Blind JPEG Steganalyzer

A project report

Dr. Jennifer Davidson
Department of Mathematics
Iowa State University, Ames, IA, USA

Contents

1 INTRODUCTION TO STEGANOGRAPHY AND STEGANALYSIS 1
1.1 JPEG Image Format . . . 2
1.2 JPEG Steganographic Algorithms . . . 3
1.3 Universal JPEG Steganalyzers . . . 4
1.3.1 Support Vector Machine . . . 5
1.3.2 Previous work on related Steganalyzers . . . 5

2 FEATURE SELECTION FOR STEGANALYSIS USING THE MAHALANOBIS DISTANCE 7
2.1 General Feature Reduction Techniques . . . 7
2.2 Previous Work on Feature Reduction in Steganalysis . . . 8
2.3 Mahalanobis distance . . . 9
2.4 Description of Experiments . . . 10
2.4.1 Experiments using UCID Database . . . 10
2.4.2 Experiments using BOWS2 Database for Merged . . . 16
2.5 Discussion of Results . . . 21

3 STEGANALYZER FEATURES DESIGNED ON STOCHASTIC MARKOV MODEL 25
3.1 POMMS for Steganalysis . . . 26
3.1.1 Intra block features . . . 27
3.1.2 Inter block features . . . 28
3.2 Experiments . . . 28
3.3 Discussion of results . . . 29
3.4 Combining other features with POMM . . . 36
3.4.1 Experiment-1 . . . 36
3.4.2 Experiment-2 . . . 36
3.4.3 Experiment-3 . . . 36
3.4.4 Experiment-4 . . . 37
3.4.5 Experiment-5 . . . 38
3.4.6 Experiment-6 . . . 39

4 Misc experiments 40
4.1 23 DCT features proposed in [26] . . . 40
4.2 Dual Histogram . . . 41
4.3 Local Histogram . . . 42
4.4 Probability density function moments and Characteristic function moments . . . 43
4.5 324 Markov based features proposed in [61] . . . 44
4.6 486 Markov based features proposed in [61] . . . 45
4.7 Extended DCT and Calibrated Markov feature proposed in [53] . . . 46
4.8 POMM2 - 432 features . . . 47

4.9 POMM3-243 features . . . 48
4.10 POMM4 - 270 features . . . 49
4.11 POMM 5-6 . . . 50
4.12 POMM 7 . . . 53
4.13 POMM8 . . . 55
4.14 Steganalyzer based on Joint Probability . . . 58
4.15 Accuracy results using Joint Probability with 3 neighbhorhood and Quantized DCT coefficient 59
4.16 Steganalyzer based on Conditional Probability from Partially Ordered Markov Model . . . 60
4.16.1 POMM on difference array of DCT coefficients . . . 62
4.16.2 Working with absolute values of differences . . . 63

5 CANVASS - A STEGANOGRAPHIC FORENSIC TOOL FOR JPEG IMAGES 64
5.1 Implementation Details . . . 65
5.2 Installer . . . 66
.1 Univariate Detection accuracy tables for Pevný Feature set [53] for BOWS2 Database . . . 68
.2 Multivariate Detection accuracy tables for Pevný Feature set [53] for BOWS2 Database . . . 70

List of Figures

1.1 JPEG compression steps 3

2.1 Mahalanobi distance graph for univariate and multivariate feature reduction steganalyzer proposed in [61] 11
2.2 Mahalanobi distance graph for univariate and multivariate feature reduction steganalyzer proposed in [53] 13
2.3 Mahalanobi distance graph for univariate and multivariate feature reduction Calibrated Markov Feature set 15
2.4 Mahalanobis distance vs. feature group size for five embedding algorithms using features in [53]. Solid line: multivariate distances. Dashed line: univariate distances. Note different scales on distance axes. 17
2.5 Detection accuracy results using univariate feature ranking for groups increasing by 10 features, cover images (solid line) and four embedding rates for five algorithms. Note different scales on distance axes. 22
2.6 Detection accuracy results using multivariate feature ranking for groups increasing by 10 features, for cover images (solid line) and four embedding rates for five algorithms. Note different scales on distance axes. 23

3.1 The adjacent lower neighborhood $adj_{\prec}(a_{i,j})$. The hash marks represent the location (i, j). . . 26
3.2 Partially Ordered Markov Model for steganalysis 27
3.3 Detection accuracy results for different steganalyzers on BOWS2 database 32
3.4 Detection accuracy results for different steganalyzers on Camera database 33
3.5 Detection accuracy results for different steganalyzers on Corel database 34
3.6 Detection accuracy results for different steganalyzers on NRCS database 35

5.1 Canvass - A software package for steganalysis 65
5.2 Canvass - Detailed Screen 66
5.3 (a)General and (b)Process Log file. (c) Sample report 67

ACKNOWLEDGEMENTS

This work was possible due to funding by the National Institute of Justice, through the Midwest Forensics Resource Center at Ames Laboratory under Interagency Agreement number 2008-DN-R-038. The Ames Laboratory is operated for the U.S. Department of Energy by Iowa State University, under contract No. DE-AC02-07CH11358. We would also like to thank Gerard Meyer for agreeing to beta test our software and provide non-monetary support in the goal of fulfilling the objective of this research. We would also like to thank Dr. Jessica Fridrich and Dr. Yun Q. Shi for providing code for the steganalyzers used in this research. Special thanks to Dr. Gwaneal Doerr for providing a rich database of JPEG image which helped in this research. Last, but not the least, we would like to acknowledge the support of John Dickerson and College of Engineering, Iowa State University for granting access to Matlab grid which saved a lot of computational time in our experiments.

Chapter 1

INTRODUCTION TO STEGANOGRAPHY AND STEGANALYSIS

Steganography is the practice of communicating a hidden message in such a way that no one, apart from the sender and intended recipient, suspects the existence of the message. The goal of steganography is to embed a *payload* into a *cover* object to obtain a *stego* object in such a way that the presence of hidden information cannot be detected by either perceptual or statistical analysis of the stego object. The counterpart of steganography is steganalysis. The main goal of steganalysis is to identify whether a given object has a payload embedded in it. Other information about the payload is often sought, including identification of the steganography algorithm, estimation of payload length, recovery of the payload, or obliteration of the payload. If there exists an algorithm that can determine whether or not a given image contains a secret message with a success rate better than random guessing, the steganographic scheme is considered to be broken. A more detailed introduction to steganography and steganalysis can be found in [10, 14, 21] .

Steganalysis can be broadly classified into two categories: *passive* or *active*. Passive steganalysis is concerned with detecting the presence or absence of hidden messages in a stego signal and identifying the stego embedding algorithm. On the other hand, active steganalysis addresses further issues such as estimating the embedded message length, locating the hidden message, finding the secret key used in embedding, identifying some parameters of the stego embedding algorithm, or extracting the hidden message itself, which is often the ultimate goal. Attacking steganography can also be classified as *targeted* and *blind* steganalysis. In targeted steganalysis, known embedding signatures, such as characteristic histogram shapes, are exploited to create specific feature values that can distinguish between stego and cover images. Blind steganalysis uses a set of generic feature values that model image statistics so as to distinguish between cover and stego images. Blind methods can be used on a variety of steganographic algorithms and do not target a specific algorithm. Recent advances in steganalysis allow for some blind detection algorithms to be almost as accurate as targeted detection algorithms.

With the advent of digital media and the Internet, multimedia objects such as still images and videos have become popular and are shared easily. Image and video data make a good choice for hiding payload. These objects are readily available and their broad presence on the Internet makes it difficult to check each one for hidden payload and thus difficult to detect the use of steganography. A single image can hold a reasonable amount of information, and a video file can hold more. In addition, there is a plethora of freeware available for hiding secret information, as can be seen by visiting the site stegoarchive.com [6]. MSU StegoVideo is a public video steganographic tool; see [7]. In this research, we restrict steganalysis of image data to Joint Photographic Experts Group (JPEG) format because of its wide use in consumer cameras and on the Internet. It also has the advantage of low bandwidth for storage and transmission, unlike raw or other uncompressed formats.

Our objective is to develop a passive blind steganalysis system which offers increased potential for classifying unknown embedding algorithms. Blind JPEG image steganalysis is generally addressed by representing an image with lower-dimensional features such as statistical properties, and then training a classifier on the feature set to differentiate between an innocent and stego image. To tackle this problem, we first analyze existing state-of-the-art steganalysis systems. In Chapter 2, we presents a filter-type feature selection algorithm that selects a reduced feature set using the Mahalanobis distance measure. It provides one solution to the problem of computational complexity, without loss of accuracy in detection. The experiment is applied to a well-known JPEG steganalyzer, and shows that using our approach, a reduced-feature steganalyzer can be obtained that performs as well as the original steganalyzer. This further helps us to understand which features are most useful for steganalysis. A manuscript describing the reduced feature steganalyzer has been accepted to the SPIE Media Forensics and Security XII, 2010 in San Jose, USA and Dr. Jennifer Davidson will be presenting the results at the conference.

In Chapter 3, we propose a new modeling technique by developing a Partially Ordered Markov Model (POMM) to characterize JPEG images. Our rationale to do so is based on our experimental observation from Chapter 2, which shows that the Markov based features used by [53] contribute significantly to the detection process. A POMM generalizes the concept of local neighborhood directionality by using a partial order underlying the pixel locations. It has been shown that this property results in a computational advantage of POMMS over Markov Random Fields (MRF) in two ways [40]: whenever the normalizing constant needs to be calculated, such as in determining the joint probability distribution function (pdf), the joint pdf of a POMM is available in closed form; and the normalizing constant for a POMM is always known and equal to the value one [23]. We also show that our steganalyzer can beat state-of-the-art on four different databases, namely BOWS2 [66], Camera, Corel and NRCS [24] with more than 20,000 images.

In Chapter 4, we present Canvass, a software package that has been developed in Java to make this research accessible to the Iowa Department of Criminal Investigation forensic lab.

The rest of the chapter is as follows. We discuss the nature of JPEG images since we restrict steganalysis to JPEG images, and provide necessary details to the reader concerning the JPEG format. We then introduce a simple case of steganography in JPEG images followed by a popular steganographic algorithm, which we use later to test our proposed steganalyzer. We end this chapter by a small discussion on state-of-the art steganalyzers, some of which are used for comparison with the proposed steganalyzer.

1.1 JPEG Image Format

A digital image can be viewed as a spatial, multivariate array of pixels (picture elements). Suppose a generic pixel location is written as s, a vector in $\mathbb{R}^2$. The quantity $g(s)$ denotes the pixel value, such as the intensity of radiation in a band of the electromagnetic spectrum, at pixel location s. Then we write an image as

$$g \equiv \{g(s) : s \in D\}$$

where D is the index set of pixel locations. This set is typically finite with regular spacing, so that, without loss of generality, we assume

$$D = \{(x, y) : x = 1, ..., M; y = 1, ..., N\}$$

Thus, an image is an MxN rectangular array of pixel values. In an 8 bit image, $g(s) \in \{0, 1, ..., 255\}$. The Joint Photographic Experts Group (JPEG) format stores image data in a lossy compressed state as quantized frequency coefficients. Image files that employ JPEG compression are commonly called "JPEG files". Figure 1.1 shows the compressing steps performed. The JPEG compressor first partitions the uncompressed image into sets of 8 by 8 pixels. The discrete cosine transformation (DCT) transforms the 8x8 spatial brightness values into 8x8 frequency coefficients (real numbers) as follows:

$$G_{u,v} = \alpha(u)\alpha(v) \sum_{x=0}^{7} \sum_{y=0}^{7} g_{x,y} cos \left[\frac{\pi}{8} \left(x + \frac{1}{2} \right) u \right] cos \left[\frac{\pi}{8} \left(y + \frac{1}{2} \right) v \right]$$

where u is the horizontal spatial frequency, for the integers $0 \leq u \leq 7$; v is the vertical spatial frequency, for the integers $0 \leq v \leq 7$; $g_{x,y}$ is the spatial pixel value at coordinates (x, y); $G_{u,v}$ is the DCT coefficient at coordinates (u, v) and α is a normalizing function given as

$$\alpha(n) = \begin{cases} \sqrt{\frac{1}{8}} & \text{if } n = 0, \\ \sqrt{\frac{2}{8}} & \text{otherwise.} \end{cases}$$

After applying the DCT, a quality matrix Q is used to quantize the frequency coefficients to integers in the range -1024 to 1023. This step loses information. The quantized DCT coefficients are computed as

$$B_{j,k} = \text{round}\left(\frac{G_{j,k}}{Q_{j,k}}\right)$$

where $0 \leq j, k \leq 7$ and B is the set of quantized DCT coefficients. After lossy quantization, the Huffman coding ensures the lossless coding of the quantized coefficients. A more detailed description of the JPEG compression can be found in [11].

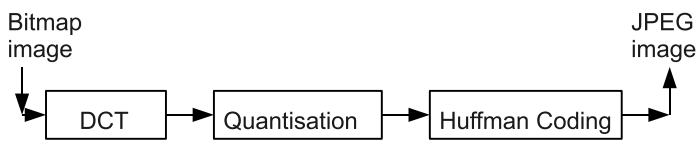


Figure 1.1: JPEG compression steps

Note that the correlations among DCT coefficients can be exploited in two following ways:

- **Intra block correlations**: It capture correlations that occur with each 8x8 block of quantized DCT coefficients.
- **Inter block correlations**: It capture correlations that occur between DCT coefficients in different blocks but the same (or close to the same) relative positions.

1.2 JPEG Steganographic Algorithms

A steganography algorithm embeds a payload into a cover object in such a way that the presence of hidden information cannot be detected by either perceptual or statistical analysis of the stego object. A payload consists of a vector of uniformly randomly generated bits 0 and 1, representing an encrypted bitstream. A JPEG steganography algorithm embeds payload by changing the quantized DCT coefficient values B. The most popular, frequently used and easy to implement steganographic method is the Least Significant Bit (LSB) steganography. The LSB steganographic methods can be classified into the following two categories: LSB replacement; and LSB matching, also called plus/minus one embedding [49]. The LSB replacement method works by replacing the least significant bit of a DCT coefficient value with the payload bit. In LSB matching, a coefficient value is modified as needed by increasing or decreasing the base 10 coefficients randomly to match the payload bit. If the bit must change, a value of +1 or -1 is added randomly to make the DCT bit value match that of the payload's. This seemingly innocent modification of the LSB embedding is significantly harder to detect, because changed pixel values are no longer paired as in LSB replacement.

Jsteg [4] is probably the first steganographic tool to embed in JPEG images. It was developed by Derek Upham in 1993. This hiding algorithm embeds a payload bit by replacing the LSB of the quantized DCT coefficients and skipping all those coefficients whose value is 0 or 1. It can embed data roughly 12% of file size of cover image, and embedding is performed in a sequential order on the coefficients.

OutGuess [54] was developed by Neil Provos in 2001. The algorithm identifies redundant coefficients that have the least effect on the cover image and modifies them if necessary during embedding of message bit using LSB replacement. For each bit changed due to payload embedding, the algorithm changes another untouched coefficient. The original global histogram of JPEG coefficients is preserved after embedding is complete. The algorithm voluntarily limits the maximum size of data that can be embedded to 6% of file size of cover image in order to make the algorithm more robust to statistical analysis.

The F5 [67] steganographic algorithm was introduced by Pfitzmann and Westfeld in 2001. It was developed from Jsteg in an iterative fashion with each new version being less detectable. It embeds message bits into randomly chosen DCT coefficients. If the payload is small enough, it uses "matrix embedding", an approach based on coding theory that minimizes the necessary number of changes to embed a message of certain length. The message length and the number of non-zero non-DC coefficients are used to determine the best matrix embedding that minimizes the number of modifications of the cover image. For payload too large for matrix embedding, it decrements the absolute value of DCT coefficients if LSB of the DCT coefficient and payload bit does not match. The author claims that its maximum steganographic content size is roughly 13% of size of cover image.

Steghide [2] uses a graph-theoretic approach to steganography. At first, the secret message is compressed and encrypted into the payload bitstream having n bits. Then a random site visitation sequence is generated in the DCT domain. The n quantized DCT coefficient values (cover values) from those locations are listed, and paired with payload bits. The pixel locations (u_i, v_i) $1 \leq i \leq n$ for each cover value are retained along with the coefficient value. For each payload bit i, $1 \leq i \leq n$, the corresponding cover value bit is inspected. If the payload bit value matches the corresponding cover value bit, then nothing is done. If payload bit i does not match the corresponding cover bit, then a graph-theoretic search algorithm is used to find a different cover value, say at position j in the original list of DCT values, whose bit matches the payload bit. The DCT values at the two sites are switched so that the DCT value at position i with pixel location (u_i, v_i) is now at location j (u_j, v_j) and vice versa, and the bit values of the newly located cover values now match with the corresponding payload bits at each location i and j in the payload list. This is done for all such possible pairs until all the payload bits match with switched cover bits, or until it is not possible to make any more exchanges. If there are any remaining payload bits not embedded, that is, not having bits that correspond to the cover value bits in the list, then those remaining cover value bits are used to embed those payload bits by LSB replacement (overwritten).

JPHide [3] uses DCT coefficients based on a fixed table such that coefficients with higher numerical value are selected first. All coefficients in the current class are used first to hide the information before the next class is chosed. The data hiding process continues in the current coefficient class even after the complete message has been hidden. A pseudo-random number generator determines if coefficients are skipped. The probability of skipping bits depends on the length of the hidden message and how many bits have been embedded already. An interesting property of JPHide is that it not only modifies the least-significant bits of the DCT coefficients, it can also switch to a mode where the second-least-significant bits are modified.

1.3 Universal JPEG Steganalyzers

Steganalysis can be considered as a two-class pattern classification problem if the test image needs to be classified as either a cover image or a stego image. Generally, the classification consists of two parts, feature extraction and pattern classification. Since image data is typically very large, a lower-dimensional representation of the information in the image, relative to the classification task at hand, is required. A feature is such a lower-dimensional representation of the image data and is crucial for many classification problems, including steganalysis. The best features for steganalysis should contain information about the changes incurred by data hiding rather than by the content of the image.

Calibration is a well known technique in steganalysis which improves detection accuracy by making features dependent on the changes incurred by data hiding rather on the image content itself. This was first introduced by Fridrich in [26]. Calibration is a process used to estimate macroscopic properties of the cover image from the stego image. During calibration, the stego JPEG image I^s is decompressed to the spatial

domain, cropped by few pixels in both directions, and compressed again with the same quality matrix as the stego image. It is assumed that the newly obtained JPEG image has most macroscopic features similar to the original cover image. This is because the cropped image is visually similar to the original image. Moreover, the cropping brings the 8x8 DCT grid "out of sync" with the previous compression, which effectively suppresses the influence of previous JPEG compression and the embedding changes. We will see how this technique can be used later in Chapter 3. More information on calibration can be found in [26, 51, 52, 41].

1.3.1 Support Vector Machine

Once the feature set is fixed, the detection performance will vary based on the pattern classifier used and the actual feature values. The support vector machines [13] (SVM) are very powerful for two-class classification. Given a training set of instance-label pairs $(\mathbf{x}_i, y_i), i = 1, ..., l$ where $\mathbf{x}_i \in \mathbb{R}^n$ are the feature vectors and $y_i \in \{1, -1\}$ are the two classes, the SVM requires the solution of the following optimization problem:

$$\begin{aligned} \max \ & \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \text{subject to } & 0 \leq \alpha_i \leq C, \\ & \sum_i \alpha_i y_i = 0. \end{aligned}$$

Here training vectors $\mathbf{x}_i$ are mapped into a higher (perhaps infinite) dimensional space by the function ϕ. Then the SVM finds a linear separating hyperplane with the maximal margin in this higher dimensional space. $C > 0$ is the penalty parameter of the error term. Furthermore, $K(\mathbf{x}_i, \mathbf{x}_j) \equiv \phi(\mathbf{x}_i).\phi(\mathbf{x}_j)$ is called the kernel function. Once a SVM has been trained, it can be used to determine the class of an unknown sample $\mathbf{x}$ by computing the sign of

$$f(\mathbf{x}) = \sum_{i=1}^{N_s} \alpha_i y_i K(\mathbf{s}_i, \mathbf{x}) + b$$

where $\mathbf{s}_i$ are training instances $\mathbf{x}_i$ (support vectors) with $\alpha_i > 0$ and N_s is the number of support vectors. In our experiments, we use a SVM with radial basis function (RBF): $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma ||\mathbf{x}_i - \mathbf{x}_j||^2)$, $\gamma > 0$. For a more comprehensive introduction to SVMs, please see [13].

1.3.2 Previous work on related Steganalyzers

In one of the earliest papers to appear on blind steganalysis, Avcibas et al. [12] use image quality measures (IQM) to detect watermarks. ANOVA is used to pick IQM measures that have greater discriminating power between watermarked and non-watermarked images in the spatial domain.

A work by Fridrich [26] has clearly been a fundamental contribution to blind JPEG detection. Using a linear classifier, 23 DCT-based features and a training set of 1600 images, accuracies for correct stego classification obtained were state-of-the-art at the time. A more recent work of Farid's [45] describes the use of various Support Vector Machine (SVMs) and 432+216 = 648 features trained on 32000 images, with five JPEG embedding algorithms. The authors performed a forward feature selection search to select individual features ranked by classification accuracy, and constructed (linear) SVMs for the sets of selected features. Following Fridrich's and Farid's works, Shi et al. [61] introduced a JPEG steganalyzer using Markov transition matrices calculated on four directional differences in neighboring values in the DCT coefficients. Their steganalyzer used 324 feature values, a polynomial kernel (nonlinear) SVM classifier, and approximately 7500 training images. Their results show an improvement over both Farid's and Fridrich's features to distinguish between stego and cover image data formatted in jpeg, trained on 800 images. Shi's work presented an investigation into the use of each of the four directions individually and how each direction performs in detection. They concluded that the combination of the four sets of features performs better than

each set individually, and did not reduce the number of features in the model. In Pevný et al.'s more recent work [53], the authors combine the four Markov feature sets into one set by averaging the four values at a location, producing 81 instead of 324 feature values, a reduction in the number of Markov feature values. They add the 81 to an extended set of 193 DCT features, producing a total feature set of 274 values. This set of feature values produces an improvement in classification accuracy over Shi's Markov model. Later on in 2008, Chen and Shi[18] extended the steganalyzer in [61] to 486 features by computing transition probability matrices for each difference JPEG 2-D array to utilize the intrablock correlation, and "averaged" transition probability matrices for those difference mode 2-D arrays to utilize interblock correlation. Although this steganalyzer is an improvement over preceding ones, we believe that a better accuracy rate can be achieved.

Chapter 2

FEATURE SELECTION FOR STEGANALYSIS USING THE MAHALANOBIS DISTANCE

As mentioned in Section 1.3, blind JPEG image steganalysis is generally addressed by representing an image with lower-dimensional features such as statistical properties, and then training a classifier on the feature set to differentiate cover and stego image. Many successful steganalysis algorithms use a large number of features relative to the size of the training set and suffer from a "curse of dimensionality": large number of feature values relative to training data size. High dimensionality of the feature space can reduce classification accuracy, obscure important features for classification, and increase computational complexity. This chapter presents a filter-type feature selection algorithm that selects reduced feature sets using the Mahalanobis distance measure, and develops classifiers from the sets.

We show that it is possible to build a classifier having many fewer features than the full suite of features, and that these lower-dimensional classifiers have performance results very comparable to that of the classifier with all features. This also gives an indication about the features which are better for the purpose of steganalysis.

2.1 General Feature Reduction Techniques

Computationally intensive, analysis for steganography content requires many hours and even days of processing image data to develop a reliable classifier. If a smaller subset of the features can be selected to represent the *intrinsic dimensionality* [34] of the data at its full dimension, then it may be possible to develop a classifier that is as effective as the one with all features, with less computation time and complexity.

There are two basic approaches to feature reduction or feature selection: filter approaches and wrapper approaches. A filter method provides a reduced feature set by ranking the feature values independently of the classifier, while a wrapper method uses a classifier to assess the effectiveness of the feature subset chosen. As such, wrapper methods are typically dependent on the type of classifier used, such as artificial neural network or support vector machine, and a classifier is developed with each choice of subset chosen. Filter methods tend to be much quicker than wrapper methods. A disadvantage of filter methods is that because of their independence from the classifier, it is not known how the selected subset will perform until a classifier is built. However, with fewer features it may be possible to develop several classifiers from which a satisfactory one can be picked.

Feature selection methods can be divided further into two categories, univariate and multivariate. Univariate methods examine features individually to determine discriminating power, while multivariate methods examine groups of feature values. Multivariate methods allow correlations and dependencies between features

to be accounted for, while univariate methods do not. Univariate methods applied to steganalysis include the Bhattacharyya distance [64, 63, 69, 44]. There appear to be no attempts to use the Bhattacharyya distance in a multivariate environment for steganalysis as of this time. Multivariate feature selection methods used in steganalysis include principal components analysis (PCA) [58, 43], although there are known limitations to PCA. The Mahalanobis distance measure along with an empirically determined threshold value was used in [30] for classification of feature values (Center of Mass, COM), and not for feature reduction.

The availability of statistical information from the data guides the selection of distance measure. The Bhattacharyya distance measure can be used if the probability density functions for the classes are known. Let $p_0(x)$ be the probability distribution for one class and $p_1(x)$ be the probability distribution for the second class. Then the Bhattacharyya distance between the two classes is defined as

$$B(p_0, p_1) = -log \int \sqrt{(p_0(x)p_1(x)}dx. \tag{2.1}$$

In theory, the Bhattacharyya distance gives upper and lower bounds of the Bayes error [25], although in practice the bounds are often not tight enough. While this univariate filter-approach is straightforward to compute for image feature data, it does not take into consideration other information, such as mean or variance values.

PCA can be used for feature reduction. It provides the optimal solution to the linear projection problem of feature reduction when using the least mean square measure of error. The general goal of using PCA is to provide a representation of the data that is de-correlated, where second-order dependencies are removed. It can also be used to reduce the number of feature variables necessary for classification. PCA has been useful in reducing the thousands of variables in microarray data [50] to a manageable amount. PCA ranks features according to the variance of features in a transformed and uncorrelated feature space described by the singular value decomposition [62]. If the data has higher order dependencies or nonlinear dependencies, then PCA will be insufficient in revealing all the structure in the data. Because PCA simply ranks the feature variables according to the variance of the data, PCA may or may not provide an adequate clustering of the data classes. Selecting features according to largest eigenvalue ranking given by principal component analysis can be suboptimal [17, 15], including for steganalysis [43, 69].

If estimates of the population mean values, variances, and covariance matrix are available from the data as is often the case in steganalysis, then the Mahalanobis distance can be used. The Mahalanobis distance between two classes can be written as

$$D^2 = (\mu^s - \mu^c)' V^{-1} (\mu^s - \mu^c) \tag{2.2}$$

here expressed for two classes s = stego and c = cover. Here, μ^s (μ^c) represents the mean vector for the stego (cover) population, prime denotes matrix transpose, and V is the covariance matrix.

The Mahalanobis distance has been used in other areas of pattern recognition for several decades, including species identification in zoology [59], diagnostic validity in neurology [35], and many other fields. It is used to provide a measure of similarity between multivariate populations and uses covariance information between variables to weight the contributions to the distance. The Euclidean distance, on the other hand, in essence gives excess weight to variables that are highly correlated and gives additional weight to variables that have similar information. The Mahalanobis distance gives less weight to those variables that have high variance and to those variables that have high correlation, so that other feature variables with lower correlations can contribute to the distance. When correlations between variables are known to exist, the Mahalanobis distance can offer an advantage for clustering. That is the case for the feature sets under consideration in this chapter.

2.2 Previous Work on Feature Reduction in Steganalysis

While the work described in Section 1.3.2 shows JPEG steganalysis on how individual features classify (as in [26]) or linear classification of selected subsets of features (as in [45]), only a few other papers present

systematic methods for reducing features in JPEG steganalysis and creating comparable classifiers. A filter approach using the Bhattacharyya distance for steganalysis feature selection was investigated in [69]; however, the application to steganalysis was very brief, comparing the detection rate on one spatial domain LSB embedding algorithm only. In [43], the authors describe three wrapper methods for grayscale steganalysis. In [39], the authors investigate the use of PCA for total variance of the data for spatial domain embedding, and perform forward and backward feature selection on the 27 features for several databases, comparing detection accuracies. In [48] , the authors describe a feature selection method for steganalysis on JPEG images using a K Nearest Neighbor approach, which takes some time to process as it is itself a classifier. In [47], the authors use a wrapper-type method to reduce the number of features for JPEG steganalysis. Authors in several papers have chosen fewer feature values so as to make their number of features manageable in the course of their feature design [19, 33].

