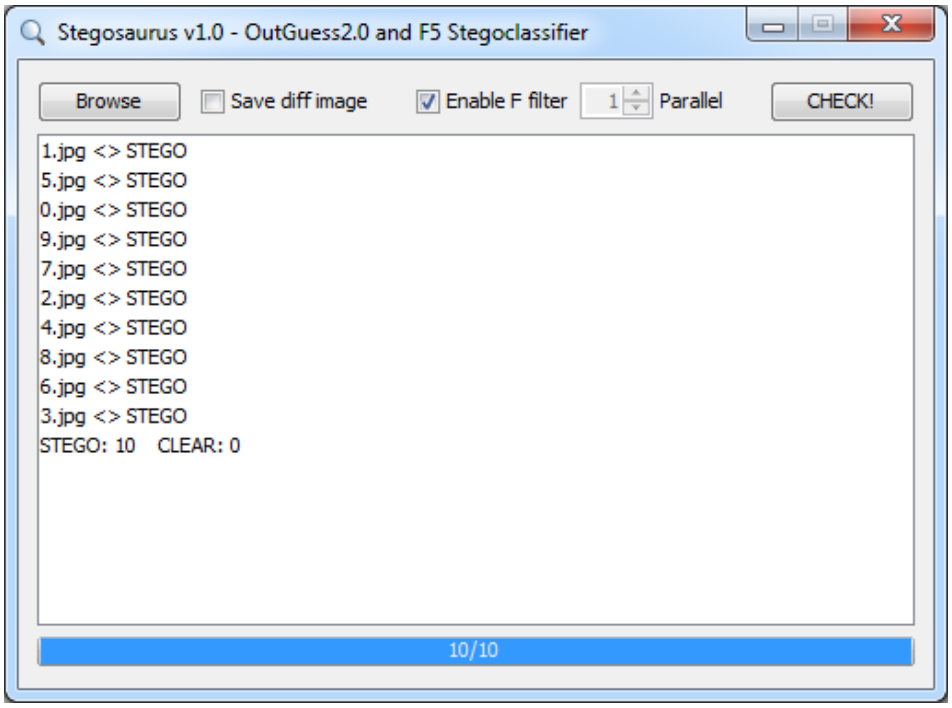


Figure C.1: Stegosaurus v1.0 - Main window.

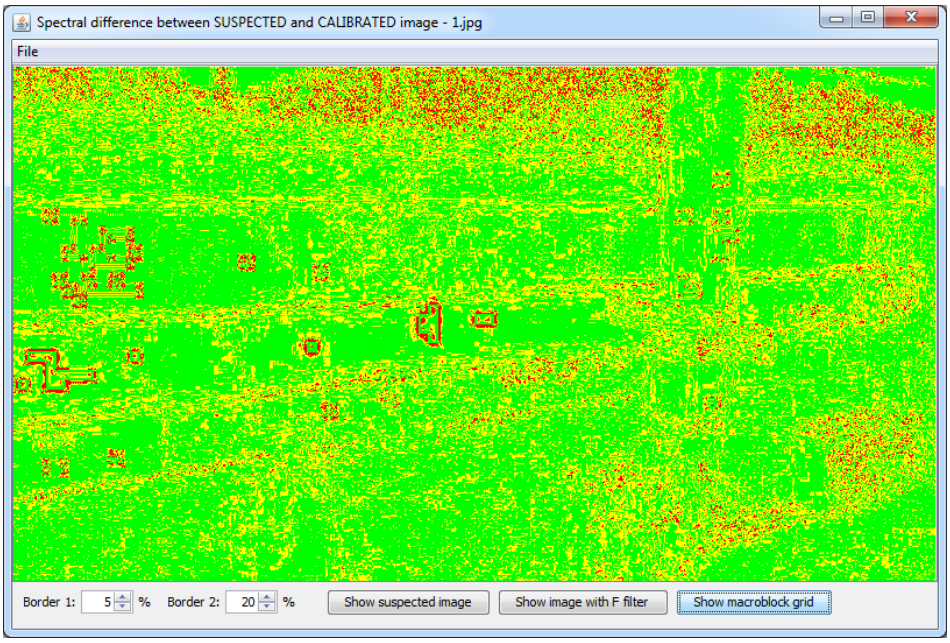


Figure C.2: Stegosaurus v1.0 - Spectral difference between suspected and calibrated image.

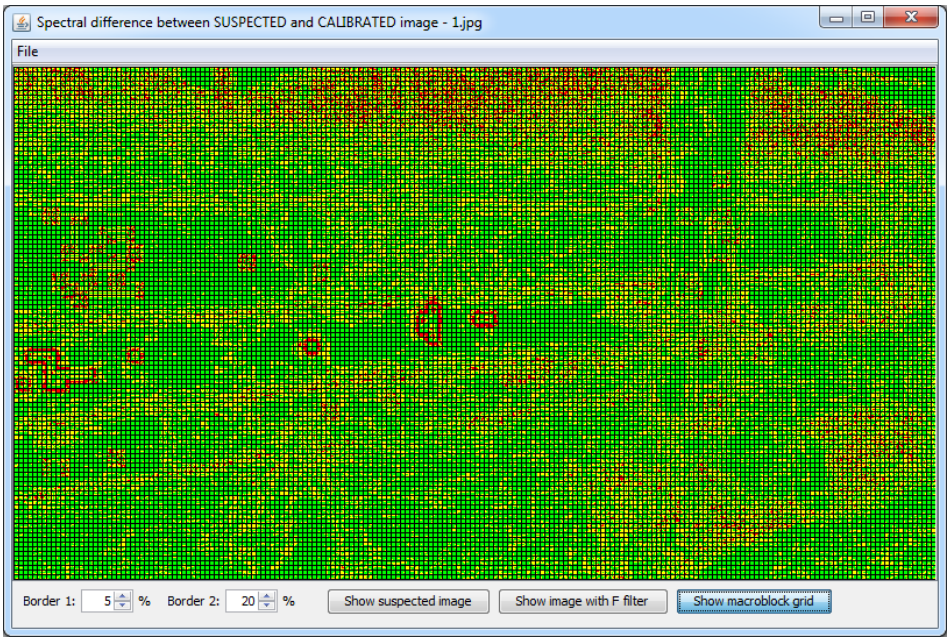


Figure C.3: Stegosaurus v1.0 - Spectral difference between suspected and calibrated image with macroblock grid.

End of Appendix C.

Quiz - image 1.3a is the stegogramme, image 1.3b is the clear image.
This text was created using LaTeX