Kernel Fisher Discriminant for Steganalysis of JPEG Hiding Methods

Article *in* Proceedings of SPIE - The International Society for Optical Engineering · February 2004

DOI: 10.1117/12.525996 · Source: CiteSeer

Jeremiah J. Harmsen
Google Inc.
4 PUBLICATIONS 475 CITATIONS

SEE PROFILE

William Pearlman
Rensselaer Polytechnic Institute
241 PUBLICATIONS 12,052 CITATIONS

SEE PROFILE

Kernel Fisher Discriminant for Steganalysis of JPEG Hiding Methods

Jeremiah J. Harmsen[a] and William A. Pearlman[a]

[a]Center for Image Processing Research,
Electrical Computer and Systems Engineering Department,
Rensselaer Polytechnic Institute, Troy, NY

ABSTRACT

The use of kernel Fisher discriminants is used to detect the presence of JPEG based hiding methods. The feature vector for the kernel discriminant is constructed from the quantized DCT coefficient indices. Using methods developed in kernel theory a classifier is trained in a high dimensional feature space which is capable of discriminating original from stegoimages. The algorithm is tested on the F5 hiding method.

Keywords: Steganalysis, steganography, kernel methods, fisher linear discriminant

1. INTRODUCTION

A fundamental goal of steganalysis is to detect the presence of covert communication. Typically the accuracy of such methods is considered with complexity being explored second, if at all. The volume of information transfered over channels such as the internet poses a major obstacle on the real world efficacy of detection schemes.

A significant amount of potential steganographic traffic is in the form of coded media such as JPEG images. The JPEG compression and decompression process is shown in Figure 1. A number of hiding methods use the quantized DCT coefficient indices for embedding, such as the F5 algorithm.[1] To detect this type of hiding many detection schemes require the processing of a test signal in the spatial domain. As shown by the top path in Figure 2, the quantized DCT coefficient indices are recovered from the bitstream after entropy decoding. These indices are used to find the quantized DCT coefficients, which then are projected into the spatial domain using the inverse discrete cosine transform (IDCT). In the spatial domain a number of signal processing operations, such as wavelet decompositions,[2] are used to extract features. These features may then be used in a classifier.

This paper address the complexity issue by implementing a detection scheme using only the indices of the quantized DCT coefficients in JPEG images. The processing sequence is shown as the bottom path in Figure 2. As can be seen, the costly signal processing steps in typical detection are circumvented. This allows for the system to process images very quickly.

Figure 1. JPEG Compression/Decompression

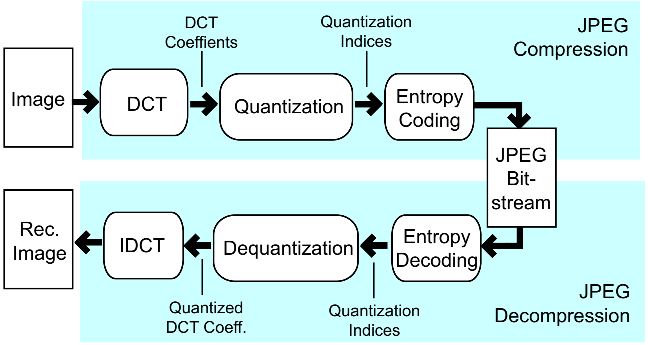
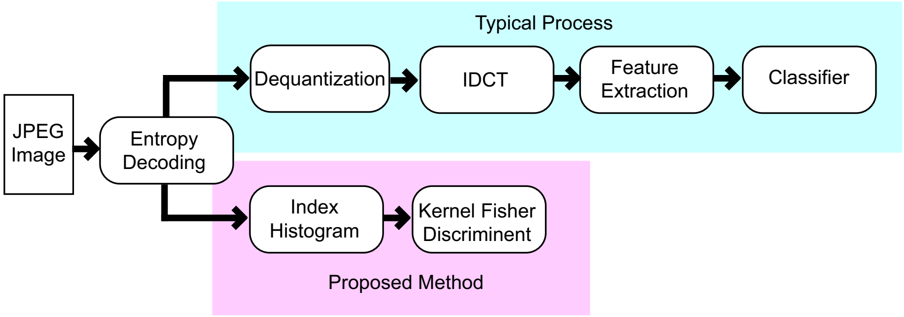


Figure 2. Classification Process



As practicality is a focus of this work, the F5 hiding algorithm has been chosen as a representative hiding method. The foremost reason for focusing on F5 is its use of the prevalent JPEG medium. In addition F5 has been designed to provide a high stealth, and is representative of the current state of the art in high capacity steganographic tools.