With the exception of the grayscale steganalysis research in [32, 43, 39, 45], and the JPEG steganalysis in [48, 47], none of these ten previous works perform any in-depth investigation into the selection of fewer features for the express purpose of overcoming the dimensionality limitations. That is the goal of our work in this chapter. Our work is the first (to our knowledge) use of the Mahalanobis measure for feature selection in steganalysis. We show how the Mahalanobis distance can be used to provide a good ranking of the features and maintain accuracy of detection relative to the original full suite of features. The benefits of using a ranking of feature variables that maintain the ability to classify well with smaller numbers of features include reduced complexity in both feature dimension and training computation time. This can lead to improved ability of the classifier to generalize its solution to unseen data. We select support vector machines for the classifiers, as they are known to offer the potential for generalizable solutions in high-dimensional feature spaces [34].

2.3 Mahalanobis distance

We now describe the Mahalanobis distance measure. Let X_i be a random variable (r.v.) representing feature i, and let x^c_{ij} (x^s_{ij}) denote a sample from cover (stego) image i of feature value j. Denote by $\bar{x}^c_i$ ($\bar{x}^s_i$) the sample mean for feature i for the cover (stego) class, and by $v^c_{jk} = \frac{1}{N-1}\sum_{i=1}^{N}(x^c_{ij} - \bar{x}^c_j)(x^c_{ik} - \bar{x}^c_k)$ the sample covariance between feature j and feature k. The sample covariance values v^s_{jk} are similarly defined for stego images. Let $V^c = (v^c_{jk})$ denote the cover sample covariance matrix, and $V^s = (v^s_{jk})$ denote the stego sample covariance matrix. Let p denote the number of features and let N denote the number of training images. In our particular case, $N = 5000$ each for cover and stego, and $p = 274$. After calculating the sample covariance matrices, we use the pooled covariance matrix to calculate the Mahalanobis distance [46]:

$$V = \frac{N}{2N-2}V^c + \frac{N}{2N-2}V^s = \frac{N}{2N-2}(V^c + V^s). \tag{2.3}$$

For a set of $k \leq p$ features, the Mahalanobis distance $D^2_{(k)}$, in quadratic form, between the centroids of those features is given by:

$$D^2_{(k)} = (\mu^c - \mu^s)'V^{-1}(\mu^c - \mu^s) \tag{2.4}$$

where $\mu^c = (\bar{x}^c_1, \bar{x}^c_2, ..., \bar{x}^c_k)'$ and $\mu^s = (\bar{x}^s_1, \bar{x}^s_2, ..., \bar{x}^s_k)'$ represent the vector of sample mean values for the cover and stego images, respectively, and V^{-1} is the inverse of the pooled covariance matrix for the k features. In practice, V may be only positive semi-definite (that is, not full rank) or ill-conditioned. The problem of inverting V may be regularized by offsetting the diagonal of V and thus creating a positive definite matrix out of V in the following way. Let $\epsilon > 0$ be a number that is significantly smaller than the average variance of the class clusters, and write $V = QDQ'$ as the diagonalization of V by orthogonal matrix Q. The values of $D = diag(\lambda_1, ..., \lambda_n)$ are the eigenvalues for V. By adding ϵI to the matrix V, where I is the identity matrix, we can solve the problem of inverting V effectively:

$$V + \epsilon I = QDQ' + \epsilon I = QDQ' + Q(\epsilon I)Q' = Q(D + \epsilon I)Q' \tag{2.5}$$

Since the eigenvalues in $V+\epsilon I$ are now strictly positive and ϵ was chosen "large enough" to avoid ill-posedness, we let $\hat{V} = V + \epsilon I$ replace the covariance matrix V in the calculation of the Mahalanobis distance, Eq. 2.4. Our value for ϵ is the minimum of the diagonal entries of the pooled covariance matrix V, divided by 10 000.

Note that the Mahalanobis distance gives less weight for features having larger variance and more weight for features having smaller variance. This can be seen by looking at the one dimensional case (only one feature), where the Mahalanobis distance is

$$D_j^2 = \frac{(\bar{x}_j^c - \bar{x}_j^s)^2}{\sigma_j^2}, j = 1, ..., p \tag{2.6}$$

where $\bar{x}_j^c$ ($\bar{x}_j^s$) are the respective sample means for the j-th cover (stego) feature, and $\sigma_j^2 = \frac{N}{2N-2}[\sum_{i=1}^N (x_{ij}^c - \bar{x}_j^c)^2 + \sum_{i=1}^N (x_{ij}^s - \bar{x}_j^s)^2]$ is the common variance. In higher dimensions with more than one feature, $\frac{1}{\sigma^2}$ is replaced by the inverse of the (regularized) covariance matrix. A larger Mahalanobis distance value indicates more class separation when the Mahalanobis distance is used, with a threshold, as a linear classifier.

The Mahalanobis distance can be expressed in terms of principal components, that is, in terms of the eigenvectors and eigenvalues that are used in principal component analysis. Following a result by [17], the relation between eigenvalues of the covariance matrix for the feature data and values for the Mahalanobis distance measure becomes clear. The PCA in this case is performed on the feature data matrix Y consisting of all mean-centered cover and stego feature vectors vertically concatenated and having dimension $2N$x p. Let W be the p dimensional covariance matrix for Y: $W = Y'Y$. Let D^2 denote the Mahalanobis distance between the two classes $c = cover$ and $s = stego$ where c takes proportion q and s takes proportion $1 - q$. Then for $d = \mu^c - \mu^s$, the respective mean vectors of length p for the two feature classes, and V in Eq. 2.3, it can be shown that

$$W = q(1-q)dd' + V \tag{2.7}$$

$$D^2 = d'V^{-1}d \tag{2.8}$$

Let $P_1, P_2, ..., P_p$ (the columns of P) denote the p eigenvectors of W, with eigenvalues $\gamma_1, \gamma_2, ..., \gamma_p$ respectively. Recall the spectral representation of W: $W = \sum_{i=1}^p \gamma_i P_i P_i'$. For given $k \leq p$, let $B_k = (P_1, P_2, ..., P_k)$ be a basis and denote $D_{(k)}^2$ as the distance between the clusters using B_k in place of W. Here it is not assumed that $\gamma_1 \geq \gamma_2 \geq .... \geq \gamma_p$. It has been shown in [17] that

$$D_{(k)}^2 = \frac{\sum_{i=1}^k \frac{(P_i'd)^2}{\gamma_i}}{\left(1 - q(1-q)\sum_{i=1}^k \frac{(P_i'd)^2}{\gamma_i}\right)} \tag{2.9}$$

In particular, for $k = 1$, for any single feature i,

$$D_{(1)}^2 = \frac{\frac{(P_i'd)^2}{\gamma_i}}{\left(1 - q(1-q)\frac{(P_i'd)^2}{\gamma_i}\right)} \tag{2.10}$$

Eq. 2.9 shows clearly that the maximum value of the Mahalanobis distance for k features may not correspond to the top principal components corresponding to the largest k eigenvalues.

In short, our use of the Mahalanobis distance is as a (linear) multivariate classifier, where we use a forward feature selection algorithm to rank features. We then follow by classification using a support vector machine on increasingly nested subsets of features.

2.4 Description of Experiments

2.4.1 Experiments using UCID Database

For our experimentation, we used the UCID image database [60] which contains a wide variety of natural images. The UCID database consists of 1338 color images in .tif format of size 512x384 or 384x512. The

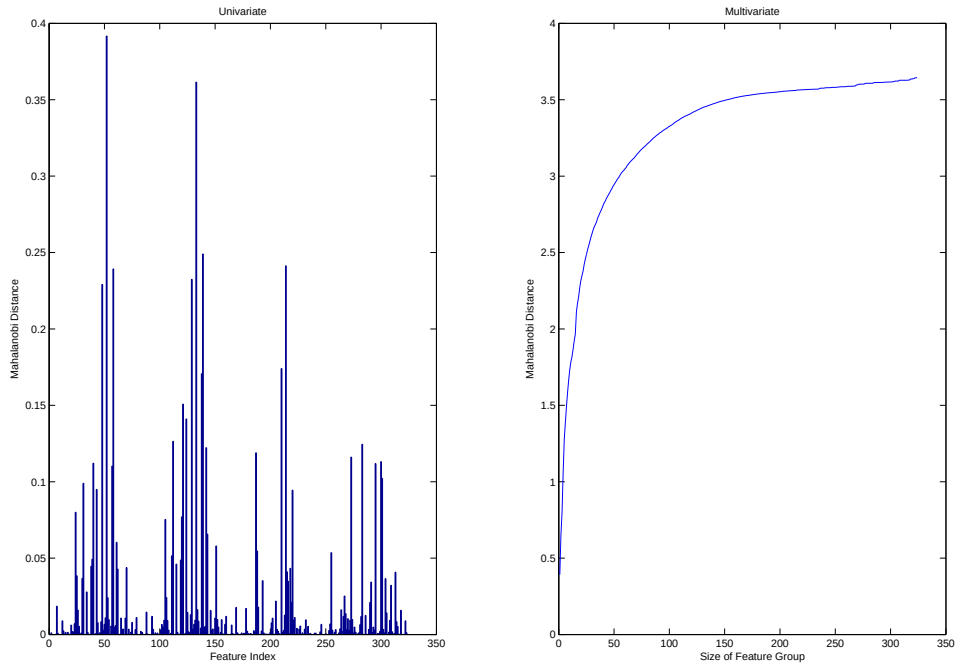


Figure 2.1: Mahalanobi distance graph for univariate and multivariate feature reduction steganalyzer proposed in [61]

images were divided randomly into two disjoint groups. The first group, consisting of 669 images, was used to create the training examples. The second group contained the remaining 669 images that were used to generate stego images to test the accuracy of the classifiers. Thus, no image or its different variations were simultaneously seen by SVM for testing and training, and there were an equal number of images used for cover and stego. This strict division of images enabled us to estimate the performance on never seen images from completely different sources. We intentionally chose to generate our cover and stego images for training and classification in a way to avoid double compression. Recall that a JPEG image is double compressed when it first compressed using quantization matrix Q^1, then uncompressed and recompressed using quantization matrix Q^2, where $Q^2 \neq Q^1$. This is because double JPEG compression can change the statistics of DCT coefficients. We generated stego images by embedding data with embedding messages of length relative to $bpc = 0.05, 0.1, 0.2, 0.4$. We did not use those stego images for which the steganography embedding algorithm exited unsuccessfully. We then extracted the features values from the cover and stego images generated based on the steganalyzer feature set.

Analyzing Markov process based features

In [61], Shi et al. presents a universal steganalyzer based on 324 features calculated directly from the quantized DCT coefficients. The Markov process is applied to model the magnitude difference of JPEG 2-D arrays along horizontal, vertical and diagonal directions so as to utilize second-order statistics for steganalysis. To understand the importance of Markov features, we combine the training files for each steganographic algorithm (each training file contained 669 cover image and 669 stego images of $bpc = 0.05, 0.1, 0.2, 0.4$ with the exception of OutGuess where stego images were generated only with $bpc = 0.05, 0.1, 0.2$) into one file which we used our sample space. This results in 3329 cover images and 3329 stego images. We employ

Mahalanobi's distance to rank the features and apply two search heuristic to reduce the number of features and also increase the accuracy at the same time. In first case, we consider each feature individually and rank them based on their Mahalanobis distance and pick to n features. This is known as Univariate Feature Reduction. In second case, we employ Mahalanobis distance for Multivariate Feature Reduction. Figure 2.1 plots Mahalanobi distance versus features in univariate and multivariate case.

Table 2.1: Feature Reduction using Univariate Mahalanobis distance for Markov Feature Set

	bpc	$n = *$	$n = 10$	$n = 20$	$n = 30$	$n = 40$	$n = 50$	$n = 60$	$n = 70$	$n = 80$
Jsteg	0.05	91.1809	76.6816	77.0553	84.7534	84.006	87.8924	88.4155	89.3871	89.7607
	0.1	99.2526	94.0209	94.0209	97.0105	97.9821	96.7862	97.3842	98.1315	98.7294
	0.2	99.4768	95.441	94.843	97.6831	98.3558	97.0852	97.6084	98.3558	98.9537
	0.4	99.4768	95.2915	94.843	97.4589	98.3558	97.0852	97.6084	98.3558	98.9537
OutGuess	0.05	74.29	65.5456	67.713	69.8057	67.4888	69.0583	68.9836	69.2825	73.8416
	0.1	95.8146	84.5291	86.5471	88.7145	89.5366	90.7324	90.8819	91.1061	94.9925
	0.2	96.7862	90.8072	92.9746	94.6188	93.5725	94.0209	94.1704	93.9462	96.7115
F5	0.05	59.6413	53.139	53.9611	54.4096	54.1854	53.5874	54.6338	54.4843	55.3812
	0.1	73.3931	59.0433	61.9581	61.8087	61.2855	62.9297	63.0045	62.7055	65.7698
	0.2	89.0135	78.6996	81.1659	82.3617	81.6143	85.1271	86.3229	86.9208	88.7145
	0.4	90.0598	91.5546	90.2093	89.6114	91.0314	91.4798	90.5082	92.8999	92.5262
StegHide	0.05	66.4425	60.5381	60.0149	62.1076	62.1076	63.154	62.5561	62.7803	66.5172
	0.1	74.9626	64.7982	64.3498	66.9656	67.1151	69.2078	68.7593	69.6562	74.6637
	0.2	86.846	72.571	73.3184	76.9806	78.849	80.3438	80.7922	81.2407	85.7997
	0.4	92.1525	82.6607	83.3333	86.9208	88.7892	89.4619	89.3124	89.9103	92.003
JPHS	0.05	50	50.299	49.9253	49.8505	50.0747	49.7758	49.7758	50.0747	50.3737
	0.1	51.3453	50.8969	50.9716	51.1958	52.1674	52.2422	51.42	51.2706	51.7937
	0.2	86.1734	70.9268	75.1868	77.8027	80.7922	84.9776	85.5755	85.5755	84.8281
	0.4	98.3307	85.8506	91.5739	93.7202	92.2099	94.7536	96.3434	96.0254	95.9459

Table 2.2: Feature Reduction using Multivariate Mahalanobis distance for Markov Feature Set

	bpc	$n = 10$	$n = 20$	$n = 30$	$n = 40$	$n = 50$	$n = 60$	$n = 70$	$n = 80$
Jsteg	0.05	76.1584	80.7175	82.9596	83.5575	85.6502	87.4439	87.5187	88.1913
	0.1	92.7504	95.142	96.562	95.9641	96.1883	98.5052	98.281	97.6831
	0.2	94.3946	96.0389	97.0105	96.2631	96.6368	99.1779	98.9537	98.4305
	0.4	94.3946	96.1136	97.2347	96.3378	96.7115	99.2526	99.0284	98.5052
OutGuess	0.05	67.8625	68.4604	69.432	73.0194	74.29	73.0942	73.5426	74.9626
	0.1	89.6114	92.0777	92.4514	94.1704	94.0209	94.1704	94.1704	95.441
	0.2	93.8714	94.6936	94.843	95.7399	95.441	96.2631	95.8894	97.2347
F5	0.05	54.1106	54.858	54.858	55.157	55.7549	55.8296	55.4559	59.7907
	0.1	60.9118	64.275	64.8729	66.4425	67.713	68.3109	66.6667	75.5605
	0.2	78.9985	87.3692	89.3124	89.0135	90.5082	90.9567	89.8356	91.4051
	0.4	89.5366	92.6756	92.3019	92.5262	92.8999	93.7967	92.7504	92.3767
Steghide	0.05	61.0613	63.6024	64.7982	63.8266	65.4709	66.2182	65.8445	67.9372
	0.1	66.2182	69.9552	72.272	71.5247	73.2436	73.991	74.29	75.2616
	0.2	76.009	82.9596	85.426	84.9776	85.2018	84.8281	86.0987	86.6218
	0.4	86.9955	90.8072	91.9283	91.704	91.704	91.1809	91.2556	92.4514
JPHS	0.05	50.0747	50	50.1495	49.9253	49.9253	49.7758	50.1495	49.8505
	0.1	51.0463	51.3453	53.5127	53.7369	51.8685	52.3169	52.7653	52.9895
	0.2	76.5321	80.0448	84.7534	85.1271	83.9312	84.006	85.1271	84.8281
	0.4	93.1638	93.7997	92.1304	91.4944	95.9459	92.8458	93.7997	92.6073

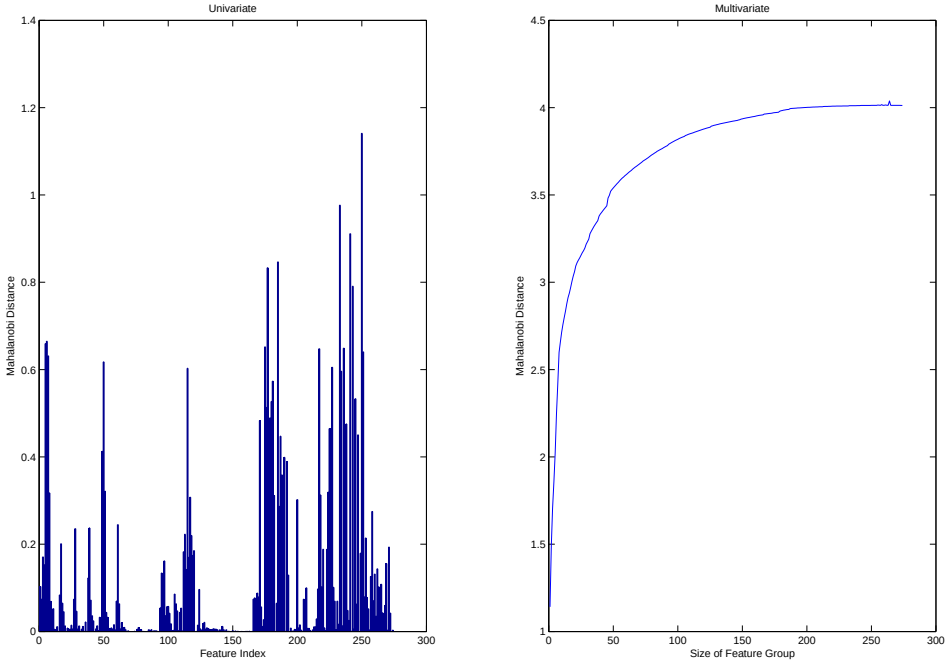


Figure 2.2: Mahalanobi distance graph for univariate and multivariate feature reduction steganalyzer proposed in [53]

Analyzing Merging Extended DCT and Markov process based features

In [53], Pevný et al. proposes a steganalyzer by extending the 23 DCT feature set [26] and applying calibration in calculating the Markov features [61]. To understand the importance of Merged features, we combine the training files for each steganographic algorithm (each training file contained 669 cover image and 669 stego images of $bpc = 0.05, 0.1, 0.2, 0.4$ with the exception of OutGuess where stego images were generated only with $bpc = 0.05, 0.1, 0.2$) into one file which we used our sample space. This results in 3329 cover images and 3329 stego images. We emply Mahalanobi's distance to rank the features and apply two search heuristic to reduce the number of features and also increase the accuracy at the same time. In first case, we consider each feature individually and rank them based on their Mahalanobis distance and pick to n features. This is known as Univariate Feature Reduction. In second case, we employ Mahalanobis distance for Multivariate Feature Reduction. Figure 2.2 plots Mahalanobi distance versus features in univariate and multivariate case.

Table 2.3: Feature Reduction using Univariate Mahalanobis distance for Merged Feature Set

	bpc	$n = *$	$n = 10$	$n = 20$	$n = 30$	$n = 40$	$n = 50$	$n = 60$	$n = 70$	$n = 80$
Jsteg	0.05	96.861	90.8072	95.2167	97.8326	97.3842	96.2631	97.6084	96.1136	97.3842
	0.1	99.701	98.281	99.5516	99.701	100	100	100	100	99.8505
	0.2	99.701	98.8789	99.5516	99.701	100	100	100	100	99.8505
	0.4	99.701	99.0284	99.5516	99.701	100	100	100	100	99.8505
OutGuess	0.05	89.0882	69.7309	83.3333	88.2661	89.7608	89.5366	86.846	91.5546	91.1061
	0.1	98.9537	88.0419	96.7115	97.2347	97.2347	97.0105	97.0105	98.5052	98.58
	0.2	99.0284	92.4514	96.9357	97.3094	97.2347	97.0105	97.1599	98.5052	98.58
F5	0.05	62.4813	57.1001	60.0149	60.6129	61.0613	60.0149	61.0613	61.5097	62.1076
	0.1	83.707	72.272	82.5859	83.1839	81.2407	81.8386	82.8102	83.3333	83.7818
	0.2	96.4873	95.142	96.6368	96.861	96.562	95.8894	96.0389	96.3378	95.9641
	0.4	96.6368	97.4589	97.6084	97.3842	97.1599	96.4873	96.562	96.7115	96.3378
StegHide	0.05	71.9731	63.154	69.9552	70.3288	69.9552	71.2257	68.6846	72.4963	72.6457
	0.1	81.3901	69.133	76.6069	77.8027	77.8027	78.5501	77.5785	82.3617	81.9133
	0.2	90.2093	77.2795	85.8744	87.145	86.9208	87.7429	85.5007	89.8356	89.4619
	0.4	94.5441	85.5007	89.9103	90.8819	91.3303	92.0777	89.9851	92.1525	92.6756
JPHS	0.05	79.5217	65.5456	83.4081	82.287	81.9133	81.2407	82.4365	81.6891	80.7175
	0.1	85.0523	66.4425	85.1271	84.3797	83.8565	83.707	86.3229	85.0523	84.6786
	0.2	94.7683	81.8386	92.8999	92.3767	92.3019	91.9283	93.2735	92.8999	94.0209
	0.4	97.2973	91.256	96.5819	96.8998	96.7409	96.7409	96.6614	96.8203	97.2973

Table 2.4: Feature Reduction using Multivariate Mahalanobis distance for Merged Feature Set

	bpc	$n = 10$	$n = 20$	$n = 30$	$n = 40$	$n = 50$	$n = 60$	$n = 70$	$n = 80$
Jsteg	0.05	92.9746	94.6936	96.1136	97.9821	98.3558	97.6831	97.3094	97.4589
	0.1	99.4021	99.2526	99.3274	99.7758	100	100	100	99.9253
	0.2	99.4021	99.3274	99.3274	99.7758	100	100	100	99.9253
	0.4	99.4021	99.3274	99.3274	99.7758	100	100	100	99.9253
OutGuess	0.05	82.8102	83.4828	83.707	90.3587	91.0314	92.2272	92.3019	91.704
	0.1	97.1599	97.7578	98.3558	98.58	98.8789	98.8789	98.8789	99.0284
	0.2	97.1599	97.7578	98.58	98.6547	98.8789	98.8789	98.8789	99.0284
F5	0.05	57.6981	60.2392	61.2855	60.8371	61.3602	61.6592	61.7339	62.5561
	0.1	76.8311	79.6712	81.9133	82.8102	84.0807	83.8565	83.5575	84.006
	0.2	96.4126	96.562	96.7115	95.6652	97.0105	96.1136	96.1136	95.9641
	0.4	97.8326	97.2347	97.1599	96.3378	97.4589	96.4873	96.4126	96.3378
StegHide	0.05	68.5351	69.5815	70.7025	71.6741	73.3931	72.7952	73.3184	73.1689
	0.1	76.3827	77.6532	78.1016	80.4185	82.287	81.2407	82.5859	81.9133
	0.2	86.2481	87.9671	88.1913	89.6861	90.8819	90.9567	90.7324	90.583
	0.4	91.2556	92.3019	92.1525	92.8251	93.0493	93.2735	92.9746	93.1988
JPHS	0.05	81.9133	83.1091	82.2123	82.3617	82.4365	81.7638	81.3154	83.0344
	0.1	84.006	85.9492	87.2197	86.9208	87.0703	85.8744	85.2018	87.3692
	0.2	92.7504	93.0493	95.142	95.441	95.441	94.5441	94.9178	94.9178
	0.4	97.8537	98.1717	98.0922	98.0127	98.2512	97.6948	98.4897	97.2973

Analyzing Calibrated Markov Feature Set

Note that in Figure 2.2 (Univariate), last 81 features corresponds to Calibrated Markov Features introduced in [53]. Based on this graph, it is interesting to note that Calibrated Markov features has high Mahalanobis distance and hence may contain high discriminatory power for steganalysis. Based on this observation, we analysis 81 Calibrated Markov features separately in this section. To understand the importance of Calibrated Markov features, we combine the training files for each steganographic algorithm (each training file contained 669 cover image and 669 stego images of $bpc = 0.05, 0.1, 0.2, 0.4$ with the exception of OutGuess where stego images were generated only with $bpc = 0.05, 0.1, 0.2$) into one file which we used our sample space. This results in 3329 cover images and 3329 stego images. We emply Mahalanobi's distance to rank the

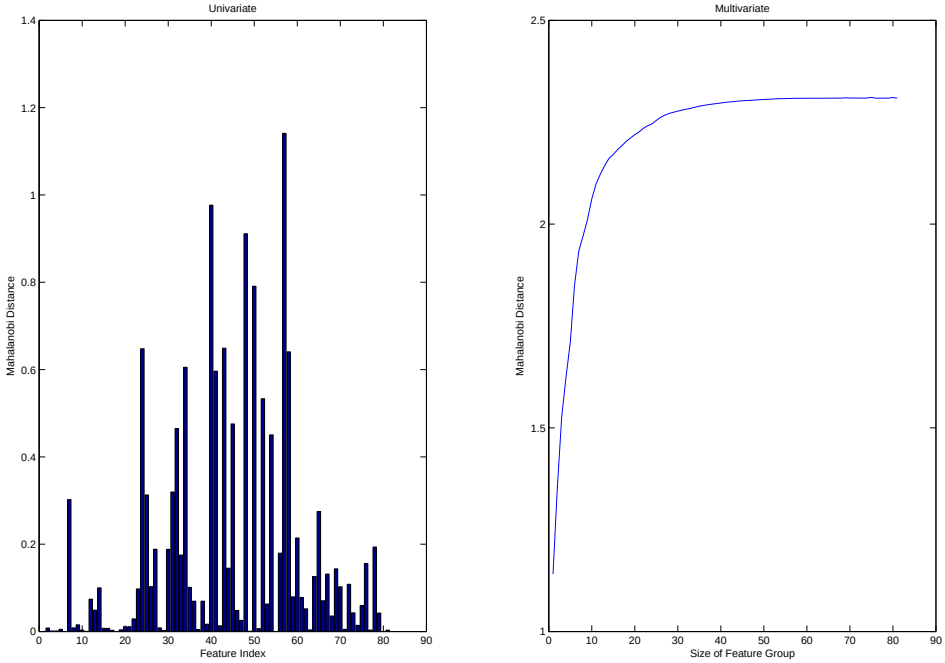


Figure 2.3: Mahalanobi distance graph for univariate and multivariate feature reduction Calibrated Markov Feature set

features and apply two search heuristic to reduce the number of features and also increase the accuracy at the same time. In first case, we consider each feature individually and rank them based on their Mahalanobis distance and pick to n features. This is known as Univariate Feature Reduction. In second case, we employ Mahalanobis distance for Multivariate Feature Reduction. Figure 2.4.1 plots Mahalanobi distance versus features in univariate and multivariate case.

Table 2.5: Feature Reduction using Univariate Mahalanobis distance for Calibrated Markov

	bpc	$n = *$	$n = 10$	$n = 20$	$n = 30$	$n = 40$	$n = 50$	$n = 60$	$n = 70$	$n = 80$
Jsteg	0.05	94.4694	94.9178	95.2915	94.843	94.9925	94.3946	93.7967	94.3946	94.0957
	0.1	99.1031	98.3558	98.9537	98.8789	99.1779	98.8789	99.0284	99.0284	98.8789
	0.2	99.1031	98.3558	98.9537	98.8789	99.1779	98.8789	99.1031	99.0284	98.8789
	0.4	99.1031	98.3558	98.9537	98.8789	99.1779	98.8789	99.1031	99.0284	98.8789
OutGuess	0.05	82.2123	82.5859	81.3901	80.9417	80.7175	80.7175	82.287	81.8386	82.287
	0.1	95.0673	93.1241	93.0493	93.7967	93.4978	94.7683	94.1704	94.4694	94.5441
	0.2	95.5904	93.4978	93.4978	94.3946	94.0209	95.2167	94.6188	94.9925	95.0673
F5	0.05	60.4634	58.3707	60.0149	60.3139	60.8371	60.0897	60.4634	60.3886	60.0897
	0.1	79.0732	72.571	76.9058	77.5785	77.9522	78.849	79.3722	79.5217	78.9985
	0.2	95.3662	93.423	94.6188	94.9925	94.9925	94.0957	94.3199	94.3199	95.441
	0.4	95.6652	95.441	95.5157	95.7399	95.5904	94.6188	94.9178	94.843	95.7399
StegHide	0.05	70.5531	70.3288	70.4783	69.8057	70.1794	69.3572	70.3288	70.3288	70.7025
	0.1	77.6532	75.6353	76.0837	75.71	76.3079	76.3827	77.6532	77.2048	77.8027
	0.2	86.5471	83.8565	83.1091	82.7354	83.4081	84.0807	85.2765	86.0987	86.846
	0.4	90.9567	87.2945	86.6966	87.0703	86.7713	87.8924	89.0882	90.0598	91.1809
JPHS	0.05	65.6203	59.9402	65.8445	66.293	65.3961	65.0972	64.7982	65.3961	66.293
	0.1	69.6562	62.4813	68.6846	68.3109	69.133	69.2825	69.3572	70.4036	70.7025
	0.2	88.1913	82.8849	87.0703	85.8744	88.1166	89.3871	89.6114	90.0598	88.4903
	0.4	93.5612	90.3021	93.7997	93.4022	93.7997	94.2766	94.1176	94.8331	93.1638

Table 2.6: Feature Reduction using Multivariate Mahalanobis distance for Calibrated Markov

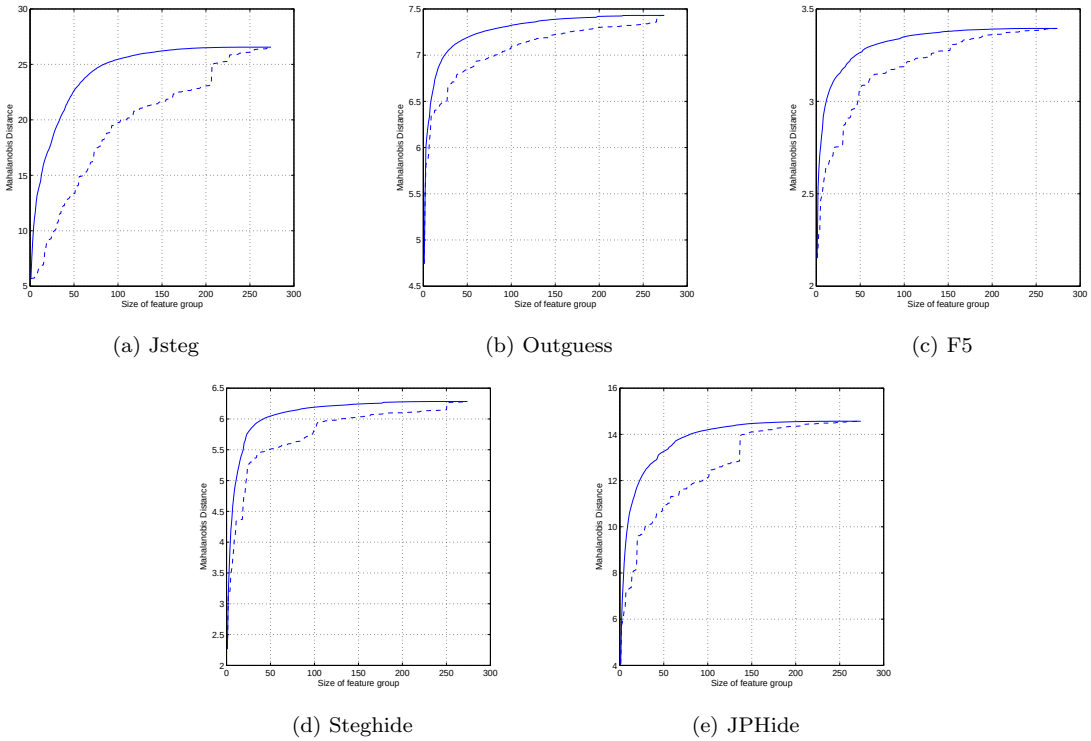
	bpc	$n = 10$	$n = 20$	$n = 30$	$n = 40$	$n = 50$	$n = 60$	$n = 70$	$n = 80$
Jsteg	0.05	92.3767	93.2735	95.2167	93.7967	94.3199	94.1704	94.2451	94.4694
	0.1	98.281	98.8042	99.0284	97.8326	98.4305	98.58	98.8042	99.1031
	0.2	98.3558	98.9537	99.0284	97.9073	98.5052	98.6547	98.8789	99.1031
	0.4	98.3558	98.9537	99.0284	97.9073	98.5052	98.6547	98.8789	99.1031
OutGuess	0.05	81.988	83.4828	82.8102	83.6323	82.8102	82.8102	81.988	81.2407
	0.1	94.4694	95.142	94.9925	94.6936	95.142	94.843	94.1704	94.4694
	0.2	94.7683	95.5157	94.9925	94.9925	95.5904	95.3662	94.6936	95.142
F5	0.05	57.7728	59.8655	58.1465	61.0613	59.4918	60.5381	60.4634	60.0897
	0.1	73.9163	77.6532	75.9342	80.1196	77.8027	78.6248	80.4185	78.849
	0.2	93.2735	94.4694	95.5157	95.2915	95.142	95.0673	95.0673	95.3662
	0.4	95.6652	95.441	96.7115	95.7399	96.0389	95.5904	95.441	95.6652
Steghide	0.05	70.1794	69.7309	71.5994	70.6278	70.2541	70.2541	70.0299	70.2541
	0.1	76.9058	75.9342	77.13	77.6532	76.7564	77.728	76.7564	77.728
	0.2	84.9776	84.0807	85.3513	85.0523	84.9776	85.6502	85.2765	86.9208
	0.4	89.0882	87.9671	89.1629	89.5366	89.1629	89.5366	89.3124	91.1809
JPHS	0.05	65.3961	65.3961	65.7698	66.5919	66.6667	66.8161	66.6667	63.6024
	0.1	69.432	69.0583	69.7309	71.151	69.7309	71.3004	70.5531	68.3109
	0.2	85.5007	87.8176	88.565	88.4155	87.8176	87.8924	88.565	90.3587
	0.4	93.8792	93.5612	94.1971	93.3227	93.7997	93.1638	93.7202	96.0254

2.4.2 Experiments using BOWS2 Database for Merged

Using the heuristic of selecting approximately 10 times as many training examples as there are features [34], we used 5000 original image data for training. Recent research results indicate [47] that using a specified steganalysis scheme - the set of extended DCT feature values from the Merged model having 193 feature variables - a minimum of 5000 image training data was needed to produce classification results that have less than 1 percent standard deviation in their output values. For our experiments, we use the BOWS2 database [66], containing 10 000 images, and pick half of them to create the training database and use the remainder for testing. With approximately 10 000 images for the training set (5000 cover and 5000 stego), we assume that this large data set will deliver relatively low variance in classifier predictability and that

the detection accuracies we obtain produce reliable steganalysis in a comparable sense. Our main interests are in comparing results of the full complement of features to classifiers trained with fewer features. Our assumption is not unreasonable, in particular the data used in [47] was also the BOWS2 database with the extended DCT feature set, very similar to our experiment. Those authors showed that for that data set and feature set, variance of results was experimentally verified to be under 1 percent. Other authors have also used approximately 5000 images for similar research [37, 43, 20].

The BOWS2 database consists of images in pgm format of size 512x512. The images were divided randomly into two disjoint groups of equal size. The first group was used to create the training examples with both cover and stego data. The second group contained the remaining images that were used for testing. We did not use those stegoimages for which the steganography embedding algorithm exited unsuccessfully. Thus, no image or its different variations were simultaneously seen by the SVMs for testing and training, and there were an equal number of images used for cover and stego. This strict division of images enabled us to estimate the performance on never seen images. We intentionally chose to generate our cover and stego images for training and classification in a way to avoid double compression, reformatting the raw pgm files into JPEG files saved with 75% quality factor. Recall that a JPEG image is double compressed when it is first compressed using quantization matrix Q^1, then uncompressed and re-compressed using quantization matrix Q^2, where $Q^2 \neq Q^1$. When an image undergoes double compression, the statistics of DCT coefficients can change, and may resulting in misclassification if the steganalyzer is not designed to handle detection of double compression. Our detection scheme assumes that the data has not been double-compressed.



(a) Jsteg (b) Outguess (c) F5

(d) Steghide (e) JPHide

Figure 2.4: Mahalanobis distance vs. feature group size for five embedding algorithms using features in [53]. Solid line: multivariate distances. Dashed line: univariate distances. Note different scales on distance axes.

We generate stego images by embedding data with different message lengths and different embedding algorithms. The five different steganography embedding methods we consider are: OutGuess [54], F5 [67], JPHide& Seek [3], StegHide [2], and JSteg [4]. Each of these methods embeds bits of value 0 or 1 directly into the quantized DCT coefficient array. The payload is assumed to be an encrypted bitstream. We use

bits per nonzero ac coefficient, or *bpnz* to describe message length, with $bpnz = 0.05, 0.1, 0.2, 0.4$. Images embedded using Outguess have only $bpnz = 0.05, 0.1, 0.2$. Once the stego images are generated, we extract feature values from the cover and stego images according to the steganalyzer by Pevný et al. in [53], which we call *Merged*. The full suite of feature values are extracted initially, and subsets of the feature values are used in accordance to the task at hand.

First, to determine the feature subset using the Mahalanobis distance, we gather all the feature data calculated from the training set for all cover and all embedding algorithms at all levels of embedding. We used 5000 cover and 1250*4 stego images (four levels of embedding) for calculating Mahalanobis distance for each steganography algorithm. Note that in Eq. 2.4, the size of the (pooled) covariance matrix is equal to the number of feature variables used. The regularized covariance matrix is invertible but for each new trial set of k features, the inverse matrix $\hat{V}^{-1}$ must be calculated, and thus this method is fairly computationally time-consuming, especially as k increases. Once the ranking is complete, a classifier is built.

The SVM classifier we implemented was a soft margin support vector machine with gaussian kernel [13] (using LIBSVM [16]). We determined the training parameters of the C-SVMs by grid-search performed on the following multiplicative grid

$$(C, \gamma) \in \left\{ \left(2^i, 2^j\right) | i \in Z, j \in Z \right\}.$$

Although we do not use them, SVMs that use Mahalanobis kernels have been developed for pattern recognition problems. In [29], the authors present a comparison of kernel Mahalanobis distance classifiers giving experimental evidence that these types of classifiers can be advantageous for nonlinear pattern distributions.

We provide two rankings of the features. For the *univariate* ranking, we computed the Mahalanobis distance using individual features. We use Equation 2.6 and a single feature j from the two classes cover and stego, and compute p univariate values. The ranking for the univariate case is simply in decreasing values determined by Equation 2.6. The Mahalanobis distance values for the univariate case are computed separately for each of the five embedding algorithms, where the data for all levels of embedding available for the specified embedding algorithm are combined (4*1250=5000). Once the univariate ranking is determined, the Mahalanobis distance for groups of features of size k, $k = 1, ..., 274$ are calculated. Graphs for these groups of univariate Mahalanobis distance values are shown in Figure 2.4 and indicated by the dashed line in each of the five graphs.

For a second experiment, we used a forward feature selection method to create subsets of features that collectively give Mahalanobis distance values. We call this *multivariate* feature selection. Starting with the feature that gave the largest univariate Mahalanobis distance value in Equation 2.6, the Mahalanobis distance values for all remaining feature vectors paired with the first feature were calculated ($k = 2$ in Equation 2.4), and the feature that gave the largest Mahalanobis distance value over all pairs was selected as the second feature. This continued in an iterative fashion, using the multivariate Mahalanobis distance measure as given by Equation 2.4 for $k = 2, 3, ..., 274$, selecting the feature that gave the largest Mahalanobis distance value at each iteration. This produced a second ranking of the features that was different from the univariate ranking. The multivariate Mahalanobis distance values for groups based on this second ranking are shown in Figure 2.4 and indicated by the solid line in each of the graphs. Based on the graphs in Figure 2.4, it is clear that every feature used in Pevný et al. [53], paper contributes some information that increases the Mahalanobi distance monotonically.

Jsteg

Univariate Ranking:
236 250 245 227 244 259 241 180 224 251 176 169 216 256 225 4 242 263 5 223 171 175 179 235 229 268 217 253 185 184 177 48 61 49 172 234 182 181 252 116 186 2 272 189 60 243 248 117 173 27 3 183 26 207 190 6 37 260 205 191 59 174 178 232 15 7 28 1 218 240 209 118 119 114 101 100 269 247 220 206 13 170 246 11 99 62 129 35 46 262 266 228 16 38 226 233 9 271 108 261 219 211 249 24 58 25 14 267 230 29 231 98 57 47 237 51 36 50 265 64 56 96 12 115 94 34 192 8 130 45 202 31 23 125 188 19 270 238 133 66 274 21 213 212 166 187 196 112 33 102 10 43 17 22 97 54 200 222 52 199 106 138 168 44 131 255 239 198 53 20 55 143 42 124 88 86 221 41 148 193 123 208 194 18 149 210 204 105 203 141 40 151 195 110 104 147 167 80 68 107 81 254

146 87 126 70 120 145 201 63 85 76 67 32 161 144 93 109 122 82 83 134 84 103 30 121 113 264 69 160 65 152 136 137 78 273 95 89 128 135 257 197 162 157 127 165 155 154 139 258 140 158 142 159 111 153 39 132 163 150 156 74 71 91 164 214 79 77 92 90 75 215 72 73

Multivariate Ranking:

236 120 116 233 180 113 106 169 183 119 177 225 6 181 218 102 166 167 114 259 249 170 176 185 117 108 124 50 118 94 51 268 16 15 224 232 174 96 175 5 186 178 8 115 250 238 40 187 244 27 38 254 48 4 215 112 18 214 110 98 97 49 179 101 111 37 99 172 199 173 171 258 129 138 7 182 100 109 198 197 257 241 141 144 261 87 240 216 76 146 122 270 90 131 41 248 251 212 213 121 272 26 31 189 184 20 168 147 202 72 93 14 86 89 58 91 88 243 273 92 128 28 247 220 39 104 47 59 253 33 79 69 68 207 9 22 3 246 134 95 67 43 149 13 136 127 73 71 75 145 263 61 195 82 80 194 83 74 84 188 142 52 70 78 81 77 85 42 274 155 237 63 103 190 162 163 44 158 66 153 60 62 125 156 135 126 19 265 256 219 231 165 239 271 35 2 159 269 217 206 133 139 32 10 1 45 262 36 140 148 152 65 57 266 200 205 154 193 191 264 12 161 160 54 157 208 105 235 30 132 23 226 203 245 242 137 229 210 130 24 228 123 11 46 17 223 196 260 150 230 204 164 21 56 211 107 34 55 222 25 143 252 151 64 192 255 209 53 267 234 29 221 227 201

OutGuess

Univariate Ranking:

251 250 171 179 236 217 232 227 169 176 180 224 243 241 173 189 234 181 6 7 242 5 48 37 177 185 186 50 254 182 256 245 115 26 215 225 240 15 252 200 59 97 238 223 233 95 187 51 96 191 105 183 248 94 192 109 188 237 258 206 271 110 4 228 107 255 98 259 244 226 8 117 27 269 214 100 99 229 266 262 172 166 175 60 20 120 199 184 61 39 101 205 124 167 268 130 53 42 113 174 247 263 231 207 235 19 139 118 3 28 216 272 178 132 190 170 196 142 40 261 103 128 219 108 9 265 13 131 116 64 35 62 140 111 31 119 127 239 134 123 208 30 46 41 260 267 253 104 2 29 230 198 24 57 76 121 63 77 16 22 270 249 145 44 202 133 17 264 10 114 136 125 144 88 85 148 143 1 220 79 218 14 18 87 21 49 157 150 112 222 141 149 38 213 90 91 43 137 257 246 32 86 47 211 11 158 52 80 78 89 102 197 36 146 55 152 193 66 84 92 156 67 25 65 201 12 212 274 33 122 135 221 195 138 160 58 209 151 68 71 69 34 204 210 162 273 153 164 54 161 93 56 194 73 203 126 23 72 70 129 83 75 82 106 154 168 165 159 155 163 81 45 74 147

Multivariate Ranking:

251 250 169 217 227 232 179 50 94 233 197 167 168 258 172 103 115 254 174 5 243 173 242 223 16 249 192 180 118 236 199 252 234 98 124 193 95 241 151 257 188 31 25 120 137 240 108 229 186 224 265 113 45 209 200 47 271 176 133 136 270 170 44 261 129 140 6 106 125 96 247 194 24 158 155 171 76 244 190 4 37 231 185 12 21 253 268 206 215 272 33 222 221 228 101 100 53 142 9 85 15 230 175 178 71 56 1 23 69 67 121 112 274 105 135 42 14 203 213 237 259 162 92 153 90 93 62 107 60 104 38 18 39 225 51 2 164 19 189 201 130 126 117 144 72 202 181 183 157 177 184 86 40 41 34 187 262 10 182 74 205 165 163 260 109 91 49 89 132 208 65 152 81 248 263 77 79 80 83 35 149 150 156 148 32 26 128 211 146 43 141 127 226 111 207 123 28 114 20 7 159 8 30 46 273 54 87 88 145 147 57 210 70 55 66 161 139 82 84 160 196 119 134 143 64 116 61 17 131 220 48 219 63 3 97 154 212 138 11 22 110 269 214 235 246 75 255 256 218 191 266 58 264 239 204 99 267 36 102 52 13 195 198 238 68 122 29 27 166 245 59 73 78 216

F5

Univariate Ranking:

185 177 187 6 233 243 5 7 175 250 234 241 242 170 192 245 258 188 174 181 190 172 191 178 171 183 184 28 50 179 115 246 61 118 249 114 167 244 4 236 120 119 8 237 117 266 247 225 238 116 257 256 193 17 189 112 169 173 232 39 217 113 168 216 271 252 49 51 202 240 62 248 29 60 208 27 220 269 200 265 199 262 9 3 166 211 198 218 224 18 255 37 19 15 41 207 186 239 16 182 231 40 229 38 209 48 260 52 180 176 201 253 254 223 219 274 20 10 2 59 63 26 30 268 230 210 42 14 36 53 235 47 94 222 21 267 272 11 35 261 43 13 31 221 130 1 103 214 25 22 95 54 105 46 139 58 126 64 34 264 110 107 44 45 131 12 123 148 212 55 24 227 86 140 85 157 128 213 125 77 57 194 65 32 68 149 108 228 251 203 76 33 142 206 158 67 70 259 96 66 205 204 132 88 122 109 79 97 195 124 151 133 93 23 160 104 134 263 138 56 98 143 87 215 89 106 90 153 145 91 141 129 101 121 147 102 71 273 150 75 165 135 137 156 83 127 270 197 226 196 82 69 136 84 80 163 161 152 162

144 73 72 146 74 111 159 164 81 154 99 155 78 100 92

Multivariate Ranking:

185 247 227 229 61 116 182 7 118 228 235 190 244 238 107 60 183 197 199 257 125 169 246 267 8 252 249 245 236 4 219 176 5 234 103 115 50 6 181 225 224 119 121 18 62 223 231 149 157 193 28 105 27 114 123 243 29 208 39 73 81 200 11 120 215 259 268 195 162 265 26 166 242 88 263 210 9 3 22 218 239 128 209 51 49 232 20 10 2 220 211 56 272 260 104 38 113 40 122 34 131 158 250 203 206 138 160 126 83 12 55 41 75 194 204 19 72 23 191 202 143 80 57 178 262 240 100 135 117 261 226 241 48 67 71 258 93 168 167 201 141 31 87 25 163 108 111 173 69 1 90 153 216 77 35 76 79 13 170 85 14 92 47 171 15 94 106 58 37 95 172 165 274 273 212 145 32 150 53 142 179 254 30 189 192 175 187 101 54 151 43 214 154 177 207 144 133 52 89 132 96 124 186 65 24 33 269 152 221 159 237 134 63 148 156 137 98 68 99 21 66 161 45 70 17 109 112 139 74 256 16 184 136 127 36 86 129 102 155 78 140 46 188 253 97 147 266 251 180 174 196 110 82 59 164 255 222 91 217 198 248 230 84 233 130 42 44 64 146 213 205 270 264 271

Steghide

Univariate Ranking:

251 250 236 224 183 191 217 171 227 179 232 243 241 245 186 182 176 180 189 173 234 193 181 169 6 50 223 242 7 5 115 185 177 254 225 233 200 256 238 215 187 214 240 252 206 271 248 196 269 226 19 205 51 117 49 52 175 268 228 48 41 130 61 262 247 110 253 255 37 229 263 105 123 199 128 26 39 63 109 127 272 131 219 133 59 121 166 97 113 30 118 120 266 4 95 28 107 15 258 167 9 244 8 116 125 207 218 3 124 96 174 237 134 170 119 94 106 98 136 188 100 192 99 132 101 220 135 260 209 137 126 111 261 267 129 231 114 103 265 18 17 38 40 249 202 27 190 102 235 239 138 178 197 112 42 29 2 246 58 14 270 201 53 198 36 20 47 1 195 257 108 273 45 11 10 143 22 35 31 104 204 230 122 60 87 70 208 25 212 34 77 13 157 184 259 216 172 141 64 93 46 54 152 211 65 150 80 144 159 24 56 146 66 145 210 74 89 73 76 12 163 222 148 162 78 67 75 44 139 84 57 72 33 82 90 142 274 69 264 203 147 21 55 68 62 140 221 16 91 164 168 43 85 154 79 86 156 194 153 165 151 71 158 213 23 32 161 83 92 88 81 155 160 149

Multivariate Ranking:

251 250 173 191 193 169 179 217 235 227 167 168 241 181 183 8 246 6 4 171 216 224 197 50 258 236 242 247 118 200 232 223 59 120 252 182 180 185 31 67 134 113 122 229 254 170 58 177 25 94 199 9 104 124 35 268 27 130 249 138 148 38 41 3 102 233 243 132 234 26 135 147 208 152 220 16 18 115 136 126 225 29 28 114 139 112 82 79 76 192 190 211 269 161 218 96 93 75 266 127 56 1 34 97 88 47 153 137 145 22 162 270 143 80 57 10 214 2 238 70 128 121 12 54 154 101 109 205 267 274 159 44 244 19 90 131 156 62 125 105 5 7 226 187 172 261 166 260 53 195 91 49 150 24 81 51 99 95 149 43 204 248 119 68 196 37 98 123 77 265 111 92 46 52 107 60 61 116 17 133 129 175 188 110 71 221 142 151 237 32 14 85 176 178 63 222 219 30 72 163 160 55 86 39 186 23 87 141 165 40 42 201 157 108 210 144 117 45 245 202 257 203 273 264 103 69 13 106 206 231 21 73 189 84 215 259 48 89 184 64 36 65 15 212 100 155 11 33 146 74 83 256 272 271 174 239 262 255 228 230 240 194 213 198 207 263 158 66 78 209 253 140 20 164

JPHide

Univariate Ranking:

115 50 51 49 177 185 216 113 6 39 234 4 117 243 256 124 174 15 236 167 8 106 238 7 170 5 245 247 181 172 192 188 41 37 19 112 190 17 184 40 178 233 38 187 94 120 246 259 225 237 189 258 48 173 18 252 52 199 179 171 169 207 223 131 193 9 3 61 260 208 215 95 130 183 191 198 175 257 122 242 263 10 2 104 116 241 268 267 20 14 253 266 62 108 13 21 262 271 240 60 28 118 166 261 224 133 205 269 11 1 126 97 235 209 250 272 16 128 186 182 217 123 22 35 43 105 12 110 100 248 222 101 180 196 176 98 168 127 96 226 46 54 121 200 119 114 134 109 139 251 136 137 148 129 211 76 85 111 218 132 26 30 255 249 36 25 204 29 42 31 142 206 44 27 231 227 88 99 244 34 229 59 63 273 157 67 265 232 102 24 32 264 23 77 33 45 210 55 195 270 149 103 135 47 138 107 274 64 230 202 58 53 152 80 254 57 65 140 144 90 89 125 155 82 86 197 83 56 150 143 91 79 221 151 220 154 219 78 66 162 145 81 72 146 153 93 73 147 92 161 163 164 74 160 213 71 156 212 84 201 75 165 228 203 70 194 158 141 87 159 68 69 239 214

Multivariate Ranking:

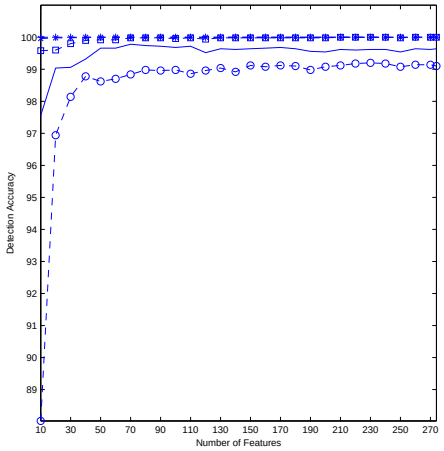
115 216 50 167 225 168 243 177 178 172 199 49 185 61 51 116 198 247 127 190 183 13 189 126 192 123 260 179 21 109 175 274 197 166 233 226 173 36 19 1 27 181 6 187 252 118 227 96 110 104 269 253 170 16 112 17 108 117 5 182 40 113 38 238 184 95 120 128 7 10 208 74 132 131 103 121 202 201 56 29 114 28 249 22 235 130 231 93 39 241 150 212 148 106 133 59 98 251 58 3 47 91 174 23 228 155 54 102 9 188 169 196 266 257 229 163 97 111 8 4 267 270 268 48 94 122 20 14 85 60 107 62 125 134 64 213 100 242 259 273 209 207 101 68 78 217 152 144 157 80 211 18 220 203 272 244 12 76 136 162 153 15 63 261 195 250 245 204 41 186 176 55 11 44 158 164 66 161 160 146 137 205 79 83 92 194 264 119 30 151 2 219 222 87 25 86 141 53 89 31 24 214 193 246 149 43 154 34 84 159 142 239 254 218 82 72 67 215 200 124 88 145 210 32 143 37 139 234 240 52 140 42 271 230 70 69 65 255 46 223 156 77 81 35 237 90 99 33 26 129 57 206 248 224 258 135 191 256 171 265 263 147 138 105 75 73 221 236 262 180 165 232 71 45

Since our goal is to produce classifiers using these subsets and determine if fewer features can produce classification results comparable to the full suite of features, we next created SVMs for feature sets of increasing size. We elected to choose feature subsets of size $10 * M, M = 1, 2, ..., 27$, plus the entire set of 274 features. For each set of features and for each embedding algorithm, we create an SVM, and produce detection accuracy data using the test set. An SVM was created for the 28 feature sets in the univariate ranking, and for the 28 feature sets in the multivariate ranking. To train each SVM, the cover image data provided 5000 feature vectors, and the stego image data (in total) provided 5000 training data. We test for accuracy on each level of *bpnz* for each embedding algorithm and separately for the cover data (no hidden data). For testing, we use the unseen data with 5000 cover images and 5000 stego images for each embedding level, for a total of 25000 test data.

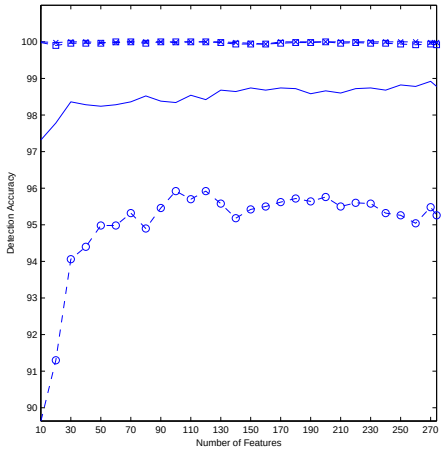
The accuracies for each embedding algorithm and each embedding level are displayed in Figures 2.5 and 2.6. The graphs show that for many instances, accuracies close to the full set of features are reached with many fewer features than the full 274. Table 2.7 displays a few of the top results in number format that supports this observation. Refer to Appendix for complete result.