2. KERNEL METHODS

2.1. Kernel Methods

Kernel methods[3] have seen a great amount of use in the field of pattern recognition, most prominently in Support Vector Machines.[4] Kernel methods operate in a high (possibly infinite) dimensional space called feature space, F. Inputs are mapped to this space by the mapping function, $\mathbf{\Phi} : \Re^N \rightarrow F$. To make use of the data in feature space it is necessary to define a dot product between two vectors in feature space. This is the so called kernel,

$$\mathcal{K}\left(\mathbf{x}, \mathbf{y}\right) = \left\langle \mathbf{\Phi}\left(\mathbf{x}\right), \mathbf{\Phi}\left(\mathbf{y}\right)\right\rangle . \tag{1}$$

If a kernel satisfies Mercer's Theorem, it is equivalent to an innerproduct in some feature space.[4] Thus by defining a valid kernel, an underlying feature space is also created. For example, the exponential kernel,

$$\mathcal{K}\left(\mathbf{x}, \mathbf{y}\right) = \exp\left(-\frac{1}{2\sigma^2}\left|\left|\mathbf{x} - \mathbf{y}\right|\right|^2\right), \tag{2}$$

defines a mapping, $\mathbf{\Phi}$, into an infinite dimensional feature space. Note that here the actual value of $\mathbf{\Phi}\left(\mathbf{x}\right)$ cannot be calculated, instead the relationship between two points $\mathbf{\Phi}\left(\mathbf{x}\right), \mathbf{\Phi}\left(\mathbf{y}\right) \in F$ may only be explored through their innerproduct, $\left\langle \mathbf{\Phi}\left(\mathbf{x}\right), \mathbf{\Phi}\left(\mathbf{y}\right)\right\rangle = \mathcal{K}\left(\mathbf{x}, \mathbf{y}\right)$.

Intuitively, the utility of the mapping is that non-linear relationships in the input-space, $\Re^N$, become linear when mapped into $\mathcal{F}$. These linearities may then be exploited using subspace methods such as principle component analysis, canonical correlation cnalysis, and Fisher analysis.

2.2. Fisher Linear Discriminant

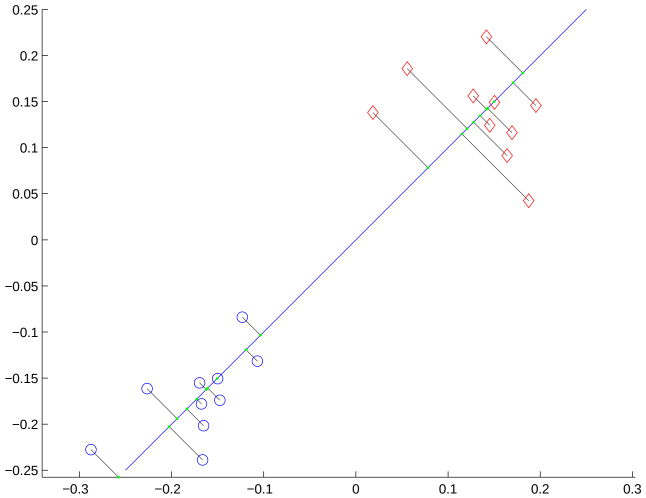
The Fisher Linear Discriminant (FLD) is a method that uses labeled training exemplars to create a classifier. The FLD operates by finding a one-dimensional subspace such that the projection of the labeled training data onto this subspace accomplishes two goals:

1. Maximization of the interclass means
2. Minimization of the intraclass variance

This subspace, $\mathbf{d}$ is called the discriminant.

Figure (3) shows the FLD for a set of toy data in $\Re^2$ with $\diamond$ representing Class 1 and $\circ$ representing Class 2. The subspace that accomplishes these goals is found by the solution of an eigenvalue problem. To classify a test vector, $\mathbf{y}$, the projection of $\mathbf{y}$ along $\mathbf{d}$ is used. If $\mathbf{d}$ is normalized ($||\mathbf{d}|| = 1$), this projection reduces to an inner-product. Using the discriminant $\mathbf{d}$ the decision rule for classifying an unknown $\mathbf{y}$ is,

Figure 3. Fisher Linear Discriminant



- $\langle \mathbf{y}, \mathbf{d} \rangle - b \geq 0 \Rightarrow \mathbf{y} \in$ Class 1
- $\langle \mathbf{y}, \mathbf{d} \rangle - b < 0 \Rightarrow \mathbf{y} \in$ Class 2

Here b is a constant to account for the projected means of each class.

2.3. Kernel Fisher Discriminant

The Kernel Fisher Discriminant (KFD) is a nonlinear extension of the Fisher Linear Discriminant into feature space. A more complete derivation of the Kernel Fisher Discriminant appears in Appendix A.

As with the Fisher Linear Discriminant, the goal is to find the discriminant vector $\mathbf{w} \in \mathcal{F}$ that maximizes the difference between projected class means and minimizes the projected intraclass variance. Let χ denote the set of ℓ labeled training examples, where $\chi = \{\mathbf{x}_1, \ldots, \mathbf{x}_\ell\}$. As feature space is potentially of high dimensionality, the discriminant is restricted to lie in a subspace spanned by the images of the training points,

$$\mathbf{w} = \sum_{\mathbf{x} \in \chi} \alpha_{\mathbf{x}} \mathbf{\Phi}\left(\mathbf{x}\right), \tag{3}$$

where again $||\mathbf{w}|| = 1$.