2.5 Discussion of Results

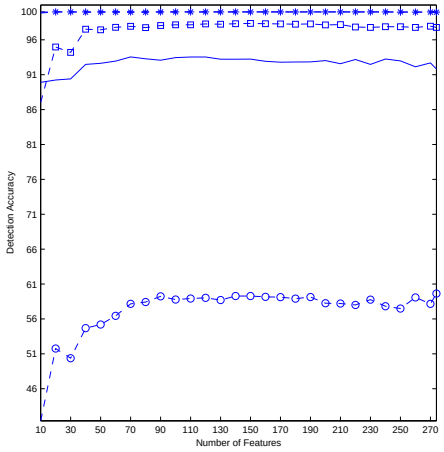
From the graphs and table, we can see there are many instances of feature sets that produce results within 1-2% of the Merged feature set. In general, the multivariate feature sets were able to give results comparable to the univariate feature sets but with fewer features, although for both rankings, the lower levels of embedding required more features than the higher levels of embedding to give comparable results to the full Merged set. Jsteg in its sequential embedding form is known to be straightforward to detect. For the freeware available on the internet, Jsteg embeds payload bits are embedded in a lexicographical order from top left to bottom right of the coefficient array. Embedding in a random order is of course preferable but software code must be written by a user to implement this. Evidence of highly accurate detection rates are displayed in graphs in Fig. 2.2(a) and Fig 2.3(a). Also, full embedding at the 0.4 bpnz level is detected at high levels of accuracies, near 100% for all the steganalyzers of all sizes except for size 10 features in JPHide using the univariate ranking, and sizes 10-30 features in Steghide for multivariate ranking. Lower rates of embedding have lower rates of detection, especially for 0.05 bpnz in F5 where matrix embedding is used. Detection rates for the class cover are generally in the upper 90s% except for F5 and Steghide. The false positive rate for each steganalzyer is 1-(cover detection rate). The way we implemented the SVMs did not allow us to choose detection accuracies based on false positive rates. Higher or lower false positive rates may be desirable or necessary for certain applications. Forensic investigations typically require higher lower positive rates. The general trend was for detection accuracies to increase as more features were added to the classifiers with increases leveling off or oscillating between 50-100 features. However, some classifiers had jumps up or down in accuracy rates as a group of 10 features were added to the current set. This is apparent in Outguess-Univariate and Multivariate, where accuracies increased from 80 to 90 features, and in JPHide-Univariate, where accuracies decreased from 60 to 70 features, and a curious dip in F5-Multivariate from 160 to 170 features. Another phenomenon that occurred was in several instances, cover detection rates changed in the opposite direction of stego detection rates from one size to another. For example, if a user wanted to choose a feature subset based on a high cover detection rate so that the false positive rate was low, then the lowest



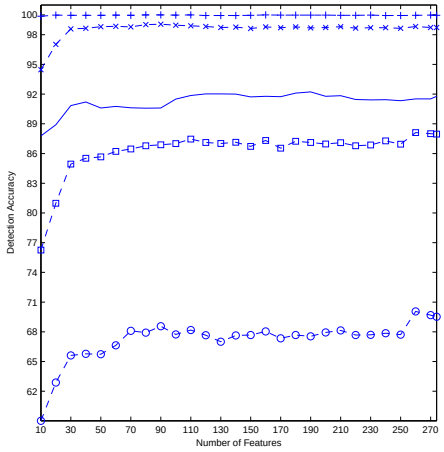
(a) Jsteg



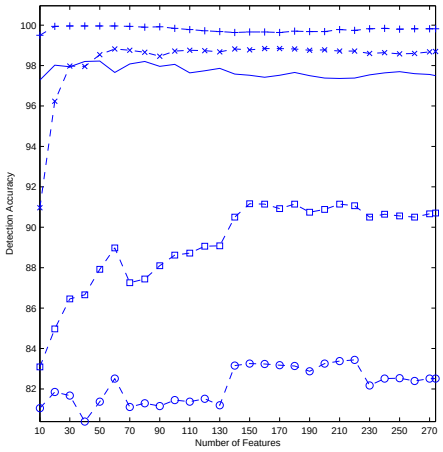
(b) Outguess



(c) F5



(d) Steghide



(e) JPHide

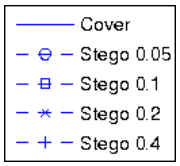
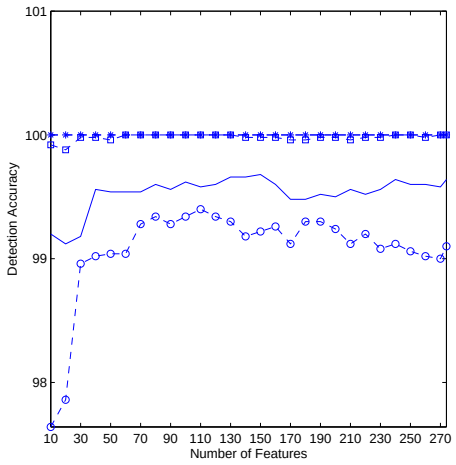
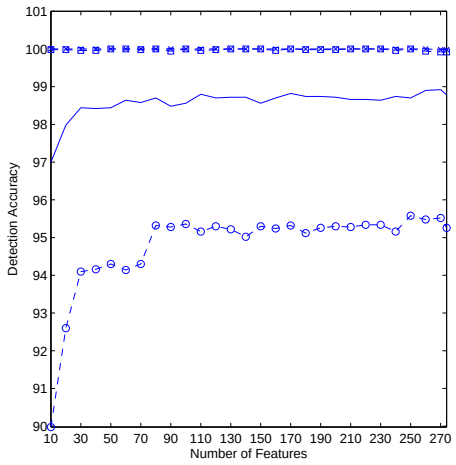


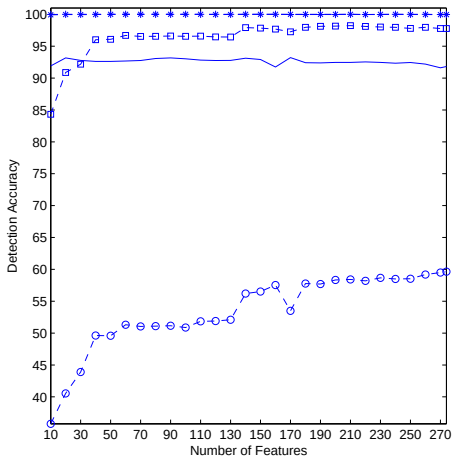
Figure 2.5: Detection accuracy results using univariate feature ranking for groups increasing by 10 features, cover images (solid line) and four embedding rates for five algorithms. Note different scales on distance axes.



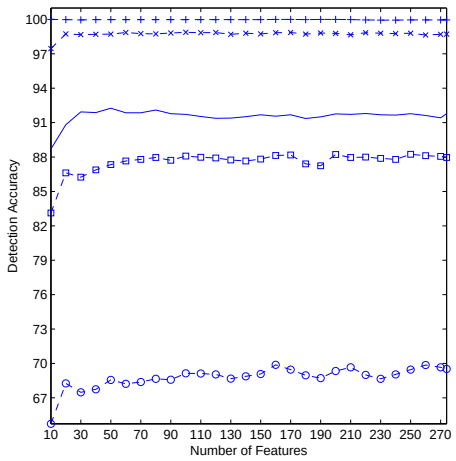
(a) Jsteg



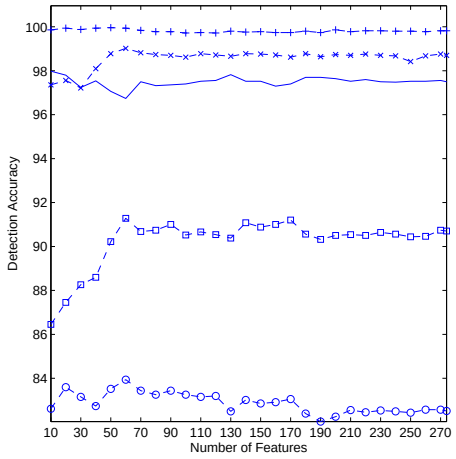
(b) Outguess



(c) F5



(d) Steghide



(e) JPHide

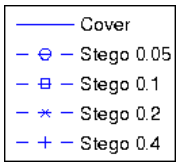


Figure 2.6: Detection accuracy results using multivariate feature ranking for groups increasing by 10 features, for cover images (solid line) and four embedding rates for five algorithms. Note different scales on distance axes.

embedding detection rate may not be as high as possible. This happens in F5-Multivariate, where one of the highest cover detection rates occurs for n=170 features, but that is also one of the lower detection rates for 0.01 embedding. Trade-offs in detection accuracies must be made. In Table 2.7, we give a listing of the top

Table 2.7: Three Top Ranked Classifiers for Univariate and Multivariate Rankings. Displayed for each embedding algorithm and level of embedding. k = number of features.

		Merged	Univariate Ranking						Multivariate Ranking					
			1st	k	2nd	k	3rd	k	1st	k	2nd	k	3rd	k
Jsteg	Cover	99.64	99.78	70	99.74	80	99.72	90	99.68	150	99.66	130	99.66	140
	0.05 bpnz	99.10	99.20	230	99.18	220	99.18	240	99.40	110	99.34	80	99.34	100
	0.10 bpnz	100.00	100.00	210	100.00	220	100.00	230	100.00	60	100.00	70	100.00	80
	0.20 bpnz	100.00	100.00	40	100.00	50	100.00	60	100.00	10	100.00	20	100.00	30
	0.40 bpnz	100.00	100.00	10	100.00	20	100.00	30	100.00	10	100.00	20	100.00	30
Outguess	Cover	98.78	98.92	270	98.82	250	98.78	260	98.92	270	98.90	260	98.82	170
	0.05 bpnz	95.26	95.92	100	95.92	120	95.76	200	95.58	250	95.52	270	95.48	260
	0.10 bpnz	99.92	100.00	60	100.00	70	100.00	90	100.00	50	100.00	60	100.00	80
	0.20 bpnz	99.98	100.00	10	100.00	30	100.00	40	100.00	10	100.00	20	100.00	30
F5	Cover	91.82	93.56	70	93.54	110	93.54	120	93.22	170	93.16	20	93.16	90
	0.05 bpnz	59.64	59.64	274	59.28	140	59.28	150	59.64	274	59.50	270	59.18	260
	0.10 bpnz	97.78	98.36	150	98.32	140	98.32	160	98.24	210	98.16	200	98.12	190
	0.20 bpnz	99.96	100.00	20	100.00	30	100.00	90	100.00	50	100.00	60	100.00	70
	0.40 bpnz	99.98	100.00	20	100.00	30	100.00	40	100.00	20	100.00	30	100.00	40
Steghide	Cover	91.78	92.22	190	92.10	180	92.02	120	92.26	50	92.10	80	91.94	30
	0.05 bpnz	69.52	70.06	260	69.68	270	69.52	274	69.88	160	69.86	260	69.66	210
	0.10 bpnz	87.96	88.12	260	88.00	270	87.96	274	88.24	250	88.22	200	88.18	170
	0.20 bpnz	98.72	99.06	90	99.02	80	98.96	100	98.88	100	98.86	60	98.86	120
	0.40 bpnz	99.96	100.00	80	100.00	90	100.00	110	100.00	10	100.00	160	100.00	170
JPHide	Cover	97.50	98.22	50	98.20	40	98.20	80	97.98	10	97.82	130	97.80	20
	0.05 bpnz	82.51	83.44	220	83.38	210	83.26	150	83.94	60	83.60	20	83.52	50
	0.10 bpnz	90.70	91.16	150	91.14	160	91.14	180	91.28	60	91.20	170	91.08	140
	0.20 bpnz	98.70	98.84	160	98.84	170	98.82	60	99.02	60	98.82	70	98.78	50
	0.40 bpnz	99.82	99.96	30	99.96	40	99.96	50	99.96	50	99.94	20	99.94	40

three ranked feature sets for each ranking algorithm, univariate and multivariate. If a user wanted to reduce the feature size significantly, there are a number of classifiers that give results close to the Merged results. For example, to detect JPHide and Seek cover images to within 1% of the Merged accuracy, 10 multivariate features could be used, and 50-60 could give as good as or better accuracy than the Merged for all levels of embedding for JPHide and Seek. We also found that many of the top features are from Calibrated Markov feature set for all the steganographic algorithm. This gives a strong indication that features which exploit neighborhood dependency among DCT coefficients are effective in steganalysis. We use this information to build a model for steganalysis in the next chapter.

Chapter 3

STEGANALYZER FEATURES DESIGNED ON STOCHASTIC MARKOV MODEL

In this section, we present a new set of features for steganalysis. Experiments in Section 2.4 show that features which exploit neighborhood dependency among DCT coefficients are well suited for steganalysis. A Markov based process has been used with success in [61, 18] to model DCT coefficients for the purpose of steganalysis. In this chapter, we show the use of the Partially Order Markov Model (POMM) [23, 22, 31] for steganalysis which generalizes the concept of local neighborhood directionality by using a partial order underlying the pixel locations.

Markov random fields are a well-known modeling tool and have been used successfully in many areas of image analysis. However, the use of MRFs continues to be problematic when problems require computing an explicit joint probability, such as for texture classification and parameter estimation. A nice subclass of MRFs was introduced by Abend et al [9] for image analysis, called *Markov mesh models* (MMMs). MMMs allow, under minimal and reasonable assumptions, an explicit closed form for the joint probability of the random variables (r.v.s) at hand, expressed in terms of a conditional probability. The conditional probabilities express the spatial dependency of the data, via a directional neighborhood, unlike the undirected neighborhood of a MRF model. Abend et al. also showed that the conditional probability of one r.v. given the rest of the r.v.s can be expressed in terms of r.v.s in a local spatial neighborhood. A partially ordered Markov model generalizes the concept of local neighborhood directionality by using a partial order underlying the pixel locations. It has been shown that this property results in a computational advantage of POMMS over MRFs: whenever the normalizing constant needs to be calculated, such as in determining the joint probability distribution function (pdf), the joint pdf of a POMM is available in closed form, and the normalizing constant for a POMM is always known and equal to the value one [23].

It is beyond the scope of this chapter to give a detailed introduction to POMMs, but a few ideas pertinent to the discussion at hand can be discussed. We assume we have a partial order $\prec$ placed on the set of pixel locations in the image, or a poset. It can be shown that for a given poset $(A, \prec)$, we have a class of acyclic directed graphs where each class corresponds to the same poset $(A, \prec)$ but the individual acyclic directed graphs in the class have different edge sets. The edge sets give rise to the neighborhood relationship between pixel values, which in turn are used to describe conditional probabilities on neighborhoods of pixels. We shall be interested in the Markovian neighborhood relationship, described for POMMs in Definitions 1 and 2, where A is the set of r.v.s in the image, and E a set of edges. Given a directed edge (C, B), tail on C, head on B, we write $C \prec B$ under the partial order $\prec$.

Definition For any $B \in A$, the *cone* of B is the set *cone* $B = \{C \in A : C \prec B, C \neq B\}$.

Definition For any $B \in A$, the *adjacent lower neighbors* of B are those elements $C \in A$ such that (C, B) is

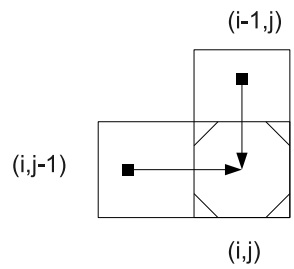


Figure 3.1: The adjacent lower neighborhood $adj_{\prec}(a_{i,j})$. The hash marks represent the location (i, j).

a directed edge in the graph (A, E). Formally, $adj_{\prec}B = \{C : (C, B) \text{ is a directed edge in } (A, E)\}$.

We give a simple example of the adjacent lower neighborhood in Figure 3.1. Here $a_{i,j-1} \prec a_{i,j}$ and $a_{i,j-1} \prec a_{i,j}$. The definition of a POMM is as follows, where L_0 is the set of minimal elements in the poset. The notation $P(A)$ denotes the (discrete) probability measure A, based on the r.v. A. We use the common notation that an upper-case letter denotes the r.v. A, while a realization of A is denoted by a. $P(A)$ is used to denote the probability measure that defines the r.v. A, and $P(A|B)$ is used to denote the conditional probability measure of A given another r.v. B.

Definition The partially ordered Markov model (POMM) is defined as follows: Let $B \in A$ where (A, E) is a finite acyclic digraph of r.v.s and $(A, \prec)$ is its corresponding poset. Describe the set of r.v.s not related to B by $Y_B = \{C : \text{B and C are not related }\}$. Then $(A, \prec)$ is called a *partially ordered Markov model* (POMM) if for any $B \in A \backslash L^0$ and any subset $U_B \subset Y_B$ we have

$$P(B|cone\ B, U_B) = P(B|adj_{\prec}B).$$

For our purposes, we assume that the acyclic digraph underlying the r.v.s of the random image A is the one induced by replicating $adj_{\prec}(a_{i,j})$ at all locations (i, j) in the pixel domain. In other words, the neighborhood is translational invariant. It must be checked that the digraph thus generated is indeed acyclic. Some restrictions that guarantee an acyclic digraph can be found in [23]. For example, the digraph generated by $adj_{\prec}(a_{i,j})$ in Figure 3.1 is acyclic.

This material is sufficient to discuss the features described next. The interested reader is directed to [23].

3.1 POMMS for Steganalysis

With the notation introduced in the previous section, we now discuss the application of POMMs to steganalysis. We develop POMM directly on quantized DCT coefficients array as the embedding takes place directly in that domain. Assume that $\mathbf{A}$ is on a rectangular pixel set, $\mathbf{A} = \{A_{i,j} : 1 \leq i \leq M, 1 \leq j \leq N\}$. Let $\Theta_{i,j}$ be an ordered indexing set that is invariant to shifts on the array where $|\Theta_{i,j}| = n$. For example

$$\Theta_{i,j} = ((i, j), (i, j + 1)) \tag{3.1}$$

is such a set and $|\Theta_{i,j}| = 2$. We describe an invariant cluster set $C_{i,j}$ on the array of r.v.s $\mathbf{A}$ by $C_{i,j} = \{A_{k,h} : (k, h) \in \Theta_{i,j}\}$. A new set of r.v.s $\mathbf{C}$ is defined as $\mathbf{C} = \{C_{i,j} : 1 \leq i \leq M, 1 \leq j \leq N\}$. Let $f : \mathbb{R}^n \to \mathbb{R}$ be a function that exploits the dependency among DCT coefficients in $\Theta_{i,j}$. Apply f via an induced manner to $\mathbf{C}$ by $f(\mathbf{C}) = \mathbf{W}$ where $\mathbf{W} = \{W_{i,j} : W_{i,j} = f(C_{i,j}) \in \mathbb{R}, 1 \leq i \leq M, 1 \leq j \leq N\}$. For example, let $\Theta_{i,j}$ be as in Equation 3.1 and define $f : \mathbb{R}^2 \to \mathbb{R}$ by

$$f(C_{i,j}) = C_{i,j}[0] - C_{i,j}[1] \tag{3.2}$$

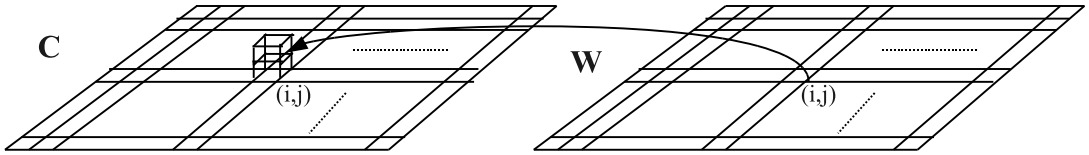


Figure 3.2: Partially Ordered Markov Model for steganalysis

Then f applied to $\mathbf{C}$ produces $f(C_{i,j}) = f(A_{i,j}, A_{i,j+1}) = A_{i,j} - A_{i,j+1} = W_{i,j}$. To create a POMM, we define the partial order as $W_{i,j} \prec C_{i,j}$ for $1 \leq i \leq M, 1 \leq j \leq N$. This in turn defines a directed edge with tail on $W_{i,j}$ and head on $C_{i,j}$. This is shown in Figure 3.2. The adjacent lower neighbors is the set of r.v.s $adj_{\prec} C_{i,j} = W_{i,j}$. The POMM defined by its conditional probabilities, given by $P(C_{i,j}|W_{i,j})$.

This satisfies the definition of a POMM and all the properties now apply. For the probabilities at hand, we are specifically interested in the function f given in Equation **??** and applied to the four directions horizontal h, vertical v, diagonal d, and minor diagonal m. In this case, we have $C^h_{i,j} = (A_{i,j}, A_{i,j+1})$, $C^v_{i,j} = (A_{i,j}, A_{i+1,j})$, $C^d_{i,j} = (A_{i,j}, A_{i+1,j+1})$, $C^m_{i,j} = (A_{i+1,j}, A_{i,j+1})$. Thus in each direction, we calculate $P(C_{i,j}|W_{i,j})$ by noting that

$$P(C_{i,j}|W_{i,j}) = P(C_1, C_2|f(C_1, C_2)) \tag{3.3}$$

$$= P(C_1, C_2|C_1 - C_2) \tag{3.4}$$

$$= \frac{P(C_1, C_2, C_1 - C_2)}{P(C_1 - C_2)} \tag{3.5}$$

which is then perform by histogram binning of the data. If we assume that $-T \leq a_{i,j} \leq T$ for $1 \leq i \leq M, 1 \leq j \leq N$, then $-2T \leq w_{i,j} \leq 2T$ and it is easy see that there will be $(2T+1)^2$ number of such conditional probabilities for each direction. We calculate conditional probabilities, given by $P(C_{i,j}|W_{i,j})$ for all the four different directions and use average of the 4 directional probabilities to characterize f on $\mathbf{A}$. These conditional average conditional probabilities, denoted by $\mathbf{F}$ will be used as features for steganalysis.

We fit the POMM described on the array of quantized DCT coefficients as the embedding takes place directly in this domain. Since the embedding process generally does not take into account the dependency among DCT coefficients, correlations between DCT coefficients in cover and stego image is affected. We fit a POMM as follows to exploit both inter and intra block dependency among DCT coefficients. Before we fit the POMM, we clip quantized DCT coefficients to between [-T,T]. This is necessary, otherwise the pdf for the POMM will be very sparse as DCT coefficient value lies within [-1024,1023]. In our case, we use T=5 as it is well known that more than 96% of the DCT coefficients are found to be within [-5,5] [27, 68]. This results in 121 intra and 121 inter block features and hence a total of 242 features for steganalysis.

3.1.1 Intra block features

Certainly, steganographic embedding causes disturbance on the smoothness, regularity, continuity, consistency, and/or periodicity of quantized DCT coefficients, and therefore correlations among DCT coefficients. To quantify this change, we apply the above developed POMM on the quantized DCT coefficients array $\mathbf{B}$ as described in Section 1.1. We use the conditional probabilities $\mathbf{F}$ of this POMM as intra block features. Since this POMM models the dependency among DCT coefficients within a 8x8 DCT block, we refer to these probabilities as intra block features.

3.1.2 Inter block features

Inter block correlation is reflected among those JPEG modes, i.e., coefficients located in the same position within the 8x8 blocks, which capture the frequency characteristics of those blocks. JPEG steganographic embedding will disturb this kind of interblock correlation. Let the quantized DCT coefficient array $\mathbf{B}$ is of size MxN. Then there are $N_r * N_c$ number of 8x8 DCT blocks where $N_r = \lceil \frac{M}{8} \rceil$ and $N_c = \lceil \frac{N}{8} \rceil$. Let $X^{i,j}$ be the array formed by collecting DCT coefficients located at i, j from every 8x8 blocks. Equivalently, $X^{i,j}_{u,v}$ is the dct coefficient located at i, j in u, v block where $1 \leq u \leq N_r$, $1 \leq v \leq N_c$. The array $X^{i,j}$ is called a *mode array* as it represents mode or specific frequency from every 8x8 block.

To capture inter block dependency, we calculate POMM probabilities as given in Equation 3.5 on every mode array $X^{i,j}$. We use the averaged conditional probabilities of all the mode image as inter block features.

We further apply calibration to reduce the dependency of the feature values on the image content itself. Let I^o be the given image, and let its calibrated image be I^{cal}. We calculate intra and inter block features for I^{cal} also, and then finally use $\mathbf{F^o} - \mathbf{F^{cal}}$ as our features.

3.2 Experiments

It has been shown recently in [39] that the performance of a steganalyzer depends on the database used for training and testing the steganalyzer. Therefore, for our experiments, we use four different databases to compare our steganalyzer based on the proposed feature set with other steganalyzers.

- **Bows2**: This database contains 10000 images of size 512x512 in pgm format. Please refer to Section 2.4 for a detailed description.
- **Camera**: This database consists of 3164 images captured using 24 different digital cameras (Canon, Kodak, Nikon, Olympus and Sony) previously used in [28]. They include photographs of natural landscapes, buildings and object details. All images are of size 512x512 and stored in a raw format (tif) i.e. the images have never undergone lossy compression.
- **Corel**: This database consists of 8185 images from the Corel database [8]. They include images of natural landscapes, people, animals, instruments, buildings, artwork, etc. Although there is no indication of how these images have been acquired, they are very likely to have been scanned from a variety of photos and slides. This database has been previously used in [70]. All images are of size 512x512 and stored in a raw format (tif).
- **NRCS**: This database consists of 2375 images from the NRCS Photo Gallery [5].The photos are of natural scenery, e.g. landscape, cornfields, etc. There is no indication of how these photos were acquired. This database has been previously used in [38]. All images in this database too are of size 512x512 and stored in a raw format (tif).

The last three databases have been downloaded from [24]. We generate the training and testing set for each database separately with a similar process as described in 2.4. For each database, feature set and for each algorithm we train a soft margin support vector machine with gaussian kernel [13] (using LIBSVM [16]). We used the Matlab grid provided by College of Engineering, Iowa State University for feature extraction. It consists of Master - Slave architecture which scheduled the feature extraction jobs on multiple 64 bit computers. Each machine runs on Quad Core Intel Xeon CPU 2.83 GHz and 4GB RAM. This sped up the feature extraction time considerably, as we used another Quad Core Intel Xeon CPU 2.93 GHz and 3GB RAM computer to run the SVMs.

We determined the training parameters of the C-SVMs by grid-search performed on the following multiplicative grid

$$(C, \gamma) \in \left\{ \left(2^i, 2^j\right) | i \in Z, j \in Z \right\}.$$

We compare our results with steganalyzer from feature set proposed in [61], [18] and [53] abbreviating them as Markov324, Markov486 and Merged, respectively. Figure 3.3, 3.4, 3.5 and 3.6 shows the detection accuracy results.

		T=1	**T=2**	**T=3**	**T=4**	**T=5**
Jsteg	Cover	92.67	99.75	99.75	99.83	100.00
	0.05 bpc	24.43	98.15	99.58	98.99	98.99
	0.10 bpc	41.62	100.00	100.00	100.00	100.00
	0.20 bpc	77.93	100.00	100.00	100.00	100.00
	0.40 bpc	98.74	100.00	100.00	100.00	100.00
Outguess	Cover	81.89	99.58	99.92	99.66	99.66
	0.05 bpc	37.18	97.39	98.31	99.16	98.90
	0.10 bpc	61.64	99.92	100.00	100.00	100.00
	0.20 bpc	90.37	100.00	100.00	100.00	100.00
F5	Cover	90.31	93.01	93.18	93.43	92.00
	0.05 bpc	29.65	48.19	56.53	50.72	49.71
	0.10 bpc	74.47	95.79	97.39	96.46	95.20
	0.20 bpc	99.92	99.92	100.00	100.00	100.00
	0.40 bpc	100.00	99.75	100.00	100.00	100.00
Steghide	Cover	71.61	95.37	96.04	96.29	96.12
	0.05 bpc	36.82	74.05	74.56	72.11	73.38
	0.10 bpc	40.86	89.81	92.25	91.24	90.73
	0.20 bpc	49.03	97.73	98.99	98.99	99.16
	0.40 bpc	62.59	99.58	99.75	100.00	100.00
JPHide	Cover	83.40	84.92	89.22	91.49	90.90
	0.05 bpc	29.16	37.45	30.68	28.23	25.44
	0.10 bpc	32.94	44.45	39.63	39.46	37.85
	0.20 bpc	51.86	69.71	84.43	88.83	92.22
	0.40 bpc	90.76	96.02	98.39	99.07	99.66

Table 3.1: Detection accuracy results for POMM based features on NRCS database

		T=1	**T=2**	**T=3**	**T=4**	**T=5**
Jsteg	Cover	86.22	99.18	99.75	99.56	99.24
	0.05 bpc	35.08	91.66	94.69	94.56	94.75
	0.10 bpc	50.51	99.30	99.62	99.62	99.62
	0.20 bpc	78.07	99.81	99.87	99.94	99.94
	0.40 bpc	98.48	99.94	100.00	99.87	100.00
Outguess	Cover	74.53	97.47	97.91	97.47	96.65
	0.05 bpc	52.39	89.80	94.14	94.39	93.69
	0.10 bpc	72.71	99.35	99.81	99.81	99.81
	0.20 bpc	94.03	99.93	99.93	100.00	100.00
F5	Cover	85.21	87.29	87.42	85.40	90.14
	0.05 bpc	44.82	52.97	53.29	54.99	45.70
	0.10 bpc	86.09	93.11	93.55	93.99	89.82
	0.20 bpc	99.05	99.43	99.49	99.56	99.37
	0.40 bpc	99.18	99.43	99.49	99.62	99.49
Steghide	Cover	58.85	90.01	90.52	90.14	89.51
	0.05 bpc	56.01	76.80	79.71	80.28	79.14
	0.10 bpc	60.05	87.55	90.39	90.39	89.82
	0.20 bpc	67.13	96.97	98.23	97.98	97.98
	0.40 bpc	77.24	99.56	99.94	99.81	99.75
JPHide	Cover	85.40	85.71	85.40	90.52	91.28
	0.05 bpc	61.90	69.18	66.90	60.95	60.57
	0.10 bpc	63.12	71.28	70.53	64.45	64.71
	0.20 bpc	72.51	80.30	87.97	87.84	89.42
	0.40 bpc	95.89	97.91	99.05	99.24	99.37

Table 3.2: Detection accuracy results for POMM based features on Camera database

3.3 Discussion of results

We first understand the effect of threshold parameter T on the detection accuracy. Note that different value of T will develop a different POMM and hence we will get a different feature set. We determine the

		T=1	**T=2**	**T=3**	**T=4**	**T=5**
Jsteg	Cover	87.02	97.80	99.76	99.80	99.95
	0.05 bpc	29.79	97.39	99.61	99.58	99.39
	0.10 bpc	54.72	99.98	100.00	100.00	100.00
	0.20 bpc	86.78	100.00	100.00	100.00	100.00
	0.40 bpc	98.53	100.00	100.00	100.00	100.00
Outguess	Cover	75.81	99.00	99.61	99.80	99.80
	0.05 bpc	42.79	97.48	98.58	98.34	98.34
	0.10 bpc	62.07	100.00	100.00	100.00	100.00
	0.20 bpc	89.89	100.00	100.00	100.00	100.00
F5	Cover	86.19	93.55	92.55	92.18	91.79
	0.05 bpc	27.32	37.51	41.10	40.81	40.74
	0.10 bpc	50.42	88.76	91.30	90.69	90.27
	0.20 bpc	99.07	100.00	100.00	100.00	100.00
	0.40 bpc	99.88	100.00	100.00	100.00	100.00
Steghide	Cover	60.68	92.67	94.26	93.94	94.79
	0.05 bpc	46.29	73.34	74.98	75.66	73.78
	0.10 bpc	49.39	89.10	91.59	91.94	90.32
	0.20 bpc	55.33	98.44	99.17	99.19	99.10
	0.40 bpc	66.01	99.95	100.00	99.98	100.00
JPHide	Cover	79.06	83.68	86.17	85.46	85.24
	0.05 bpc	42.18	42.59	43.26	44.33	44.41
	0.10 bpc	44.52	48.80	52.54	54.94	56.23
	0.20 bpc	54.37	72.63	89.44	92.66	92.88
	0.40 bpc	81.80	99.14	99.93	99.90	99.95

Table 3.3: Detection accuracy results for POMM based features on Corel database

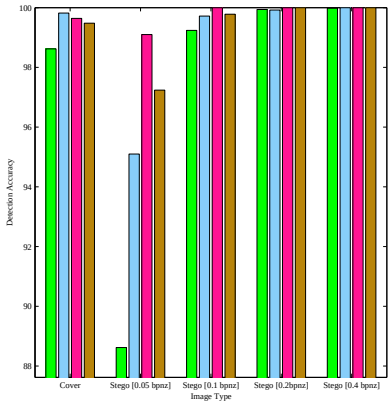
		T=1	**T=2**	**T=3**	**T=4**	**T=5**
Jsteg	Cover	93.18	98.84	99.48	99.46	99.48
	0.05 bpc	33.48	92.72	97.24	96.80	96.38
	0.10 bpc	58.30	99.52	99.78	99.78	99.72
	0.20 bpc	83.48	99.98	100.00	100.00	99.96
	0.40 bpc	97.14	100.00	100.00	100.00	100.00
Outguess	Cover	81.84	98.70	99.38	99.08	99.38
	0.05 bpc	38.63	94.88	96.32	96.90	96.90
	0.10 bpc	62.75	99.84	99.88	99.90	99.92
	0.20 bpc	92.29	100.00	100.00	100.00	100.00
F5	Cover	87.66	90.86	91.12	90.66	89.42
	0.05 bpc	29.16	49.56	51.56	52.66	54.58
	0.10 bpc	66.90	95.90	96.78	96.90	97.18
	0.20 bpc	99.46	99.98	99.98	100.00	99.98
	0.40 bpc	99.78	99.98	99.98	100.00	100.00
Steghide	Cover	66.34	92.48	94.66	93.96	94.34
	0.05 bpc	44.36	74.40	77.28	78.04	77.82
	0.10 bpc	48.60	88.40	91.56	92.14	91.56
	0.20 bpc	55.88	97.36	98.62	98.82	98.74
	0.40 bpc	70.28	99.82	99.88	99.92	99.94
JPHide	Cover	90.34	92.64	94.66	95.14	94.98
	0.05 bpc	64.95	71.73	69.60	70.90	70.80
	0.10 bpc	66.66	74.13	73.25	74.67	75.32
	0.20 bpc	74.18	85.12	94.01	94.25	94.79
	0.40 bpc	92.88	99.14	99.78	99.60	99.58

Table 3.4: Detection accuracy results for POMM based features on Bows2 database

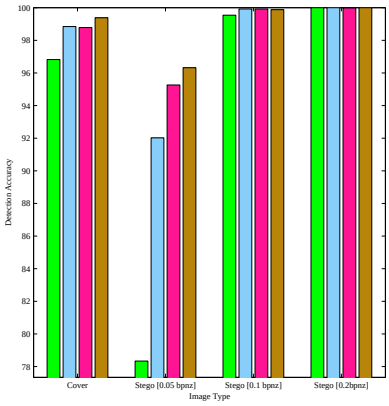
detection accuracy rate for different POMM with $T = 1, 2, 3, 4, 5$ across 4 different database as described in experimental section. It is clear from Table 3.1, 3.2, 3.3 and 3.4 that POMM with $T = 1$ (18 features) and $T = 2$ (50 features) does not capture steganographic changes at lower embedding rate which is true across different database. It is also clear that POMM with $T = 3$ performs better than other POMM in smaller database (NRCS and Camera) for JPHide whereas POMM with $T = 5$ performs better on larger database. This trend is also true for F5. For example, POMM with $T = 3$ gives a detection accuracy of 56% for 0.05 bpnz embedding for F5 algorithm whereas POMM with $T = 5$ gives an accuracy of 50%. But the trend is reversed once you go to a larger database. In Bows2 database which has 10000 images, POMM with $T = 5$ gives a detection accuracy rate of 54% for F5 at 0.05 bpnz whereas POMM with $T = 3$ gives 51%. It can also be seen that POMM with $T = 3$ gives an overall better detection for OutGuess irrespective of database size. Performance of POMM steganalyzer with $T \geq 3$ is approximately same for Jsteg and Steghide across different database. It is clear that based on the database size, a POMM can be build by selecting appropriate value of T. Based on above observation, we decide to chose POMM with $T = 3$ to compare our approach with other steganalyzer proposed in the literature.

From Figure 3.3, 3.4, 3.5 and 3.6, it is clear that the steganalyzer based on our proposed feature set clearly beats Markov324 and Makorv486 in all the databases at all the embedding rates for almost every steganography algorithm. Even though the performances of the steganalyzers are very close to 100% for higher embedding rates, their performances vary much more at lower embedding rates. Note that Markov486 is an extension of Markov324 and the added inter block features have helped in boosting the performance. However, our POMM based steganalyzer shows significant improvement in performance at the lower embedding rates compared either Markov scheme. For example, for Bows2 database, Markov324 and Markov486 has a detection accuracy close to 6% whereas POMM based steganalyzer gave a detection accuracy of 71%.

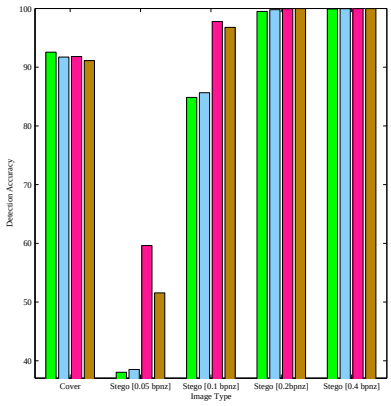
Our proposed steganalyzer has also performed better than Merged for Outguess and Steghide across all the databases whereas Merged performs better at lower embedding rates for F5 and JPHide. Both steganalyzers perform equivalently at higher embedding rates and for detecting Jsteg. For example, the Merged steganalyzer gave a detection accuracy of 55% for Steghide at 0.5 bpnz for Corel database whereas our POMM based steganalyzer gave a detection accuracy of 74%. On the other hand, for the same database, Merged performed better for JPHide at 0.5 bpnz with a detection accuracy of 57% whereas our POMM based steganalyzer detected 44% of the stego images correctly. And this trend holds across different databases.



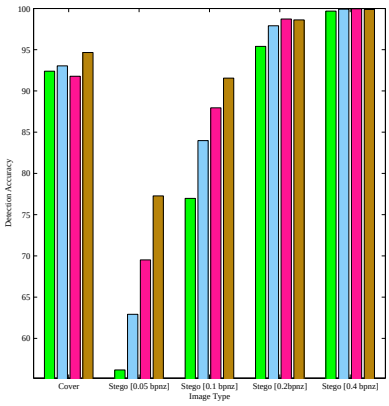
(a) Jsteg



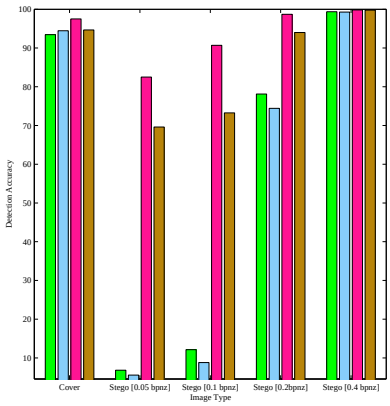
(b) Outguess



(c) F5



(d) Steghide



(e) JPHide

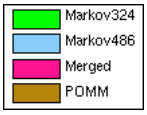
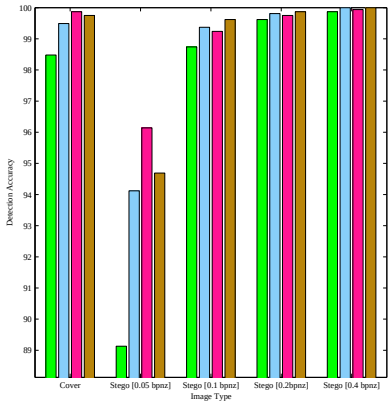
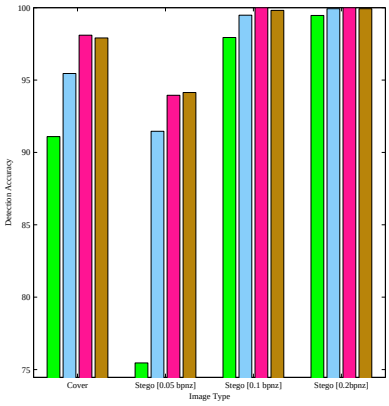


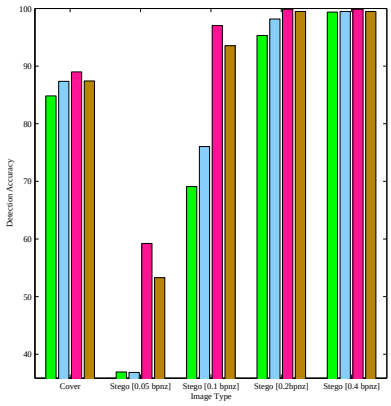
Figure 3.3: Detection accuracy results for different steganalyzers on BOWS2 database



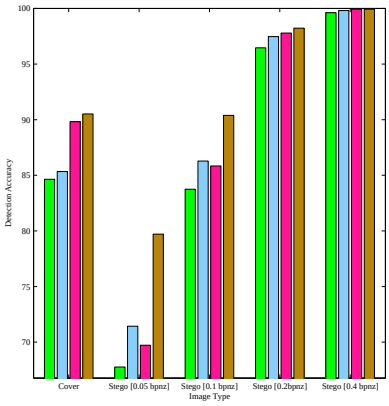
(a) Jsteg



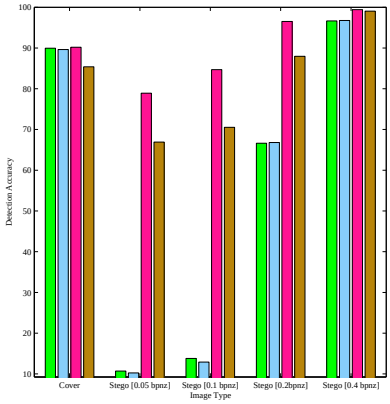
(b) Outguess

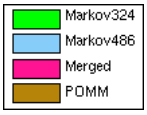


(c) F5



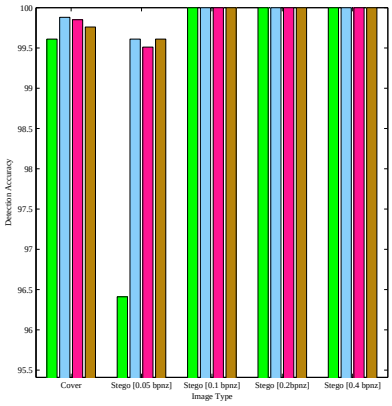
(d) Steghide



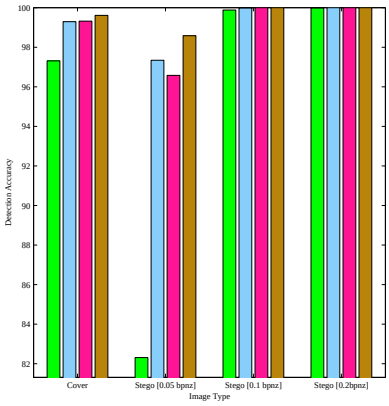


(e) JPHide

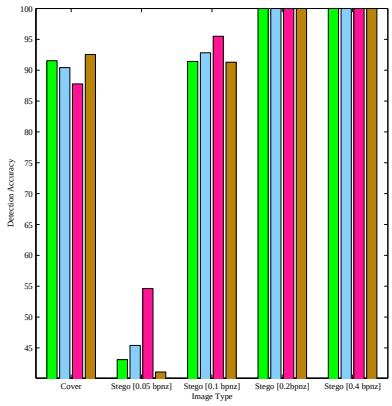
Figure 3.4: Detection accuracy results for different steganalyzers on Camera database



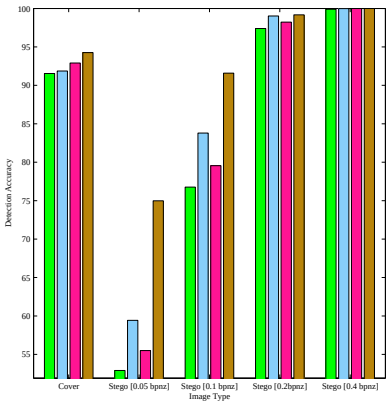
(a) Jsteg



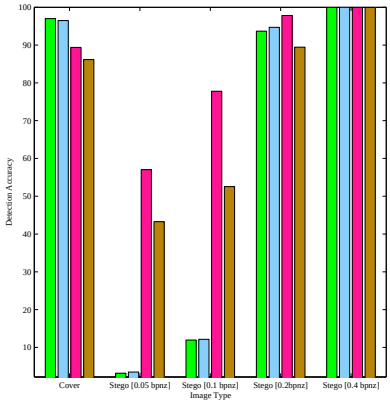
(b) Outguess

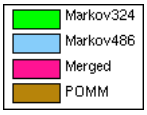


(c) F5



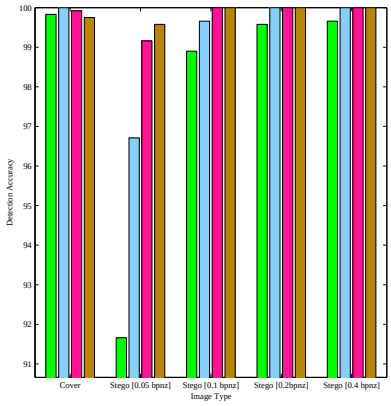
(d) Steghide



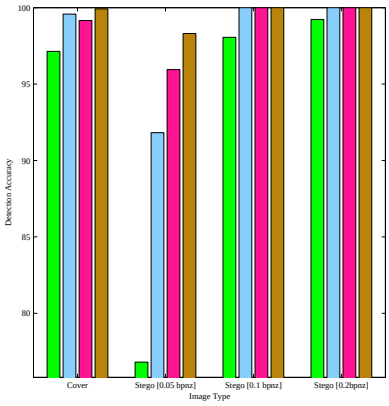


(e) JPHide

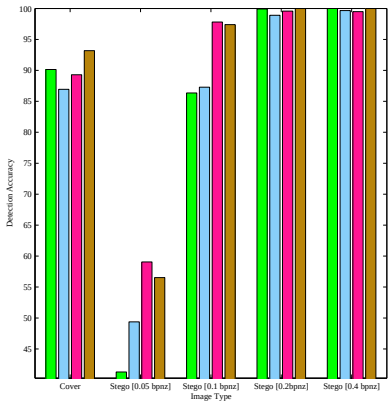
Figure 3.5: Detection accuracy results for different steganalyzers on Corel database



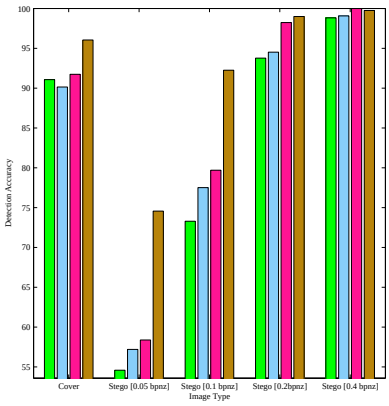
(a) Jsteg



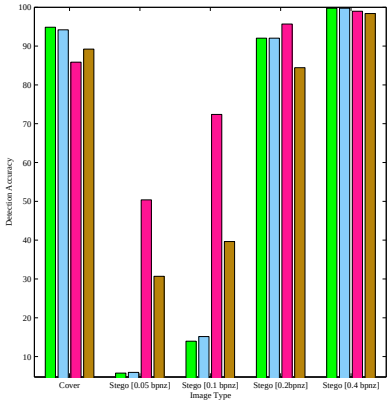
(b) Outguess

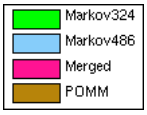


(c) F5



(d) Steghide





(e) JPHide

Figure 3.6: Detection accuracy results for different steganalyzers on NRCS database

3.4 Combining other features with POMM

3.4.1 Experiment-1

Description: With POMM T=5 and Local Histogram described in [53]. Total number of features is 297 (=242+55).
Database: Bows2

	Jsteg	**Outguess**	**F5**	**Steghide**	**JPHide**
Cover	99.58	99.46	90.72	94.88	97.78
0.05 bpc	98.16	96.68	53.28	76.96	78.70
0.10 bpc	99.90	99.94	96.84	91.16	84.67
0.20 bpc	99.98	100.00	100.00	98.86	96.98
0.40 bpc	100.00	-	100.00	99.94	99.94

3.4.2 Experiment-2

Description: With POMM T=5 and Dual Histogram described in [53]. Total number of features is 341 (=242+99).
Database: Bows2

	Jsteg	**Outguess**	**F5**	**Steghide**	**JPHide**
Cover	99.52	99.50	89.82	94.86	98.54
0.05 bpc	98.70	96.52	54.82	76.78	77.68
0.10 bpc	100.00	100.00	96.98	91.28	82.15
0.20 bpc	100.00	100.00	99.98	98.56	95.93
0.40 bpc	100.00	-	100.00	99.94	99.80

3.4.3 Experiment-3

Description: With POMM T=5 and StatMoments described in Chapter 4. Total number of features is 314 (=242+72).
Database: Bows2

	Jsteg	**Outguess**	**F5**	**Steghide**	**JPHide**
Cover	99.28	98.58	0.00	94.10	93.02
0.05 bpc	93.24	89.56	0.00	63.08	69.82
0.10 bpc	99.86	99.50	0.00	80.76	73.47
0.20 bpc	99.96	100.00	0.00	94.10	92.97
0.40 bpc	100.00	-	0.00	99.24	99.42

3.4.4 Experiment-4

Description: With POMM T=5 and DCT features described in [26]. Total number of features is 265 (=242+23).
Database: Various

		NRCS	**Camera**	**Corel**	**Bows2**
Jsteg	Cover	100.00	99.62	99.85	99.48
	0.05 bpc	98.90	93.68	99.41	97.32
	0.10 bpc	100.00	99.62	100.00	99.84
	0.20 bpc	100.00	100.00	100.00	100.00
	0.40 bpc	100.00	100.00	100.00	100.00
Outguess	Cover	100.00	98.48	99.83	99.60
	0.05 bpc	98.48	93.31	97.61	96.38
	0.10 bpc	99.58	99.94	100.00	99.96
	0.20 bpc	99.83	100.00	100.00	100.00
F5	Cover	92.42	87.67	91.98	91.18
	0.05 bpc	55.60	58.66	41.69	58.02
	0.10 bpc	97.64	94.94	91.81	97.92
	0.20 bpc	99.75	99.30	100.00	100.00
	0.40 bpc	99.07	99.30	100.00	99.98
Steghide	Cover	96.04	92.23	95.41	94.74
	0.05 bpc	72.87	77.05	71.41	78.58
	0.10 bpc	91.15	88.75	89.39	92.28
	0.20 bpc	99.07	98.04	99.14	98.94
	0.40 bpc	100.00	99.81	100.00	99.92
JPHide	Cover	90.14	89.06	89.17	97.22
	0.05 bpc	38.29	76.77	51.48	82.21
	0.10 bpc	53.85	81.09	68.92	87.94
	0.20 bpc	92.13	93.79	97.75	98.38
	0.40 bpc	99.07	99.24	99.93	99.86

3.4.5 Experiment-5

Description: With POMM T=5 and following DCT features:

- 11 Global Histogram from [53].
- 5 Local Histogram from [26].
- 11 Dual Histogram from [26].
- 1 Variation from [26].
- 2 Blockiness from from [26].
- 25 Cooccurence from [53].

Total number of features is 297(=242+55).
Database: Various

		NRCS	**Camera**	**Bows2**
Jsteg	Cover	100.00	99.81	99.58
	0.05 bpc	99.24	95.95	97.58
	0.10 bpc	100.00	99.68	99.86
	0.20 bpc	100.00	100.00	100.00
	0.40 bpc	100.00	100.00	100.00
Outguess	Cover	100.00	98.74	99.68
	0.05 bpc	98.48	95.79	96.98
	0.10 bpc	99.49	99.94	99.96
	0.20 bpc	99.66	100.00	100.00
F5	Cover	93.77	90.52	93.00
	0.05 bpc	51.98	54.61	55.74
	0.10 bpc	97.81	95.64	98.26
	0.20 bpc	100.00	99.43	100.00
	0.40 bpc	100.00	99.56	100.00
Steghide	Cover	96.04	92.04	95.18
	0.05 bpc	72.37	76.74	78.62
	0.10 bpc	89.81	91.15	92.60
	0.20 bpc	98.65	98.93	99.22
	0.40 bpc	99.41	99.87	99.94
JPHide	Cover	89.81	90.01	97.20
	0.05 bpc	38.46	77.91	83.32
	0.10 bpc	54.36	82.92	88.76
	0.20 bpc	93.23	96.45	98.94
	0.40 bpc	99.49	99.30	99.82

3.4.6 Experiment-6

Description: With POMM T=3 and DCT features described in [26]. Total number of features is 121 (=98+23).
Database: Various

		NRCS	**Camera**	**Corel**	**Bows2**
Jsteg	Cover	99.75	99.49	99.90	99.66
	0.05 bpc	99.58	97.35	99.54	97.84
	0.10 bpc	100.00	99.81	100.00	99.92
	0.20 bpc	100.00	99.87	100.00	100.00
	0.40 bpc	100.00	96.65	99.95	100.00
Outguess	Cover	100.00	97.79	99.76	99.62
	0.05 bpc	98.82	95.60	97.92	96.92
	0.10 bpc	100.00	99.81	100.00	99.92
	0.20 bpc	100.00	99.93	100.00	100.00
F5	Cover	93.60	89.82	93.99	92.86
	0.05 bpc	60.91	58.60	41.25	57.50
	0.10 bpc	98.82	96.14	93.01	97.96
	0.20 bpc	99.92	99.43	100.00	99.98
	0.40 bpc	99.92	99.49	100.00	99.98
Steghide	Cover	96.04	93.24	95.36	95.02
	0.05 bpc	76.24	78.70	73.61	79.32
	0.10 bpc	93.68	90.64	90.88	92.82
	0.20 bpc	99.16	98.55	99.05	99.20
	0.40 bpc	100.00	99.87	100.00	99.92
JPHide	Cover	85.93	87.42	88.54	96.86
	0.05 bpc	45.14	78.67	54.39	83.50
	0.10 bpc	56.48	82.61	68.39	88.40
	0.20 bpc	88.41	94.30	96.53	98.38
	0.40 bpc	96.95	99.56	99.95	99.80

Chapter 4

Misc experiments

4.1 23 DCT features proposed in [26]

Database Used: Bows2

	Jsteg	**Outguess**	**F5**	**Steghide**	**JPHide**
Cover	98.56	92.66	93.32	76.90	95.52
0.05 bpc	93.08	70.41	44.38	55.06	82.33
0.10 bpc	99.76	99.16	91.34	67.80	87.14
0.20 bpc	99.90	99.90	99.80	85.46	97.14
0.40 bpc	99.90	-	99.76	98.20	99.94

4.2 Dual Histogram

Description: Let $d_{i,j}(k)$ represents i, j quantized DCT coefficient in kth block. Since every block is of size 8x8, therefore $1 \leq i, j \leq 8$ and $1 \leq k \leq n_b$ where n_b are the total number of 8x8 DCT blocks. Then dual histogram is given as:

$$g^d_{i,j} = \sum_{k=1}^{n_b} \delta(d, d_{ij}(k))$$

where $\delta(x, y) = 1$ if $x = y$ and 0 otherwise. We calculate 99 such values by for $d \in \{-5, -4, ..., 4, 5\}$ and $(i, j) \in \{(2, 1), (3, 1), (4, 1), (1, 2), (2, 2), (3, 2), (1, 3), (2, 3), (1, 4)\}$. These are in total 99 features as proposed by Pevný in [53].

To use calibration, calculate $g^d_{i,j}$ for candidate and its calibrated image and use point wise difference values as the final feature values.

Database: Bows2

		with-calibation	**without-calibration**
Jsteg	Cover	97.82	97.54
	0.05 bpc	91.20	93.52
	0.10 bpc	96.44	97.16
	0.20 bpc	98.70	98.74
	0.40 bpc	99.44	99.12
Outguess	Cover	91.94	92.32
	0.05 bpc	69.43	67.51
	0.10 bpc	98.04	95.60
	0.20 bpc	99.88	98.83
F5	Cover	87.12	69.62
	0.05 bpc	33.92	33.72
	0.10 bpc	74.12	38.76
	0.20 bpc	99.36	54.34
	0.40 bpc	99.94	83.20
Steghide	Cover	72.48	68.08
	0.05 bpc	51.32	50.02
	0.10 bpc	60.66	57.76
	0.20 bpc	77.16	70.70
	0.40 bpc	94.86	87.70
JPHide	Cover	96.90	80.84
	0.05 bpc	78.32	19.59
	0.10 bpc	82.19	41.76
	0.20 bpc	86.06	62.85
	0.40 bpc	97.78	87.71

4.3 Local Histogram

Description: Let $\mathbf{h}^{ij}$ is the histogram of (i, j) mode, i.e histogram made by collecting DCT coefficients located at i, j of every 8x8 block. Calculate histogram for modes $(i, j) \in \{(1, 2), (2, 1), (3, 1), (2, 2), (1, 3)\}$ and use bin values for coefficient $l \in \{-5, -4, ..., 4, 5\}$. This will result in 55 features as described in [53]. Calibration technique is used in the similar manner as used in the case of Dual Histogram.
Database: Bows2

		with-calibation	**without-calibration**
Jsteg	Cover	98.16	98.54
	0.05 bpc	88.42	81.10
	0.10 bpc	95.96	95.88
	0.20 bpc	99.54	99.74
	0.40 bpc	99.92	99.98
Outguess	Cover	88.80	89.58
	0.05 bpc	57.53	47.91
	0.10 bpc	92.90	85.95
	0.20 bpc	99.88	97.70
F5	Cover	86.78	68.12
	0.05 bpc	35.62	35.50
	0.10 bpc	67.38	40.30
	0.20 bpc	96.26	57.58
	0.40 bpc	99.90	87.74
Steghide	Cover	67.54	61.86
	0.05 bpc	49.74	45.66
	0.10 bpc	55.88	50.18
	0.20 bpc	67.34	57.80
	0.40 bpc	85.36	70.56
JPHide	Cover	96.22	89.50
	0.05 bpc	76.52	11.27
	0.10 bpc	84.99	37.33
	0.20 bpc	90.67	80.97
	0.40 bpc	99.10	97.65

4.4 Probability density function moments and Characteristic function moments

Description: For the modes $(i, j) \in \{(1,2),(2,1),(3,1),(2,2),(1,3),(1,4),(2,3),(3,2),(4,1)\}$, calculate its probability density function and characteristic function. Then use first four pdf moments and first 4 cf moments. This will result in 72 features. We finally show results for feature set obtained using calibration

- Using difference between features of candidate and calibrated image.
- Using **absolute** value of difference between features of candidate and calibrated image.

Database: Bows2

		without abs value	**with abs value**
Jsteg	Cover	81.76	78.20
	0.05 bpc	46.38	36.88
	0.10 bpc	63.74	47.06
	0.20 bpc	85.98	65.32
	0.40 bpc	97.56	89.20
Outguess	Cover	66.34	62.18
	0.05 bpc	40.93	40.45
	0.10 bpc	47.98	42.38
	0.20 bpc	61.39	48.60
F5	Cover	77.84	62.54
	0.05 bpc	27.58	40.18
	0.10 bpc	34.90	42.84
	0.20 bpc	58.24	53.86
	0.40 bpc	88.70	69.64
Steghide	Cover	47.28	87.24
	0.05 bpc	54.94	13.62
	0.10 bpc	54.66	13.38
	0.20 bpc	56.52	13.56
	0.40 bpc	57.48	13.56
JPHide	Cover	75.08	64.64
	0.05 bpc	57.41	62.96
	0.10 bpc	58.20	63.33
	0.20 bpc	59.72	63.65
	0.40 bpc	68.41	69.37

4.5 324 Markov based features proposed in [61]

Database: Various

		NRCS	**Camera**	**Corel**	**Bows2**
Jsteg	Cover	99.83	98.48	99.61	98.62
	0.05 bpc	91.66	89.13	96.41	88.62
	0.10 bpc	98.90	98.74	100.00	99.24
	0.20 bpc	99.58	99.62	100.00	99.94
	0.40 bpc	99.66	99.87	100.00	99.98
Outguess	Cover	97.14	91.09	97.31	96.82
	0.05 bpc	76.81	75.46	82.31	78.34
	0.10 bpc	98.06	97.94	99.88	99.54
	0.20 bpc	99.23	99.46	99.98	100.00
F5	Cover	90.14	84.83	91.54	92.56
	0.05 bpc	41.28	36.92	43.11	38.04
	0.10 bpc	86.35	69.09	91.42	84.86
	0.20 bpc	99.92	95.32	100.00	99.50
	0.40 bpc	100.00	99.37	100.00	99.92
Steghide	Cover	91.07	84.64	91.54	92.40
	0.05 bpc	54.59	67.76	52.91	56.18
	0.10 bpc	73.29	83.75	76.76	76.98
	0.20 bpc	93.77	96.46	97.39	95.40
	0.40 bpc	98.82	99.62	99.90	99.68
JPHide	Cover	94.86	89.95	96.99	93.44
	0.05 bpc	5.75	10.70	3.18	6.82
	0.10 bpc	13.97	13.79	11.96	12.14
	0.20 bpc	92.05	66.62	93.69	78.13
	0.40 bpc	99.75	96.65	99.98	99.36

4.6 486 Markov based features proposed in [61]

Database: Various

		NRCS	**Camera**	**Corel**	**Bows2**
Jsteg	Cover	100.00	99.49	99.88	99.82
	0.05 bpc	96.71	94.12	99.61	95.10
	0.10 bpc	99.66	99.37	100.00	99.72
	0.20 bpc	100.00	99.81	100.00	99.92
	0.40 bpc	100.00	100.00	100.00	100.00
Outguess	Cover	99.58	95.45	99.29	98.84
	0.05 bpc	91.82	91.46	97.34	92.02
	0.10 bpc	100.00	99.48	99.98	99.92
	0.20 bpc	100.00	99.93	100.00	100.00
F5	Cover	86.94	87.36	90.42	91.72
	0.05 bpc	49.37	36.85	45.41	38.52
	0.10 bpc	87.28	76.04	92.82	85.66
	0.20 bpc	98.90	98.17	100.00	99.80
	0.40 bpc	99.66	99.49	100.00	99.96
Steghide	Cover	90.14	85.34	91.86	93.04
	0.05 bpc	57.20	71.43	59.43	62.92
	0.10 bpc	77.51	86.28	83.80	83.98
	0.20 bpc	94.52	97.47	99.02	97.90
	0.40 bpc	99.07	99.81	99.98	99.92
JPHide	Cover	94.19	89.63	96.48	94.46
	0.05 bpc	5.92	10.25	3.48	5.57
	0.10 bpc	15.16	12.90	12.13	8.80
	0.20 bpc	92.05	66.81	94.69	74.42
	0.40 bpc	99.75	96.77	100.00	99.28

4.7 Extended DCT and Calibrated Markov feature proposed in [53]

Database: Various

		NRCS	**Camera**	**Corel**	**Bows2**
Jsteg	Cover	99.92	99.87	99.85	99.64
	0.05 bpc	99.16	96.14	99.51	99.10
	0.10 bpc	100.00	99.24	100.00	100.00
	0.20 bpc	100.00	99.75	100.00	100.00
	0.40 bpc	100.00	99.94	100.00	100.00
Outguess	Cover	99.16	98.10	99.32	98.78
	0.05 bpc	95.95	93.95	96.58	95.26
	0.10 bpc	100.00	100.00	100.00	99.92
	0.20 bpc	100.00	100.00	100.00	99.98
F5	Cover	89.30	89.00	87.78	91.82
	0.05 bpc	59.06	59.23	54.62	59.64
	0.10 bpc	97.81	97.03	95.50	97.78
	0.20 bpc	99.58	99.87	100.00	99.96
	0.40 bpc	99.49	99.87	100.00	99.98
Steghide	Cover	91.74	89.82	92.91	91.78
	0.05 bpc	58.38	69.72	55.50	69.52
	0.10 bpc	79.70	85.84	79.55	87.96
	0.20 bpc	98.23	97.79	98.22	98.72
	0.40 bpc	100.00	99.94	100.00	99.96
JPHide	Cover	85.85	90.20	89.39	97.50
	0.05 bpc	50.38	78.92	57.04	82.51
	0.10 bpc	72.40	84.69	77.78	90.70
	0.20 bpc	95.69	96.52	97.82	98.70
	0.40 bpc	98.98	99.43	99.95	99.82

4.8 POMM2 - 432 features

Description: In this POMM, we work with absolute values of DCT coefficient. We then, take the difference in three directions and clip the differences between [-2,2]. We then calculate the conditional probabilities for 0,1,2 given its neighbors. We end up with 216 features (intra). We perform the same operation for each mode block and average them to obtain another set of 216 features (inter). Following table shows the results for inter, intra and combined feature set. Calibration is applied by using absolute value of difference between candidate and calibrated image features value.
Database: Bows2

		All	**Intra**	**Inter**
Jsteg	Cover	98.00	96.40	96.72
	0.05 bpc	78.78	65.40	68.12
	0.10 bpc	98.84	98.36	91.24
	0.20 bpc	99.98	99.96	99.16
	0.40 bpc	100.00	100.00	100.00
Outguess	Cover	97.54	90.60	98.18
	0.05 bpc	79.16	41.47	75.96
	0.10 bpc	98.24	91.18	94.46
	0.20 bpc	100.00	99.98	99.50
F5	Cover	91.88	91.32	87.80
	0.05 bpc	22.66	19.80	26.48
	0.10 bpc	67.06	62.44	57.90
	0.20 bpc	99.80	99.82	97.28
	0.40 bpc	100.00	99.98	99.76
Steghide	Cover	93.82	77.02	94.34
	0.05 bpc	63.58	35.76	62.22
	0.10 bpc	79.30	46.00	77.18
	0.20 bpc	93.06	65.04	90.34
	0.40 bpc	99.06	92.06	97.32
JPHide	Cover	87.82	90.34	82.12
	0.05 bpc	68.94	56.39	59.80
	0.10 bpc	71.99	56.16	62.19
	0.20 bpc	87.06	64.35	78.01
	0.40 bpc	98.72	93.06	94.09

4.9 POMM3-243 features

Description: POMM-3 is defined as follows. We take the DCT coefficient array and clip the values between -T to T. After that, we build 3 differences array in horizontal, vertical and diagonal direction, similar to Shi paper. We then clip the values in all three different arrays between -T1 to T1. For this experiment, we have T=4 and T1=1. We end up 243 intra block features. We just look at the accuracy results obtained due to intra block. (There is no calibration involved).
Database: Bows2.

	Jsteg	**Outguess**	**F5**	**Steghide**	**JPHide**
Cover	99.20	91.11	85.67	77.21	88.55
0.05 bpc	91.12	57.55	24.36	47.10	10.13
0.10 bpc	99.36	95.56	57.82	57.34	14.37
0.20 bpc	99.94	99.76	98.82	74.06	72.72
0.40 bpc	99.98	-	99.98	93.54	98.78

In order to explore this model further, we use both inter and intra block probabilities, which results in 486 total features. We then apply calibration and use absolute value of difference between candidate and calibrated image features value.

	Jsteg	**Outguess**	**F5**	**Steghide**	**JPHide**
Cover	99.28	97.72	92.12	88.42	85.32
0.05 bpc	88.90	83.60	17.08	42.96	70.78
0.10 bpc	99.68	99.32	40.66	61.10	70.29
0.20 bpc	99.96	99.98	96.60	85.76	81.17
0.40 bpc	100.00	-	100.00	98.88	98.04

4.10 POMM4 - 270 features

Description:POMM-4 is defined as follows. We first take the absolute of DCT array. After that we calculate, the conditional probability $p(a_{i,j}||a_{i,j} - a_{i-1,j}|, |a_{i,j} - a_{i-1,j-1}|, |a_{i,j} - a_{i,j-1}|)$. The differences are clipped to experimental parameter δ. In this experiment, δ is set to 2. We also, just look at the joint probability for $0 \leq a_{i,j} \leq 4$. So, in total we have 5*3*3*3 = 135 conditional probability. We do this for both inter and intra block and also applying calibration technique. In total we end up with 270 probabilities. We then apply calibration and use absolute value of difference between candidate and calibrated image features value.
Database: Bows2.

		All	**Intra**	**Inter**
Jsteg	Cover	98.48	97.66	97.30
	0.05 bpc	82.94	74.34	76.82
	0.10 bpc	99.44	99.04	98.90
	0.20 bpc	100.00	99.94	99.86
	0.40 bpc	100.00	99.98	99.96
Outguess	Cover	94.80	91.84	92.72
	0.05 bpc	61.13	40.07	60.39
	0.10 bpc	97.06	92.12	93.76
	0.20 bpc	99.94	100.00	99.40
F5	Cover	91.72	93.36	90.40
	0.05 bpc	38.46	18.94	36.78
	0.10 bpc	88.72	61.50	84.96
	0.20 bpc	99.90	99.76	99.84
	0.40 bpc	99.98	99.94	100.00
Steghide	Cover	85.82	77.08	83.76
	0.05 bpc	47.70	37.58	47.42
	0.10 bpc	63.40	46.08	61.94
	0.20 bpc	86.72	66.76	82.70
	0.40 bpc	98.88	93.06	96.52
JPHide	Cover	90.36	89.60	84.46
	0.05 bpc	59.17	57.73	48.02
	0.10 bpc	61.97	58.59	49.85
	0.20 bpc	86.38	74.06	79.99
	0.40 bpc	99.66	98.50	98.86

4.11 POMM 5-6

In this section, we describe the proposed model and feature set generation for steganalysis of JPEG images. We derive features directly from the quantized DCT blocks to attack the steganography algorithm. Our justification is that many steganographic algorithms embed the payload directly in the quantized DCT coefficients. Let dir be a particular direction, then the hierarchial pyramid corresponds to dir is a set of levels $\{L^k_{dir} : k \geq 0\}$. We define the level set corresponding to four different direction: horizontal (L_h), vertical (L_v), major diagonal (L_d) and minor diagonal (L_m) as follows:

$$L^k_h(i,j) = L^{k-1}_h(i,j) - L^{k-1}_h(i,j+1)$$
$$L^k_v(i,j) = L^{k-1}_v(i,j) - L^{k-1}_v(i+1,j)$$
$$L^k_d(i,j) = L^{k-1}_d(i,j) - L^{k-1}_d(i+1,j+1)$$
$$L^k_m(i,j) = L^{k-1}_m(i+1,j) - L^{k-1}_m(i,j+1)$$

where $k \geq 1$. If D is the quantized DCT coefficient array corresponding to a given JPEG image, then

$$L^0_h = L^0_v = L^0_d = L^0_m = D$$

This completely defines the differential pyramid in all four different directions. Note that if L^{k-1}_{dir} lies within the range $[-T,T]$, then from the defintion of L^k_{dir}, $\forall w \in L^k_{dir}, \exists u,v \in L^{k-1}_{dir}$ such that $w = u - v$. Since $u,v \in [-T,T]$, therefore $w \in [-2T,2T]$. Therefore the range of elements in L^k_{dir} is $[-2T,2T]$. We now use Markov process to model the adjacent levels in this pyramid. Let the value at $L^k_{dir}(i,j)$ depends on coefficients from its immediate lower neighborhood L^{k-1}_{dir}. This set of coefficients is known as lower adjacent neighbors of $L^k_{dir}(i,j)$, denoted by $adj(L^k_{dir}(i,j))$. Different neighborhoods can be fitted on this differential pyramid to model different problem. We now define the transition probability set for level k in a particular direction dir given by $p^k_{dir} = \{p^k_{dir}(\overrightarrow{V}|w) : -2T \leq w \leq 2T, \forall v \in \overrightarrow{V} - T \leq v \leq T\}$ for $k \geq 1$ where

$$p^k_{dir}(\overrightarrow{V}|w) = \sum_{i=1}^{m-1}\sum_{j=1}^{n-1} \frac{\delta(adj(L^k_m(i,j)) = \overrightarrow{V})}{\delta(L^k_m(i,j) = w)}$$

where L^{k-1}_{dir} is of the size mxn and

$$\delta(A = B) = \begin{cases} 1 & \text{if } A = B \\ 0 & \text{otherwise} \end{cases}$$

In this paper, we focus on a simple neighborhood given as follows:

$$adj(L^k_h(i,j)) = (L^{k-1}_h(i,j), L^{k-1}_h(i,j+1))$$
$$adj(L^k_v(i,j)) = (L^{k-1}_v(i,j), L^{k-1}_v(i+1,j))$$
$$adj(L^k_d(i,j)) = (L^{k-1}_d(i,j), L^{k-1}_d(i+1,j+1))$$
$$adj(L^k_m(i,j)) = (L^{k-1}_m(i+1,j), L^{k-1}_m(i,j+1))$$

We believe (and further show in the experimental section) that this will effectively model DCT coefficient array to solve the problem of steganalysis based on following intution. During embedding, values are changed at L^0 either randomly or sequentially using LSB matching or LSB replacement. Therefore, it will distort the distribution of the difference between two adjacent coefficients at L^0. Hence the probabilities of occuring of elements will change at L^1. This effect will further be to higher levels, hence creating a ripple effect. The level set probability will effectively capture the distribution of pattern of adjacent coefficients in L^{k-1} given their difference value. Since the cardinality of set p^k_{dir} plays an important role in determining the size of feature space for steganalyzer, we now prove the following claim for the given neighborhood .

		All	**Intra**	**Inter**	**POMM6**
Jsteg	Cover	99.10	99.00	98.38	99.30
	0.05 bpc	93.96	93.88	85.24	94.94
	0.10 bpc	99.84	99.74	98.10	99.74
	0.20 bpc	99.96	99.96	99.88	99.98
	0.40 bpc	100.00	100.00	100.00	100.00
Outguess	Cover	99.14	96.04	98.60	95.80
	0.05 bpc	89.12	45.55	89.48	65.07
	0.10 bpc	99.48	97.54	99.10	98.98
	0.20 bpc	100.00	99.90	99.96	99.98
F5	Cover	90.48	92.56	87.88	91.24
	0.05 bpc	30.96	20.36	32.32	26.52
	0.10 bpc	77.94	61.38	70.32	72.16
	0.20 bpc	99.92	99.84	99.38	99.80
	0.40 bpc	99.98	99.92	100.00	99.96
Steghide	Cover	93.58	76.70	92.14	77.02
	0.05 bpc	64.80	33.64	67.76	36.78
	0.10 bpc	81.66	41.28	82.98	46.88
	0.20 bpc	94.20	59.46	94.04	68.14
	0.40 bpc	99.30	89.60	98.76	94.74
JPHide	Cover	92.02	90.94	85.06	92.06
	0.05 bpc	71.51	50.39	69.58	57.25
	0.10 bpc	74.81	53.62	73.27	63.23
	0.20 bpc	93.65	86.26	88.78	90.25
	0.40 bpc	99.44	99.38	98.14	99.64

Claim: Let dir be any direction and let $k \geq 1$. If L_{dir}^{k-1} lies within the range $[-T, T]$, then $|p_{dir}^k| = (2T+1)^2$.
Proof: $\forall u \in L_{dir}^k$, the possible number of pairs in L_{dir}^{k-1} that can produce u will be $2T - (|u| - 1)$. Since $u \in [-2T, 2T]$, therefore

$$\begin{aligned} |p_{dir}^k| &= \sum_{i=-2T}^{2T} 2T - (|i| - 1) \\ &= 2 * \frac{2T * (2T+1)}{2} + (2T+1) \\ &= (2T+1)^2 \end{aligned}$$

Finally, we define the level transition probabilty set for an image as the average of the level transition probability set in all the possible four directions i.e $\{p^k : k \geq 1\}$ where $p^k = \frac{p_h^k + p_v^k + p_d^k + p_m^k}{4}$. Let I^c be the calibrated version of given image I^o which is produced by decompressing I^o to the spatial domain with its quantization matrix Q^o, cropping 4 pixels from each side in the spatial domain, and then compressing image with Q^o. The spatial shift by 4 pixels and subsequent recompression effectively breaks the statistical relationships of the quantized DCT coefficients because the second compression "does not see" the first compression. As a result, certain macroscopic properties, such as the statistics of DCT coefficients, remain largely unchanged after calibration. More information on calibration can be found in [**?**, **?**, **?**]. If we let D^c represents the JPEG DCT array of the calibrated image, then we define calibrated level probability set as $\text{abs}(p_o^k - p_c^k)$. Figure 1 shows the level transition probability set obtained by using level transition probability set with $k = 1$ and $L_h^0 = L_v^0 = L_d^0 = L_m^0 = D'$ where D' is the DCT coefficient array with values clipped between $[-5, 5]$. It is easy to evaluate the possible number of features in this case is 121.

So we do couple of things. We try evaluating performance of 121 features extracted from DCT array. We also evaluate performance using 2 level with in total 562 features. And then we also try evaluating using inter and intra block features 121+121 = 242 features. Results are shown below

4.12 POMM 7

Let dir be a particular direction, then the hierarchial pyramid corresponds to dir is a set of levels $\{L^k_{dir} : k \geq 0\}$. We define the level set corresponding to four different direction: horizontal (L_h), vertical (L_v), major diagonal (L_d) and minor diagonal (L_m) as follows:

$$L^k_h(i,j) = L^{k-1}_h(i,j) - L^{k-1}_h(i,j+1)$$
$$L^k_v(i,j) = L^{k-1}_v(i,j) - L^{k-1}_v(i+1,j)$$
$$L^k_d(i,j) = L^{k-1}_d(i,j) - L^{k-1}_d(i+1,j+1)$$
$$L^k_m(i,j) = L^{k-1}_m(i+1,j) - L^{k-1}_m(i,j+1)$$

where $k \geq 1$. If D is the quantized DCT coefficient array corresponding to a given JPEG image, then

$$L^0_h = L^0_v = L^0_d = L^0_m = D$$

This completely defines the differential pyramid in all four different directions. After this we build a POMM

		All	**Intra**	**Inter**
Jsteg	Cover	98.68	98.06	97.72
	0.05 bpc	87.20	82.16	86.24
	0.10 bpc	99.60	99.20	99.20
	0.20 bpc	100.00	99.98	99.94
	0.40 bpc	100.00	100.00	100.00
Outguess	Cover	98.36	93.86	97.16
	0.05 bpc	77.76	45.47	77.96
	0.10 bpc	98.22	94.24	97.00
	0.20 bpc	99.98	99.84	99.94
F5	Cover	90.94	91.56	87.50
	0.05 bpc	36.88	27.24	35.78
	0.10 bpc	87.10	75.00	78.80
	0.20 bpc	99.86	99.80	99.68
	0.40 bpc	99.96	99.94	99.98
Steghide	Cover	95.00	83.24	93.62
	0.05 bpc	61.56	37.62	63.92
	0.10 bpc	79.02	49.30	78.84
	0.20 bpc	93.32	71.42	91.58
	0.40 bpc	99.30	94.82	98.38
JPHide	Cover	94.76	93.46	89.24
	0.05 bpc	70.48	58.41	74.15
	0.10 bpc	75.00	65.08	76.72
	0.20 bpc	94.59	92.91	90.43
	0.40 bpc	99.80	99.72	98.70

on the above differential pyramid as follows. Let a coefficient at $L^k(i,j)$ is dependent on its neighbors $adj(L^k(i,j)) \in L^{k+1}(i,j)$. We define the adjacency as:

$$adj(L^k_h(i,j)) = (L^{k+1}_h(i,j-1), L^{k+1}_h(i,j))$$
$$adj(L^k_v(i,j)) = (L^{k+1}_v(i-1,j), L^{k-1}_v(i,j))$$
$$adj(L^k_d(i,j)) = (L^{k+1}_d(i-1,j-1), L^{k+1}_d(i,j))$$

$$adj(L_m^k(i,j)) = (L_m^{k+1}(i-1,j), L_m^{k+1}(i,j-1))$$

We now define the transition probability set for level k in a particular direction dir given by $p_{dir}^k = \{p_{dir}^k(w|\overrightarrow{V}) : 0 \leq w \leq T, \forall v \in \overrightarrow{V} - T \leq v \leq T\}$ for $k \geq 1$ where

$$p_{dir}^k(w|\overrightarrow{V}) = \sum_{i=1}^{m-1} \sum_{j=1}^{n-1} \frac{\delta(L_{dir}^k(i,j) = w, adj(L_{dir}^k(i,j)) = \overrightarrow{V})}{\delta(adj(L_{dir}^k(i,j)) = \overrightarrow{V})}$$

where L_{dir}^{k-1} is of the size mxn and

$$\delta(A = B) = \begin{cases} 1 & \text{if } A = B \\ 0 & \text{otherwise} \end{cases}$$

We now calculate the level transition probability set for k=0 and D formed from the magnitude of DCT coefficient array clipped between [0,5]. These are 227 features. 216 conditional probabilities of elements in L^1 given its neighbor at L^0. So, we calculate these probabilites from both inter and intra block features. In total , we have 227*2 = 454 features.

4.13 POMM8

Description: In this POMM, we explore the inter and intra block dependencies together using different scanning orders for inter and intra block. We use the absolute values of DCT coefficient and clip them between [0,5] before we use them. We use different intra block scanning order and expand each block in a column vector of size 63 (excluding DC coefficients). We then employ different inter block scanning order to arrange all column vectors to make a matrix of size 63 x N_b where N_b is the number of blocks in image.

Finally we apply calibration and use the absolute value of the difference between candidate image and calibrated image as final feature.
Database: Bows2

Intra Block Column Scan

Here we use different inter block scanning and obtain the results as follows:

Table 4.1: Detection Accuracy results for different steganalyzer

		ColumnColumn	**ColumnHilbert**	**ColumnRow**	**ColumnSlalom**
Jsteg	Cover	97.12	97.00	97.72	98.12
	0.05 bpc	79.84	79.22	78.42	74.48
	0.10 bpc	98.80	99.08	98.90	98.52
	0.20 bpc	99.64	99.86	99.80	99.90
	0.40 bpc	99.86	99.96	99.96	100.00
Outguess	Cover	93.52	93.60	94.90	94.72
	0.05 bpc	61.69	61.93	57.07	58.71
	0.10 bpc	98.32	98.14	97.30	97.58
	0.20 bpc	99.96	99.94	99.96	99.96
F5	Cover	89.18	89.12	88.96	89.44
	0.05 bpc	28.44	26.40	26.46	27.20
	0.10 bpc	69.68	67.56	68.36	68.98
	0.20 bpc	99.58	99.84	99.78	99.88
	0.40 bpc	99.82	100.00	100.00	99.98
Steghide	Cover	85.50	85.40	86.60	86.72
	0.05 bpc	45.22	45.16	43.08	44.56
	0.10 bpc	60.58	61.84	59.18	59.56
	0.20 bpc	85.16	85.44	83.64	83.92
	0.40 bpc	99.10	98.80	98.76	98.88
JPHide	Cover	84.82	85.18	84.54	83.60
	0.05 bpc	43.69	42.93	40.75	42.45
	0.10 bpc	45.18	43.84	43.20	44.36
	0.20 bpc	68.54	67.75	66.79	66.43
	0.40 bpc	98.10	98.32	98.68	98.66

Intra Block Row Scan

Table 4.2: Detection Accuracy results for different steganalyzer

		RowColumn	**RowHilbert**	**RowRow**	**RowSlalom**
Jsteg	Cover	98.02	97.60	97.64	97.54
	0.05 bpc	78.10	77.40	75.92	77.78
	0.10 bpc	98.64	98.68	98.58	98.70
	0.20 bpc	99.82	99.84	99.82	99.78
	0.40 bpc	99.96	99.92	99.98	99.94
Outguess	Cover	94.16	94.74	93.44	95.08
	0.05 bpc	55.63	56.33	60.01	55.41
	0.10 bpc	96.90	96.34	97.00	97.44
	0.20 bpc	99.88	99.90	99.86	99.98
F5	Cover	90.70	90.16	89.22	89.98
	0.05 bpc	24.40	23.96	27.60	27.68
	0.10 bpc	63.94	66.42	70.96	71.18
	0.20 bpc	99.68	99.78	99.76	99.90
	0.40 bpc	99.98	99.96	100.00	99.98
Steghide	Cover	86.14	86.06	85.62	85.40
	0.05 bpc	41.36	42.60	42.16	43.80
	0.10 bpc	56.56	59.22	59.00	59.60
	0.20 bpc	80.92	81.96	82.72	84.50
	0.40 bpc	98.24	98.10	98.34	98.40
JPHide	Cover	81.64	82.16	86.40	84.08
	0.05 bpc	33.39	32.38	29.28	31.82
	0.10 bpc	38.25	37.95	34.64	37.97
	0.20 bpc	69.22	71.66	70.64	74.12
	0.40 bpc	98.88	99.48	99.48	99.64

Intra Block ZigZag Scan

		ZigzagColumn	ZigzagHilbert	ZigzagRow	ZigzagSlalom
Jsteg	Cover	97.90	97.94	97.78	97.38
	0.05 bpc	82.36	82.22	82.80	82.16
	0.10 bpc	98.54	98.90	99.04	99.02
	0.20 bpc	99.58	99.78	99.78	99.80
	0.40 bpc	99.88	99.94	99.90	99.92
Outguess	Cover	94.12	93.62	93.72	93.22
	0.05 bpc	57.09	60.35	59.97	62.47
	0.10 bpc	96.52	97.14	97.24	97.28
	0.20 bpc	99.82	99.82	99.78	99.74
F5	Cover	92.04	89.28	87.94	89.64
	0.05 bpc	17.56	22.36	27.34	24.90
	0.10 bpc	52.74	63.02	70.24	67.80
	0.20 bpc	99.60	99.82	99.70	99.86
	0.40 bpc	100.00	100.00	99.98	100.00
Steghide	Cover	84.72	85.16	85.00	85.74
	0.05 bpc	45.80	45.52	46.00	45.92
	0.10 bpc	61.16	61.08	62.76	62.50
	0.20 bpc	84.90	85.44	85.66	86.70
	0.40 bpc	98.44	98.80	99.12	99.00
JPHide	Cover	81.68	85.20	85.52	84.78
	0.05 bpc	39.88	35.75	37.00	37.22
	0.10 bpc	41.41	37.03	38.63	37.63
	0.20 bpc	60.12	58.18	57.50	57.84
	0.40 bpc	97.57	98.16	98.36	98.44

Based on analysis of the above data, it seems that ColumnColumn, RowSlalom and ZigZagRow scanning order gives good results. We end up with two possiblities: Either we take the average of all the three feature set to get another feature set of 216 or we use the together to form a feature set of size 216*3=648. Since the second option results in very high dimensionality of feature set, we will use the first option to derive our final POMM set of 216 features. Results are shown in Table 4.3.

Table 4.3: Detection Accuracy obtained by average of ColumnColumn, RowSlalom and ZigZagRow

	Jsteg	Outguess	F5	Steghide	JPHide
Cover	98.44	96.62	91.90	91.82	87.98
0.05 bpc	88.96	75.34	36.96	52.22	36.58
0.10 bpc	99.08	99.46	88.78	72.64	42.32
0.20 bpc	99.86	99.96	99.96	94.42	81.07
0.40 bpc	99.96	-	100.00	99.62	99.58

4.14 Steganalyzer based on Joint Probability

Description: In this section, we explore a new feature set based on joint probability. For every i, j location in DCT array, we calculate

$$p(q_{i,j} - q_{i,j-1}, q_{i,j} - q_{i-1,j-1}, q_{i,j} - q_{i-1,j}) = \frac{\#(q_{i,j} - q_{i,j-1}, q_{i,j} - q_{i-1,j-1}, q_{i,j} - q_{i-1,j})}{(r-1)*(c-1)} \tag{4.1}$$

where r is the number of rows in the array and c is the number of columns the array. Note that we ignore the locations where the neighborhood is not defined. Therefore, in our case, possible locations are $\{(i,j) : i \in \{2, ..., r\}, j \in \{2, ..., c\}\}$. Since $q_{i,j} \in \{0, 1, ...1024\}$, the difference values will between any two dct coefficient will range from $\{-1024, .., 0, .., 1024\}$, in total 2049 such possible values. This will be huge and computabily difficult. Therefore, we clip the difference values in range $\{-T, T\}$. In order to calculate the above conditional probability, we count the number of patterns $(q_{i,j} - q_{i,j-1}, q_{i,j} - q_{i-1,j-1}, q_{i,j} - q_{i-1,j})$ clipped between $\{-T, T\}$. For this experiment, we set $T = 2$. Therefore, we have 125 intra block and 125 inter block features. In total, we have 250 Joint probability as features (JPF). **Database**: UCID

Table 4.4: Accuracy results using Joint Probability with 3 neighborhood

		250 JPF			250 JPF + 25 Coocurrence		
	bpc	TPR	FPR	AR	TPR	FPR	AR
Jsteg	0.05	86.2481	4.03587	91.1061	90.284	2.09268	94.0957
	0.1	99.4021	4.03587	97.6831	99.8505	2.09268	98.8789
	0.2	100	4.03587	97.9821	100	2.09268	98.9537
	0.4	100	4.03587	97.9821	100	2.09268	98.9537
OutGuess	0.05	72.7952	5.38117	83.707	78.9238	2.69058	88.1166
	0.1	99.5516	5.38117	97.0852	99.701	2.69058	98.5052
	0.2	100	5.38117	97.3094	100	2.69058	98.6547
F5	0.05	30.4933	8.66966	60.9118	31.3901	7.7728	61.8087
	0.1	67.4141	8.66966	79.3722	73.2436	7.7728	82.7354
	0.2	99.5516	8.66966	95.441	99.8505	7.7728	96.0389
	0.4	100	8.66966	95.6652	100	7.7728	96.1136
StegHide	0.05	56.3528	15.2466	70.5531	60.6876	12.2571	74.2152
	0.1	70.852	15.2466	77.8027	74.5889	12.2571	81.1659
	0.2	91.6293	15.2466	88.1913	93.722	12.2571	90.7324
	0.4	99.5516	15.2466	92.1525	100	12.2571	93.8714
JPHS	0.05	51.5695	9.71599	70.9268	41.2556	5.08221	68.0867
	0.1	54.5695	9.71599	72.1973	47.8326	5.08221	71.3752
	0.2	82.6607	9.71599	86.4723	84.9028	5.08221	89.9103
	0.4	98.9813	9.71599	94.3561	99.6604	5.08221	97.1383

4.15 Accuracy results using Joint Probability with 3 neighbhorhood and Quantized DCT coefficient

Description: In this section, we extend joint probabilities introduced in section 2.5. For every i, j location in DCT array, we calculate

$$p(q_{ij}, q_{i,j} - q_{i,j-1}, q_{i,j} - q_{i-1,j-1}, q_{i,j} - q_{i-1,j}) = \frac{\#(q_{ij}, q_{i,j} - q_{i,j-1}, q_{i,j} - q_{i-1,j-1}, q_{i,j} - q_{i-1,j})}{(r-1)*(c-1)} \tag{4.2}$$

where r is the number of rows in the array and c is the number of columns the array. Note that we ignore the locations where the neighbhorhood is not defined. Therefore, in our case, possible locations are $\{(i,j) : i \in \{2, ..., r\}, j \in \{2, ..., c\}\}$. Since $q_{i,j} \in \{0, 1, ...1024\}$, the difference values will between any two dct coefficient will range from $\{-1024, .., 0, .., 1024\}$, in total 2049 such possible values. This will be huge and computabily difficult. Therefore, we clip the difference values in range $\{-T, T\}$. Since $q_{i,j} \in \{0, 1, ...1024\}$, we will have in total $1025 * (2T+1)^3$ number of such conditional probability. We will just look at the joint probability by clipping $q_{i,j} \in \{0, 1, ...T1\}$. Therefore, we will have $(T1+1)*(2T+1)^3$ such joint probability. Since some of them are not possible, they will further be reduced. In this experiment, we set $T1 = 2$ and $T = 2$. So it will be in total 375 features. But since some of them are not possible, it will reduce to 216 features. We combine both inter and intra JPF. **Database**: UCID

Table 4.5: Using 432 Joint Probability as Features

	bpc	TPR	FPR	AR
Jsteg	0.05	91.6293	2.09268	94.7683
	0.1	100	2.09268	98.9537
	0.2	100	2.09268	98.9537
	0.4	99.701	2.09268	98.8042
OutGuess	0.05	70.5531	6.12855	82.2123
	0.1	98.6547	6.12855	96.2631
	0.2	100	6.12855	96.9357
F5	0.05	42.7504	12.855	64.9477
	0.1	82.9596	12.855	85.0523
	0.2	100	12.855	93.5725
	0.4	100	12.855	93.5725
StegHide	0.05	62.0329	14.2003	73.9163
	0.1	77.728	14.2003	81.7638
	0.2	94.6188	14.2003	90.2093
	0.4	99.8505	14.2003	92.8251
JPHS	0.05	28.2511	10.7623	58.7444
	0.1	41.2556	10.7623	65.2466
	0.2	87.5934	10.7623	88.4155
	0.4	99.8302	10.7623	94.1971

It is also interesting to see how the moments in general perform. We calculate centralized first 4 moments.

Table 4.6: Using 32 Joint Probability Moments as Features

	bpc	TPR	FPR	AR
Jsteg	0.05	51.8685	13.9013	68.9836
	0.1	86.6966	13.9013	86.3976
	0.2	97.9073	13.9013	92.003
	0.4	99.4021	13.9013	92.7504
OutGuess	0.05	41.8535	11.2108	65.3214
	0.1	76.6816	11.2108	82.7354
	0.2	98.0568	11.2108	93.423
F5	0.05	35.725	10.1644	62.7803
	0.1	75.7848	10.1644	82.8102
	0.2	99.701	10.1644	94.7683
	0.4	100	10.1644	94.9178
StegHide	0.05	45.8894	25.8595	60.0149
	0.1	54.2601	25.8595	64.2003
	0.2	66.8161	25.8595	70.4783
	0.4	84.006	25.8595	79.0732
JPHS	0.05	60.5381	19.133	70.7025
	0.1	68.7593	19.133	74.8132
	0.2	87.8924	19.133	84.3797
	0.4	98.8115	19.133	89.2687

4.16 Steganalyzer based on Conditional Probability from Partially Ordered Markov Model

POMM model calculates the conditional probability $p(q_{ij}|q_{i,j} - q_{i,j-1}, q_{i,j} - q_{i-1,j-1}, q_{i,j} - q_{i-1,j})$ where $q_{i,j}$ is the absolute value of DCT coefficient at i, j. This probability is gives as

$$p(q_{i,j}|q_{i,j} - q_{i,j-1}, q_{i,j} - q_{i-1,j-1}, q_{i,j} - q_{i-1,j}) = \frac{p(q_{ij}, q_{i,j} - q_{i,j-1}, q_{i,j} - q_{i-1,j-1}, q_{i,j} - q_{i-1,j})}{p(q_{i,j} - q_{i,j-1}, q_{i,j} - q_{i-1,j-1}, q_{i,j} - q_{i-1,j})} \tag{4.3}$$

Since $q_{i,j} \in \{0, 1, ...1024\}$, the difference values will between any two dct coefficient will range from $\{-1024, .., 0, .., 1024\}$, in total 2049 such possible values. This will be huge and computabily difficult. Therefore, we clip the difference values in range $\{-T, T\}$. In order to calculate the above conditional probability, we count the number of patterns $(q_{i,j} - q_{i,j-1}, q_{i,j} - q_{i-1,j-1}, q_{i,j} - q_{i-1,j})$ clipped between $\{-T, T\}$. Then, the probability is given as:

$$p(q_{i,j} - q_{i,j-1}, q_{i,j} - q_{i-1,j-1}, q_{i,j} - q_{i-1,j}) = \frac{\#(q_{i,j} - q_{i,j-1}, q_{i,j} - q_{i-1,j-1}, q_{i,j} - q_{i-1,j})}{(r-1) * (c-1)} \tag{4.4}$$

where r is the number of rows in the array and c is the number of columns the array. Note that we ignore the locations where the neighbhorhood is not defined. Therefore, in our case, possible locations are $\{(i, j) : i \in \{2, ..., r\}, j \in \{2, ..., c\}\}$. Once this probability has been counted, we need to count the $p(q_{ij}, q_{i,j} - q_{i,j-1}, q_{i,j} - q_{i-1,j-1}, q_{i,j} - q_{i-1,j})$. Note that for every value of $q_{i,j}$, there will be in total $(2T+1)^3$ number of such patterns, since every difference value is clipped between $\{-T, T\}$. Furthermore, it is trivial to note that no such patterns will occuer when any of the difference value within the 3 neighbhorhood is greater than $q_{i,j}$. For example, if $q_{i,j} = 0$, then only those patterns will occur where $q_{i,j} - q_{i,j-1} \leq 0$ and $q_{i,j} - q_{i-1,j-1} \leq 0$ and $q_{i,j} - q_{i-1,j} \leq 0$

$$p(q_{ij}, q_{i,j} - q_{i,j-1}, q_{i,j} - q_{i-1,j-1}, q_{i,j} - q_{i-1,j}) = \frac{\#(q_{ij}, q_{i,j} - q_{i,j-1}, q_{i,j} - q_{i-1,j-1}, q_{i,j} - q_{i-1,j})}{(r-1) * (c-1)} \tag{4.5}$$

Hence

$$p(q_{i,j}|q_{i,j}-q_{i,j-1}, q_{i,j}-q_{i-1,j-1}, q_{i,j}-q_{i-1,j}) = \frac{\#(q_{ij}, q_{i,j}-q_{i,j-1}, q_{i,j}-q_{i-1,j-1}, q_{i,j}-q_{i-1,j})}{\#(q_{i,j}-q_{i,j-1}, q_{i,j}-q_{i-1,j-1}, q_{i,j}-q_{i-1,j})} \tag{4.6}$$

Since $q_{i,j} \in \{0, 1, ...1024\}$, we will have in total $1025 * (2T+1)^3$ number of such conditional probability. We will just look at the conditional probability where $q_{i,j} \in \{0, 1, ...T1\}$. There are two ways to restrict $q_{i,j}$:

- By clipping values greater than T to T
- By ignoring values greater than T

Table 2.11 shows the accuracy results based on conditional probability from POMM. For this experiment, we set $T = 2$ and $T1 = 2$. In total,we have 216 intra block conditional and 216 inter block conditional probability.

Table 4.7: Accuracy results based on conditional probability from POMM

		By clipping			By ignoring		
	bpc	TPR	FPR	AR	TPR	FPR	AR
Jsteg	0.05	80.867	1.04634	89.9103	87.8924	1.04634	**93.423**
	0.1	99.2526	1.04634	99.1031	99.701	1.04634	**99.3274**
	0.2	99.701	1.04634	99.3274	99.701	1.04634	**99.3274**
	0.4	99.5516	1.04634	99.2526	99.701	1.04634	**99.3274**
OutGuess	0.05	76.5321	3.13901	**86.6966**	75.1868	2.69058	86.2481
	0.1	98.5052	3.13901	97.6831	98.8042	2.69058	**98.0568**
	0.2	100	3.13901	98.4305	100	2.69058	**98.6547**
F5	0.05	29.8954	4.78326	**62.5561**	29.8954	6.12855	61.8834
	0.1	70.5531	4.78326	**82.8849**	71.3004	6.12855	82.5859
	0.2	99.2526	4.78326	**97.2347**	98.9537	6.12855	96.4126
	0.4	100	4.78326	**97.6084**	99.8505	6.12855	96.861
StegHide	0.05	54.559	7.62332	73.4679	60.8371	8.52018	**76.1584**
	0.1	73.6921	7.62332	83.0344	80.568	8.52018	**86.0239**
	0.2	92.9746	7.62332	92.6756	94.7683	8.52018	**93.1241**
	0.4	99.701	7.62332	**96.0389**	99.701	8.52018	95.5904
JPHS	0.05	52.1674	14.9477	68.6099	50.3737	10.3139	**70.0299**
	0.1	62.0329	14.9477	73.5426	62.7803	10.3139	**76.2332**
	0.2	89.9851	14.9477	87.5187	91.1809	10.3139	**90.4335**
	0.4	100	14.9477	92.0509	100	10.3139	**94.5151**

Since overall, we get higher accuracy by ignoring the values greater than T, we are going to further analyse this model by separating inter and intra block conditional probability.

There are several variations possible here. The variable parameters to experiemnt are:

- Change the Intra Block features : Right now, we are looking at DCT coefficient as one big array to calculate intra block features. In this case, we calculate Intra block features from each 8x8 block and then average them out to get an estimate of overall intra block conditional probability.
- L1 Norm: The above results are produced by using the difference of feature value of candidate and its calibrated image: We can either work with:
  - Absolute value of L1 Norm.
  - Do not take L1 norm and work with features evaluated from candidate image itself.

Following table shows the results obtained by calculating intra block features from each 8x8 block and then average them out to get an estimate of overall intra block conditional probability. We also work with absolute value of L1 Norm.

Table 4.8: Accuracy results of Inter and Intra block based on conditional probability from POMM

		216 Inter			216 Intra		
	bpc	TPR	FPR	AR	TPR	FPR	AR
Jsteg	0.05	77.5785	5.08221	86.2481	83.4081	4.03587	89.6861
	0.1	98.5052	5.08221	96.7115	99.4021	4.03587	97.6831
	0.2	99.701	5.08221	97.3094	99.701	4.03587	97.8326
	0.4	100	5.08221	97.4589	99.701	4.03587	97.8326
OutGuess	0.05	71.7489	3.8864	83.9312	57.8475	6.87593	75.4858
	0.1	95.3662	3.8864	95.7399	95.2167	6.87593	94.1704
	0.2	99.8505	3.8864	97.9821	100	6.87593	96.562
F5	0.05	28.6996	11.3602	58.6697	26.3079	5.8296	60.2392
	0.1	59.6413	11.3602	74.1405	68.9088	5.8296	81.5396
	0.2	95.2167	11.3602	91.9283	99.2526	5.8296	96.7115
	0.4	99.701	11.3602	94.1704	100	5.8296	97.0852
StegHide	0.05	53.2138	10.4634	71.3752	54.559	17.7877	68.3857
	0.1	69.6562	10.4634	79.5964	66.0688	17.7877	74.1405
	0.2	88.9387	10.4634	89.2377	86.0987	17.7877	84.1555
	0.4	96.7115	10.4634	93.1241	98.5052	17.7877	90.3587
JPHS	0.05	54.1106	26.009	64.0508	36.1734	10.3139	62.9297
	0.1	63.0792	26.009	68.5351	43.9462	10.3139	66.8161
	0.2	80.867	26.009	77.429	83.4081	10.3139	86.5471
	0.4	95.7555	26.009	84.1812	99.4907	10.3139	94.2766

4.16.1 POMM on difference array of DCT coefficients

In this case, we use the 4 difference array calculated from dct array in Shi paper. On each array, we fit the POMM. We calculate the conditional probability $p(q_{i,j}|adj_{i,j})$. Before calculating this, we also clip the value of DCT values between $\{2, 2\}$. Therefore in total, we will have 625 conditional probability in each direction. We calibrate feature values in each direction by taking absolute value of difference between candidate and its calibrated image. Since we have 4 direction, it will result in $625 * 4$ number of features. Therefore to reduce the feature space, we take the average of the feature values in each direction, and therefore reducing it to just 625. Note that we are not calculating these conditional probability on mode images.

Table 4.9: Accuracy results by fitting POMM on differences array instead computed in Shi paper

	bpc	TPR	FPR	AR
Jsteg	0.05	36.4723	6.72646	64.8729
	0.1	87.7429	6.72646	90.5082
	0.2	99.701	6.72646	96.4873
	0.4	100	6.72646	96.6368
OutGuess	0.05	43.1988	8.96861	67.1151
	0.1	89.2377	8.96861	90.1345
	0.2	100	8.96861	95.5157
F5	0.05	15.6951	12.855	51.42
	0.1	28.6996	12.855	57.9223
	0.2	86.9955	12.855	87.0703
	0.4	99.8505	12.855	93.4978
StegHide	0.05	39.6114	21.5247	59.0433
	0.1	47.3842	21.5247	62.9297
	0.2	65.7698	21.5247	72.1226
	0.4	89.5366	21.5247	84.006
JPHS	0.05	35.5755	7.7728	63.9013
	0.1	35.1271	7.7728	63.6771
	0.2	44.843	7.7728	68.5351
	0.4	95.7555	7.7728	93.8792

4.16.2 Working with absolute values of differences

In this section, we work with the absolute values of differences, and therefore clip them between $\{0, , ., T\}$. For this experiment, we take T=3 and T1 = 3, and therefore we will have 256 features.

Table 4.10: Using Inter and Intra block conditional probability as features with absolute value of difference

	bpc	TPR	FPR	AR
Jsteg	0.05	84.6039	1.34529	91.6293
	0.1	99.2526	1.34529	98.9537
	0.2	99.701	1.34529	99.1779
	0.4	99.701	1.34529	99.1779
OutGuess	0.05	76.5321	2.69058	86.9208
	0.1	98.6547	2.69058	97.9821
	0.2	100	2.69058	98.6547
F5	0.05	28.849	7.7728	60.5381
	0.1	71.4499	7.7728	81.8386
	0.2	99.1031	7.7728	95.6652
	0.4	100	7.7728	96.1136
StegHide	0.05	65.9193	10.0149	77.9522
	0.1	80.4185	10.0149	85.2018
	0.2	95.0673	10.0149	92.5262
	0.4	99.701	10.0149	94.843
JPHS	0.05	55.3064	7.32436	73.991
	0.1	66.8161	7.32436	79.7459
	0.2	94.7683	7.32436	93.722
	0.4	99.8302	7.32436	96.0254

Chapter 5

CANVASS - A STEGANOGRAPHIC FORENSIC TOOL FOR JPEG IMAGES

There is a growing concern within the community that steganography is being used for illicit purposes. The New York Times, USA Today and United States Institute of Peace have reported that terrorists may be using steganography and cryptography on the web as a means of covert communication [36, 42, 65]. A recent report from National Institute of Justice encourages investigators to look for steganographic information while dealing with child abuse or exploitation and terrorism cases [1]. While steganalysis algorithms are abound in the academic literature, there are few software programs that address the needs of local police departments who perform computer forensic functions for steganalysis.

Two major stego-detection tools in existence today are StegoSuite and StegDetect. StegoSuite was developed by WetStone Technologies for the U.S. Air Force. This software is expensive and hence is not readily available to state police forensics labs whose budget does not allow purchase of these expensive software. In 2001, Neil Provos developed StegDetect [56] to perform steganalysis on suspected stego images. Using this software, he analyzed millions of JPEG images from sites like eBay [57] and USENET [55], but was unable to detect a single image with hidden data in it.
One of the objectives of this project, as mentioned in the original research proposal submitted to Midwest Forensics Resource Center (MFRC), Ames Laboratory, is to develop a software that address the needs of local police departments who perform computer forensic functions for steganalysis. Canvass is a cross-platform software that has been designed after several meetings with Internet Crime Against Children Lab (ICAC), Iowa investigators and understanding their requirement. It is developed in Java with a graphical user interface which implements the steganalzer proposed in Chapter 3 earlier. The current version 1.0 is shown in Figure 5.1 and provides following features:

- Ability to process multiple images with one command. User can specify the source of images from following locations:
  - *Local Machine*: User can specify multiple images located in different folders on the local machine and the software will classify each image against the steganographic algorithms present in the system.
  - *Web*: User can specify a web address to steganalyze jpeg images on the internet starting from the web address specified. Given a link, it searches for jpeg images on the page, and stores all the links on that page which it has not visited. It then recursively performs a search for images on those links in breadth first manner until there are no more new links to visit.

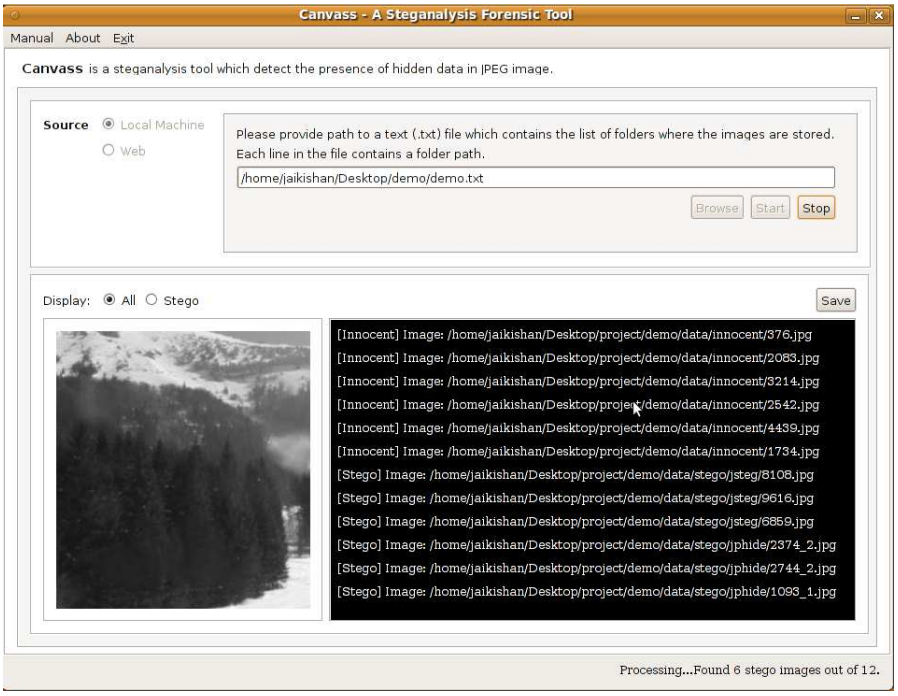


Figure 5.1: Canvass - A software package for steganalysis

- Displays detailed processing information in real time. It shows various information such as steganography algorithm used, time of processing along with other useful information. This is shown in Figure 5.2.
- Provides option to save the processing information at any time. A sample report is shown in 5.3c.
- Ability to stop the processing in between.
- Displays image for visual inspection.
- Ability to run on multiple platforms. In this version, it is currently supported for Windows and *nix systems.

However, the current version only accepts JPEG images with width$\geq$ 250 and height$\geq$ 250. This restriction ensures that features contains enough data to characterize the image as cover or stego. This software will be made available from MFRC, Ames Laboratory for limited distribution to recognized police departments.

5.1 Implementation Details

A mutli-class classifier is implemented in Canvass with $\binom{n}{2}$ binary classifiers where n is the number of classes. In our case, $n = 6$ for cover, jsteg, outguess, f5, steghide and jphide. For an unknown image, features are extracted and passed through all the binary classifiers and assign it the class which appeared maximum time in all the binary classifiers. Model-View-Controller (MVC) architecture has been used to design this software. Due to this reason, it can be easily extended using a different steganalyzer from backend, that is, if additional binary classifiers are added.

Figure 5.2: Canvass - Detailed Screen

5.2 Installer

Canvass comes with a jar excecutable file - Canvass.jar along with several other folders. To run this software, user must have Java installed on their machine. The software can be started by either double clicking on the software or from command line as

java -jar Canvass.jar

It comes with the following folder:

- log: This folder contains various messages generated during the execution of the software. It stores all the error messages under **error** sub folder which can help to debug the application. It also stores the results for every image under **process** sub folder so that results are not lost in case of any unexpected failure of software or system. Finally it also stores a summary of the processing information under **general** sub folder. A sample general and log file is shown in Figure 5.3a and 5.3b.
- external: This folder contains compiled code for feature extraction for Windows and Linux machine. For every image, Canvass calls this code to extract the features from image and pass them to SVM for testing purpose.
- lib: This folder contains necessary Java Swing libraries necessary to support the application.
- data: This folder contains following sub-folders:
  - config: This folder contains various config files, logo of the software and other related files.
  - manual: This folder contains a manual for easy use of the software.
  - steganalyzer: This folder contains SVM model files for different binary SVM's. Files for every binary classifier is stored in sub folder **i_j**. For example **0_1** folder represents SVM model files for binary SVM between cover and jsteg class. We use the following numbering convention for classes: 0 - Cover, 1 - Jsteg, 2- OutGuess, 3 - F5, 4 - Steghide, 5 - JPHide. In the future, if the capability of Canvass needs to be extended to incorporate a new steganographic algorithm, all that is required is to build binary SVM's of the new class with every other existing class.
  - workspace: This folder is used by the software to store any temporary files and images computed during the execution of Canvass.

```
4:45 PM: Number of root folders to be examined: 2
4:45 PM: Local Machine Session starts
****************************************

4:49 PM: JSteg: 3
4:49 PM: OutGuess: 0
4:49 PM: F5: 6
4:49 PM: StegHide: 6
4:49 PM: JPHide: 6
4:49 PM: Done. Found 20 stego images out of 26.
****************************************
```

(a) General Log

```
4:45 PM: Local Machine Session starts
****************************************

4:46 PM: [Innocent] Image: /home/jaikishan/Desktop/demo/
data/innocent/376.jpg
Dimension: 512x512   Size: 48.37 KB
Processing Time: 21.0 s.
4:46 PM: [Innocent] Image: /home/jaikishan/Desktop/demo/
data/innocent/2083.jpg
Dimension: 512x512   Size: 38.5 KB
Processing Time: 7.0 s.
4:46 PM: [Innocent] Image: /home/jaikishan/Desktop/demo/
data/innocent/3214.jpg
Dimension: 512x512   Size: 16.45 KB
Processing Time: 7.0 s.
4:46 PM: [Innocent] Image: /home/jaikishan/Desktop/demo/
data/innocent/2542.jpg
Dimension: 512x512   Size: 31.96 KB
Processing Time: 6.0 s.
4:46 PM: [Innocent] Image: /home/jaikishan/Desktop/demo/
data/innocent/4439.jpg
Dimension: 512x512   Size: 39.17 KB
Processing Time: 6.0 s.
```

(b) Process Log

```
Canvass Report for Steganalysis.
Copyright 2009. MFRC, Ames Laboratory. All Rights
Reserved.
Date: 01.11.09


[Innocent] Image: /home/jaikishan/Desktop/project/
demo/data/innocent/376.jpg
Dimension: 512x512   Size: 48.37 KB
Processing Time: 9.0 s.

[Innocent] Image: /home/jaikishan/Desktop/project/
demo/data/innocent/2083.jpg
Dimension: 512x512   Size: 38.5 KB
Processing Time: 8.0 s.

[Stego] Image: /home/jaikishan/Desktop/project/demo/
data/stego/jsteg/8108.jpg
Dimension: 512x512   Size: 23.41 KB
Processing Time: 8.0 s.
Possible Steganographic algorithm used: JSteg


[Stego] Image: /home/jaikishan/Desktop/project/demo/
data/stego/jphide/4181_3.jpg
Dimension: 512x512   Size: 42.41 KB
Processing Time: 8.0 s.
Possible Steganographic algorithm used: JPHide
```

(c) Sample report

Figure 5.3: (a)General and (b)Process Log file. (c) Sample report

.1 Univariate Detection accuracy tables for Pevný Feature set [53] for BOWS2 Database

		10	**20**	**30**	**40**	**50**	**60**	**70**
Jsteg	Cover	97.56	99.04	99.06	99.32	99.66	99.66	99.78
	0.05 bpc	88.02	96.94	98.14	98.78	98.62	98.70	98.84
	0.10 bpc	99.58	99.60	99.80	99.90	99.92	99.92	99.98
	0.20 bpc	99.96	99.98	99.98	100.00	100.00	100.00	100.00
	0.40 bpc	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Outguess	Cover	97.32	97.78	98.36	98.28	98.24	98.28	98.36
	0.05 bpc	89.64	91.30	94.06	94.40	94.98	94.98	95.32
	0.10 bpc	99.98	99.90	99.96	99.96	99.96	100.00	100.00
	0.20 bpc	100.00	99.98	100.00	100.00	99.98	99.98	100.00
F5	Cover	89.92	90.24	90.40	92.48	92.64	92.96	93.56
	0.05 bpc	41.40	51.76	50.36	54.68	55.20	56.44	58.16
	0.10 bpc	87.04	94.98	94.20	97.52	97.44	97.80	97.94
	0.20 bpc	99.84	100.00	100.00	99.96	99.98	99.98	99.98
	0.40 bpc	99.98	100.00	100.00	100.00	100.00	100.00	100.00
Steghide	Cover	87.78	88.90	90.84	91.20	90.60	90.76	90.62
	0.05 bpc	59.02	62.88	65.62	65.78	65.74	66.64	68.10
	0.10 bpc	76.26	80.98	84.94	85.52	85.66	86.22	86.46
	0.20 bpc	94.46	97.02	98.58	98.64	98.80	98.84	98.78
	0.40 bpc	99.86	99.98	99.96	99.96	99.96	99.98	99.96
JPHide	Cover	97.28	98.02	97.94	98.20	98.22	97.66	98.08
	0.05 bpc	81.05	81.85	81.67	80.39	81.37	82.51	81.11
	0.10 bpc	83.09	84.97	86.46	86.66	87.92	88.98	87.26
	0.20 bpc	90.97	96.23	97.98	97.96	98.54	98.82	98.76
	0.40 bpc	99.50	99.94	99.96	99.96	99.96	99.96	99.94

		80	**90**	**100**	**110**	**120**	**130**	**140**
Jsteg	Cover	99.74	99.72	99.68	99.72	99.52	99.64	99.62
	0.05 bpc	98.98	98.96	98.98	98.86	98.96	99.04	98.92
	0.10 bpc	99.98	99.98	99.96	99.98	99.94	99.98	99.98
	0.20 bpc	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.40 bpc	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Outguess	Cover	98.52	98.38	98.34	98.54	98.42	98.68	98.64
	0.05 bpc	94.90	95.46	95.92	95.70	95.92	95.58	95.18
	0.10 bpc	99.96	100.00	100.00	100.00	100.00	99.98	99.94
	0.20 bpc	100.00	100.00	99.98	100.00	100.00	100.00	99.98
F5	Cover	93.28	93.08	93.46	93.54	93.54	93.22	93.22
	0.05 bpc	58.42	59.24	58.78	58.92	59.02	58.70	59.28
	0.10 bpc	97.76	98.06	98.16	98.16	98.28	98.24	98.32
	0.20 bpc	99.96	100.00	100.00	100.00	100.00	100.00	100.00
	0.40 bpc	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Steghide	Cover	90.58	90.60	91.50	91.86	92.02	92.02	92.00
	0.05 bpc	67.92	68.56	67.74	68.18	67.66	67.00	67.64
	0.10 bpc	86.78	86.88	87.00	87.46	87.12	87.02	87.14
	0.20 bpc	99.02	99.06	98.96	98.90	98.82	98.72	98.76
	0.40 bpc	100.00	100.00	99.98	100.00	99.94	99.94	99.94
JPHide	Cover	98.20	97.96	98.06	97.64	97.74	97.86	97.58
	0.05 bpc	81.29	81.15	81.45	81.37	81.51	81.19	83.16
	0.10 bpc	87.44	88.10	88.62	88.72	89.06	89.08	90.50
	0.20 bpc	98.66	98.46	98.72	98.76	98.74	98.68	98.82
	0.40 bpc	99.90	99.92	99.84	99.78	99.72	99.68	99.64

		150	**160**	**170**	**180**	**190**	**200**	**210**
Jsteg	Cover	99.64	99.66	99.68	99.64	99.56	99.54	99.62
	0.05 bpc	99.12	99.08	99.12	99.10	98.98	99.08	99.12
	0.10 bpc	99.98	99.98	99.98	99.98	99.98	99.98	100.00
	0.20 bpc	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.40 bpc	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Outguess	Cover	98.74	98.68	98.74	98.72	98.58	98.66	98.60
	0.05 bpc	95.42	95.50	95.62	95.72	95.64	95.76	95.50
	0.10 bpc	99.94	99.94	99.96	99.98	99.98	100.00	99.96
	0.20 bpc	99.96	99.94	100.00	100.00	100.00	100.00	100.00
F5	Cover	93.24	92.92	92.80	92.82	92.84	93.02	92.58
	0.05 bpc	59.28	59.16	59.12	58.90	59.14	58.24	58.20
	0.10 bpc	98.36	98.32	98.28	98.24	98.28	98.14	98.18
	0.20 bpc	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.40 bpc	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Steghide	Cover	91.72	91.78	91.74	92.10	92.22	91.78	91.84
	0.05 bpc	67.68	68.04	67.34	67.68	67.54	67.94	68.14
	0.10 bpc	86.72	87.32	86.54	87.24	87.12	86.96	87.10
	0.20 bpc	98.62	98.78	98.70	98.78	98.68	98.72	98.80
	0.40 bpc	99.96	100.00	99.98	99.98	99.98	99.98	99.96
JPHide	Cover	97.52	97.42	97.52	97.66	97.50	97.38	97.36
	0.05 bpc	83.26	83.24	83.18	83.14	82.88	83.26	83.38
	0.10 bpc	91.16	91.14	90.92	91.14	90.74	90.88	91.14
	0.20 bpc	98.78	98.84	98.84	98.82	98.76	98.78	98.72
	0.40 bpc	99.66	99.66	99.64	99.70	99.68	99.68	99.78

		220	**230**	**240**	**250**	**260**	**270**	**274**
Jsteg	Cover	99.60	99.62	99.62	99.54	99.64	99.62	99.64
	0.05 bpc	99.18	99.20	99.18	99.08	99.14	99.14	99.10
	0.10 bpc	100.00	100.00	100.00	99.98	100.00	100.00	100.00
	0.20 bpc	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.40 bpc	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Outguess	Cover	98.72	98.74	98.68	98.82	98.78	98.92	98.78
	0.05 bpc	95.60	95.58	95.32	95.26	95.04	95.48	95.26
	0.10 bpc	99.98	99.96	99.96	99.94	99.92	99.94	99.92
	0.20 bpc	100.00	100.00	100.00	100.00	100.00	99.98	99.98
F5	Cover	93.18	92.48	93.26	92.98	92.14	92.70	91.82
	0.05 bpc	58.00	58.76	57.82	57.48	59.08	58.16	59.64
	0.10 bpc	97.84	97.78	97.88	97.88	97.78	97.96	97.78
	0.20 bpc	100.00	100.00	100.00	100.00	99.98	99.98	99.96
	0.40 bpc	100.00	100.00	100.00	99.98	99.98	100.00	99.98
Steghide	Cover	91.46	91.42	91.44	91.34	91.52	91.52	91.78
	0.05 bpc	67.68	67.70	67.86	67.72	70.06	69.68	69.52
	0.10 bpc	86.78	86.86	87.28	86.94	88.12	88.00	87.96
	0.20 bpc	98.66	98.70	98.70	98.64	98.82	98.72	98.72
	0.40 bpc	99.96	99.98	99.94	99.94	99.96	99.98	99.96
JPHide	Cover	97.38	97.54	97.64	97.70	97.60	97.56	97.50
	0.05 bpc	83.44	82.17	82.51	82.53	82.39	82.51	82.51
	0.10 bpc	91.06	90.50	90.64	90.56	90.50	90.66	90.70
	0.20 bpc	98.72	98.60	98.64	98.58	98.60	98.68	98.70
	0.40 bpc	99.74	99.82	99.84	99.80	99.82	99.82	99.82

.2 Multivariate Detection accuracy tables for Pevný Feature set [53] for BOWS2 Database

		10	**20**	**30**	**40**	**50**	**60**	**70**
Jsteg	Cover	99.20	99.12	99.18	99.56	99.54	99.54	99.54
	0.05 bpc	97.64	97.86	98.96	99.02	99.04	99.04	99.28
	0.10 bpc	99.92	99.88	99.98	99.98	99.96	100.00	100.00
	0.20 bpc	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.40 bpc	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Outguess	Cover	97.00	97.98	98.44	98.42	98.44	98.64	98.58
	0.05 bpc	89.98	92.60	94.10	94.16	94.30	94.14	94.30
	0.10 bpc	99.98	99.98	99.96	99.96	100.00	100.00	99.98
	0.20 bpc	100.00	100.00	100.00	100.00	100.00	100.00	100.00
F5	Cover	91.92	93.16	92.78	92.62	92.62	92.68	92.76
	0.05 bpc	35.78	40.54	43.90	49.62	49.58	51.32	51.04
	0.10 bpc	84.32	90.88	92.18	96.02	96.08	96.70	96.54
	0.20 bpc	99.92	99.98	99.96	99.98	100.00	100.00	100.00
	0.40 bpc	99.98	100.00	100.00	100.00	100.00	100.00	100.00
Steghide	Cover	88.70	90.82	91.94	91.88	92.26	91.86	91.86
	0.05 bpc	64.74	68.26	67.48	67.74	68.56	68.22	68.38
	0.10 bpc	83.12	86.62	86.24	86.88	87.34	87.66	87.80
	0.20 bpc	97.46	98.74	98.68	98.70	98.72	98.86	98.76
	0.40 bpc	100.00	99.98	99.96	99.98	99.98	99.98	99.98
JPHide	Cover	97.98	97.80	97.24	97.54	97.06	96.74	97.50
	0.05 bpc	82.61	83.60	83.16	82.74	83.52	83.94	83.44
	0.10 bpc	86.46	87.46	88.26	88.60	90.22	91.28	90.68
	0.20 bpc	97.36	97.56	97.22	98.10	98.78	99.02	98.82
	0.40 bpc	99.86	99.94	99.88	99.94	99.96	99.94	99.84

		80	**90**	**100**	**110**	**120**	**130**	**140**
Jsteg	Cover	99.60	99.56	99.62	99.58	99.60	99.66	99.66
	0.05 bpc	99.34	99.28	99.34	99.40	99.34	99.30	99.18
	0.10 bpc	100.00	100.00	100.00	100.00	100.00	100.00	99.98
	0.20 bpc	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.40 bpc	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Outguess	Cover	98.70	98.48	98.56	98.80	98.70	98.72	98.72
	0.05 bpc	95.32	95.28	95.36	95.16	95.30	95.22	95.02
	0.10 bpc	100.00	99.94	100.00	99.96	99.98	100.00	100.00
	0.20 bpc	100.00	100.00	100.00	100.00	100.00	100.00	100.00
F5	Cover	93.08	93.16	93.02	92.82	92.76	92.78	93.12
	0.05 bpc	51.10	51.16	50.88	51.84	51.90	52.10	56.20
	0.10 bpc	96.56	96.60	96.56	96.58	96.46	96.44	97.92
	0.20 bpc	100.00	100.00	100.00	100.00	99.98	99.98	100.00
	0.40 bpc	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Steghide	Cover	92.10	91.78	91.72	91.54	91.38	91.40	91.52
	0.05 bpc	68.66	68.58	69.14	69.12	69.04	68.68	68.88
	0.10 bpc	87.96	87.72	88.08	87.98	87.92	87.74	87.66
	0.20 bpc	98.74	98.82	98.88	98.84	98.86	98.70	98.80
	0.40 bpc	99.98	99.98	99.98	99.98	99.98	99.98	99.98
JPHide	Cover	97.32	97.36	97.40	97.52	97.56	97.82	97.52
	0.05 bpc	83.26	83.44	83.26	83.16	83.20	82.49	83.02
	0.10 bpc	90.74	91.00	90.52	90.66	90.54	90.38	91.08
	0.20 bpc	98.74	98.70	98.62	98.78	98.72	98.66	98.78
	0.40 bpc	99.78	99.78	99.72	99.74	99.72	99.80	99.76

		150	**160**	**170**	**180**	**190**	**200**	**210**
Jsteg	Cover	99.68	99.60	99.48	99.48	99.52	99.50	99.56
	0.05 bpc	99.22	99.26	99.12	99.30	99.30	99.24	99.12
	0.10 bpc	99.98	99.98	99.96	99.96	99.98	99.98	99.96
	0.20 bpc	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.40 bpc	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Outguess	Cover	98.56	98.70	98.82	98.74	98.74	98.72	98.66
	0.05 bpc	95.30	95.24	95.32	95.12	95.26	95.30	95.28
	0.10 bpc	100.00	99.96	100.00	99.98	99.98	99.98	100.00
	0.20 bpc	100.00	100.00	100.00	100.00	100.00	100.00	100.00
F5	Cover	92.92	91.74	93.22	92.42	92.38	92.46	92.46
	0.05 bpc	56.52	57.54	53.50	57.78	57.70	58.34	58.42
	0.10 bpc	97.84	97.68	97.26	97.96	98.12	98.16	98.24
	0.20 bpc	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.40 bpc	100.00	100.00	99.98	99.98	99.98	99.98	100.00
Steghide	Cover	91.68	91.56	91.68	91.36	91.50	91.76	91.72
	0.05 bpc	69.08	69.88	69.46	68.96	68.72	69.34	69.66
	0.10 bpc	87.82	88.14	88.18	87.40	87.24	88.22	87.96
	0.20 bpc	98.74	98.84	98.86	98.72	98.82	98.78	98.66
	0.40 bpc	99.98	100.00	100.00	99.98	100.00	100.00	99.98
JPHide	Cover	97.52	97.30	97.40	97.70	97.70	97.64	97.52
	0.05 bpc	82.86	82.92	83.06	82.39	82.03	82.25	82.55
	0.10 bpc	90.88	91.00	91.20	90.56	90.32	90.50	90.54
	0.20 bpc	98.76	98.72	98.62	98.78	98.64	98.74	98.70
	0.40 bpc	99.78	99.74	99.74	99.80	99.74	99.86	99.78

		220	**230**	**240**	**250**	**260**	**270**	**274**
Jsteg	Cover	99.52	99.56	99.64	99.60	99.60	99.58	99.64
	0.05 bpc	99.20	99.08	99.12	99.06	99.02	99.00	99.10
	0.10 bpc	99.98	99.98	100.00	100.00	99.98	100.00	100.00
	0.20 bpc	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.40 bpc	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Outguess	Cover	98.66	98.64	98.74	98.70	98.90	98.92	98.78
	0.05 bpc	95.34	95.34	95.16	95.58	95.48	95.52	95.26
	0.10 bpc	100.00	100.00	99.96	100.00	99.94	99.92	99.92
	0.20 bpc	100.00	100.00	100.00	100.00	100.00	99.98	99.98
F5	Cover	92.54	92.46	92.34	92.44	92.20	91.62	91.82
	0.05 bpc	58.20	58.66	58.50	58.52	59.18	59.50	59.64
	0.10 bpc	98.10	98.02	97.96	97.80	97.96	97.78	97.78
	0.20 bpc	99.98	100.00	100.00	99.98	99.98	99.96	99.96
	0.40 bpc	99.98	99.98	100.00	100.00	100.00	99.98	99.98
Steghide	Cover	91.80	91.68	91.66	91.78	91.62	91.42	91.78
	0.05 bpc	69.00	68.66	69.04	69.46	69.86	69.66	69.52
	0.10 bpc	88.00	87.88	87.80	88.24	88.12	88.06	87.96
	0.20 bpc	98.84	98.80	98.76	98.80	98.64	98.70	98.72
	0.40 bpc	99.96	99.94	99.94	99.96	99.96	99.96	99.96
JPHide	Cover	97.60	97.50	97.48	97.52	97.52	97.56	97.50
	0.05 bpc	82.45	82.53	82.49	82.43	82.57	82.57	82.51
	0.10 bpc	90.50	90.64	90.56	90.44	90.46	90.74	90.70
	0.20 bpc	98.76	98.70	98.68	98.42	98.68	98.76	98.70
	0.40 bpc	99.82	99.82	99.80	99.80	99.78	99.82	99.82

Bibliography

[1] Electronic crime scene investigation: A guide for first responders, second edition, April 2008. National Institute of Justice. Available online at `http://www.ojp.usdoj.gov/nij/publications/ecrime-guide-219941/chapter7-219941.pdf`.

[2] Steghide. Available online at `http://steghide.sourceforge.net/`.

[3] Jphide&seek. Available online at `http://linux01.gwdg.de/~alatham/stego.html`.

[4] Jsteg. Available online at `http://zooid.org/paul/crypto/jsteg/`.

[5] Natural resources conservation service photo gallery. Available online at `http://photogallery.nrcs.usda.gov`.

[6] Stegoarchive. Available online at `http://www.stegoarchive.com/`.

[7] Msu. Available online at `http://www.compression.ru/video/`.

[8] Corel stock photo library 3.

[9] K. Abend, T. Harley, and L. Kanal. Classification of binary random patterns. *IEEE Transactions on Information Theory*, volume 11(4):538–544, 1965.

[10] R. Anderson and F. Petitcolas. On the limits of steganography. *IEEE Journal on Selected Areas in Communications*, 16(4):474–481, May 1998.

[11] T. Aura. Practical invisibility in digital communication. In *Lecture Notes in Computer Science, vol.1174*, pages 265–278. SpringerVerlag, 1996.

[12] I. Avcibas, N. Memon, and B. Sankur. Steganalysis of watermarking techniques using image quality metrics. In *Security and Watermarking of Multimedia Contents VI*, volume 5306, pages 83–97. SPIE, 2001.

[13] C. J. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[14] C. Cachin. An information-theoretic model for steganography. In *Information Hiding*, volume 1525 of *Lecture Notes in Computer Science*, pages 306–318. Springer Berlin / Heidelberg, 1998.

[15] A. Caprihan, G. D. Pearlson, and V. D. Calhoun. Application of principal component analysis to distinguish patients with schizophrenia from healthy controls based on fractional anisotropy measurements. *Neuroimage*, 42:675–682, 2008.

[16] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[17] W.-C. Chang. On using principal components before separating a mixture of two multivariate normal distributions. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 32(3):267–275, 1983.

[18] C. Chen and Y. Q. Shi. Jpeg image steganalysis utilizing both intrablock and interblock correlations. In *IEEE International Symposium on Circuits and Systems*, pages 3029–3032, May 2008.

[19] C. Chen and Y. Q. Shi. Jpeg image steganalysis utilizing both intrablock and interblock correlations. In *IEEE International Symposium on Circuits and Systems*, pages 3029–3032, May 2008.

[20] J. Cheng, A. Kot, and S. Rahardja. Steganalysis of binary cartoon image using distortion measure. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 2, pages II–61–II–64, 2007.

[21] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker. *Digital Watermarking and Steganography*. Elsevier, 2007.

[22] N. Cressie and J. Davidson. Image analysis with partially ordered markov models. *Computational Statistics and Data Analysis*, 29:1–26, 1998.

[23] J. L. Davidson, N. Cressie, and X. Hua. Texture synthesis and pattern recognition for partially ordered markov models. *Pattern Recognition*, 32(9):1475–1505, 1999.

[24] G. Doerr. Image database for steganalysis studies [online]. `http://www.cs.ucl.ac.uk/staff/G.Doerr/`, 2007.

[25] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley-Interscience, 2001.

[26] J. Fridrich. Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes. In *Information Hiding*, volume 3200 of *Lecture Notes in Computer Science*, pages 67–81. Springer Berlin / Heidelberg, 2005.

[27] D. Fu, Y. Shi, D. Zou, and G. Xuan. Jpeg steganalysis using empirical transition matrix in block dct domain. In *IEEE 8th Workshop on Multimedia Signal Processing*, pages 310–313, October 2006.

[28] M. Goljan, J. Fridrich, and T. Holotyak. New blind steganalysis and its implications. In E. J. D. III and P. W. Wong, editors, *Security, Steganography, and Watermarking of Multimedia Contents VIII*, volume 6072, page 607201. SPIE, 2006.

[29] B. Haasdonk and E. Pkalska. Classification with kernel mahalanobis distance classifiers. In *Advances in Data Analysis, Data Handling and Business Intelligence*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 351–361. Springer Berlin Heidelberg, 2009.

[30] J. J. Harmsen and W. A. Pearlman. Steganalysis of additive-noise modelable information hiding. In E. J. D. III and P. W. Wong, editors, *Security and Watermarking of Multimedia Contents V*, volume 5020, pages 131–142. SPIE, 2003.

[31] J. Helterbrand, M. Cressie, and J. Davidson. Optimal closed boundary identification in gray-scale imagery. *Journal of Mathematical Imaging and Vision*, volume 5(3):179–206, 1995.

[32] T. Holotyak, J. Fridrich, and S. Voloshynovskiy. Blind statistical steganalysis of additive steganography using wavelet higher order statistics. In *Communications and Multimedia Security*, volume 3677 of *Lecture Notes in Computer Science*, pages 273–274. Springer Berlin / Heidelberg, 2005.

[33] F. Huang, J. Huang, and B. Li. Universal jpeg steganalysis based on microscopic and macroscopic calibration. In *15th IEEE International Conference on Image Processing*, pages 2068–2071, October 2008.

[34] A. K. Jain, R. P. Duin, and J. Mao. Statistical pattern recognition: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, January 2000.

[35] E. John, L. Prichep, J. Fridman, and P. Easton. Neurometrics: computer-assisted differential diagnosis of brain dysfunctions. *Science*, 239(4836):162–169, 1988.

[36] J. Kelly. Terrorist instructions hidden online. USA Today, February 2001.

[37] A. Ker. Quantitive evaluation of pairs and rs steganalysis. In *Security, Steganography, and Watermarking of Multimedia Contents III*, volume 3677, pages 273–274. SPIE, 2004.

[38] A. Ker. Steganalysis of lsb matching in grayscale images. *IEEE Signal Processing Letters*, 12(6):441–444, June 2005.

[39] A. D. Ker and I. Lubenko. Feature reduction and payload location with wam steganalysis. In *Media Forensics and Security*, volume 7254, page 72540A. SPIE, 2009.

[40] R. Kindermann and J. L. Snell. *Markov Random Fields and Their Applications*. AMS, 1980.

[41] J. Kodovský and J. Fridrich. Calibration revisited. In *MM&Sec '09: Proceedings of the 11th ACM workshop on Multimedia and security*, pages 63–74, New York, NY, USA, 2009. ACM.

[42] G. Kolata. Veiled messages of terrorists may lurk in cyberspace. USA Today, October 2001.

[43] Q. Liu, A. H. Sung, Z. Chen, and J. Xu. Feature mining and pattern classification for steganalysis of lsb matching steganography in grayscale images. *Pattern Recognition*, 41(1):56–66, 2008.

[44] X.-Y. Luo, D.-S. Wang, P. Wang, and F.-L. Liu. A review on blind detection for image steganography. *Signal Processing*, 88(9):2138–2157, 2008.

[45] S. Lyu and H. Farid. Steganalysis using higher-order image statistics. *IEEE Transactions on Information Forensics and Security*, 1(1):111–119, March 2006.

[46] B. F. Manly. *Multivariate Statistical Methods: A Primer, Third Edition*. Chapman and Hall Ltd, 2004.

[47] Y. Miche, P. Bas, A. Lendasse, C. Jutten, and O. Simula. Reliable steganalysis using a minimum set of samples and features. *EURASIP Journal on Information Security*, 2009(901381):13, 2009.

[48] Y. Miche, B. Roue, A. Lendasse, and P. B. and. A feature selection methodology for steganalysis. In *Multimedia Content Representation, Classification and Security*, volume 4105 of *Lecture Notes in Computer Science*, pages 49–56. Springer Berlin / Heidelberg, 2006.

[49] J. Mielikainen. Lsb matching revisited. *Signal Processing Letters, IEEE*, 13(5):285–287, May 2006.

[50] D. V. Nguyen and D. M. Rocke. On partial least squares dimension reduction for microarray-based classification: a simulation study. *Computational Statistics & Data Analysis*, 46(3):407–425, 2004.

[51] T. Pevný and J. Fridrich. Towards multi-class blind steganalyzer for JPEG images. In *Digital Watermarking*, volume 3710 of *Lecture Notes in Computer Science*, pages 39–53. Springer Berlin / Heidelberg, 2005.

[52] T. Pevný and J. Fridrich. Multi-class blind steganalysis for jpeg images. In E. J. D. III and P. W. Wong, editors, *Security, Steganography, and Watermarking of Multimedia Contents VIII*, volume 6072, page 60720O. SPIE, 2006.

[53] T. Pevný and J. Fridrich. Merging markov and dct features for multi-class jpeg steganalysis. In E. J. D. III and P. W. Wong, editors, *Security, Steganography, and Watermarking of Multimedia Contents IX*, volume 6505. SPIE, 2007.

[54] N. Provos. Defending against statistical steganalysis. In *Proceedings of the 10th conference on USENIX Security Symposium*, pages 24–24, Berkeley, CA, USA, 2001. USENIX Association.

[55] N. Provos. Scanning usenet for steganography. `http://www.citi.umich.edu/u/provos/stego/usenet.php`, 2001.

[56] N. Provos. Steganography detection with stegdetect. `http://www.outguess.org/detection.php`, 2004.

[57] N. Provos and P. Honeyman. Detecting steganographic content on the internet. Technical report, In ISOC NDSS02, 2001.

[58] J. Qin, X. Sun, X. Xiang, and C. Niu. Principal feature selection and fusion method for image steganalysis. *Journal of Electronic Imaging*, 18(3):033009, 2009.

[59] J. W. Robinson and R. S. Hoffmann. Geographical and interspecific cranial variation in big-eared ground squirrels (spermophilus): A multivariate study. *Systematic Zoology*, 24(1):79–88, March 1975.

[60] G. Schaefer and M. Stich. Ucid - an uncompressed colour image database. *Proc. SPIE, Storage and Retrieval Methods and Applications for Multimedia*, pages 472–480, 2004.

[61] Y. Q. Shi, C. Chen, and W. Chen. A markov process based approach to effective attacking JPEG steganography. In *Information Hiding*, volume 4437 of *Lecture Notes in Computer Science*, pages 249–264. Springer Berlin / Heidelberg, 2007.

[62] J. Shlens. A tutorial on principal component analysis. `http://www.snl.salk.edu/~shlens/pub/notes/pca.pdf`, 2009.

[63] R. Wang and W. Zhao. Improvement of tnd steganalysis based on image complexity. In *9th International Conference on Signal Processing*, pages 2777–2781, October 2008.

[64] Y. Wang and P. Moulin. Optimized feature extraction for learning-based image steganalysis. *IEEE Transactions on Information Forensics and Security*, 2(1):31–45, March 2007.

[65] G. Weimann. How modern terrorism uses the internet. Special Report 116, United States Institute of Peace, March 2004.

[66] A. Westfeld. Reproducible signal processing (bows2 challenge image database, public).

[67] A. Westfeld. F5: A steganographic algorithm. In *Information Hiding*, volume 2137 of *Lecture Notes in Computer Science*, pages 289–302. Springer Berlin / Heidelberg, 2001.

[68] G. Xuan, X. Cui, Y. Shi, W. Chen, X. Tong, and C. Huang. Jpeg steganalysis based on classwise non-principal components analysis and multi-directional markov model. In *IEEE International Conference on Multimedia and Expo.*, pages 903–906, July 2007.

[69] G. Xuan, X. Zhu, P. Chai, Z. Zhang, Y. Q. Shi, and D. Fu. Feature selection based on the bhattacharyya distance. *International Conference on Pattern Recognition*, 4:957, 2006.

[70] J. Zhang, I. Cox, and G. Doerr. Steganalysis for lsb matching in images with high-frequency noise. In *IEEE 9th Workshop on Multimedia Signal Processing, 2007*, pages 385–388, October 2007.