As in the linear case, to classify a new point, $\mathbf{y}$ we must find the sign of the projection of $\mathbf{\Phi}\left(\mathbf{y}\right)$ onto the subspace defined by $\mathbf{w}$. Since $||\mathbf{w} = 1||$ this projection is the inner-product, $\langle \mathbf{w}, \mathbf{\Phi}\left(\mathbf{y}\right) \rangle$. Using the linearity of the inner products we have,

$$\langle \mathbf{w}, \mathbf{\Phi}\left(\mathbf{y}\right) \rangle = \left\langle \sum_{\mathbf{x} \in \chi} \alpha_{\mathbf{x}} \mathbf{\Phi}\left(\mathbf{x}\right), \mathbf{\Phi}\left(\mathbf{y}\right) \right\rangle = \sum_{\mathbf{x} \in \chi} \alpha_{\mathbf{x}} \left\langle \mathbf{\Phi}\left(\mathbf{x}\right), \mathbf{\Phi}\left(\mathbf{y}\right) \right\rangle = \sum_{\mathbf{x} \in \chi} \alpha_{\mathbf{x}} \mathcal{K}\left(\mathbf{x}, \mathbf{y}\right).$$

This makes the classification function,

- $\sum_{\mathbf{x} \in \chi} \alpha_{\mathbf{x}} \mathcal{K}\left(\mathbf{x}, \mathbf{y}\right) - b \geq 0 \Rightarrow \mathbf{y} \in$ Class 1
- $\sum_{\mathbf{x} \in \chi} \alpha_{\mathbf{x}} \mathcal{K}\left(\mathbf{x}, \mathbf{y}\right) - b < 0 \Rightarrow \mathbf{y} \in$ Class 2

Again, b accounts for the projected means of each class.

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

Table 1. Coefficient Order

$\mathcal{C}_i = \{i\},\ i \in \{1, \ldots, 10\}$
$\mathcal{C}_{11} = \{11, 12\}$
$\mathcal{C}_{12} = \{13, 14\}$
$\mathcal{C}_{13} = \{15, 16\}$
$\mathcal{C}_{14} = \{17, 18\}$
$\mathcal{C}_{15} = \{19, 20\}$
$\mathcal{C}_{16} = \{21, 22, 23\}$
$\mathcal{C}_{17} = \{24, 25, 26\}$
$\mathcal{C}_{18} = \{27, \ldots, 63\}$

Table 2. Frequency Partition

2.3.1. KFD and Kernel Least Squares

It is well known the Fisher linear discriminant is equivalent to a least squares regression onto the class labels of ±1. This has also been shown to extend to feature space.[5, 6] This allows the use of both the existing theory and tools used in kernel least squares[7] to perform kernel Fisher discriminant analysis.

3. JPEG COEFFICIENTS

3.1. F5 Hiding Algorithm

The F5 algorithm[1] hides data by modifying the quantized AC DCT coefficient indices. Rather than overwriting the least significant bits (as in the program JSteg), the method decreases the absolute value of the indices (skipping indices with a value of 0). This preserves the symmetric and monotonic nature of the index histograms. The approach was designed to discourage detection with the χ^2 attack.[8]

3.2. Quantized Coefficient Index Histogram

For this discussion it is assumed the JPEG compression uses a uniform quantization based on the quantization step sizes for each element of the DCT. These step sizes are denoted Δ_k and defined for $k = 0, \ldots, 63$ for an 8×8 DCT. Thus the index, i, for the kth DCT coefficient, c, is found as $i = \lfloor c/\Delta_k \rfloor$. In decompression, the index is used in the lookup table f_k to find the quantized coefficient $\hat{c}$. The quantized coefficient is recovered as $\hat{c} = f_k(i) = \Delta_k i$.

As the F5 algorithm alters the coefficient indices, a natural feature to use is the histogram of the indices. This will improve speed over current methods in that the image does not need to be fully reconstructed for classification.

The histogram of the indices of an individual DCT element is denoted $h_k\,[d]$ where $k \in \{0, \ldots, 63\}$ corresponds to the frequency component of an 8×8 DCT, pictured in Table 1.

As a consequence of quantization, the expected number of nonzero coefficients decreases as the frequency indices increase. To make use of these higher frequency coefficients, groupings of frequencies are created. By combining the statistics of a group of high frequency coefficients, the goal is to create a statistically meaningful feature for the classifier.

We use the index histogram over $\mathcal{C}$ defined as,

$$h_{\mathcal{C}}\,[d] = \sum_{k \in \mathcal{C}} h_k[d], \tag{4}$$

with $\mathcal{C} \subseteq \{0, \ldots, 63\}$ for an 8×8 DCT. This is simply the number of indices equal to d for the DCT frequencies in $\mathcal{C}$. For example $h_{\mathcal{C}}\,[-2]$ with $\mathcal{C} = \{0, \ldots, 63\}$ would be the total number of quantized coefficient indices with a value of -2 and $h_{\mathcal{C}}\,[3]$ with $\mathcal{C} = \{0\}$ is the total number of DC indices with value 3.

This creates the issue of how to partition the frequency components to make use of the higher frequency components. Specifically we find $\mathcal{C}_i$ for $i \in I$ with $\bigcap_{i \in I} \mathcal{C}_i = \{\phi\}$ and $\bigcup_{i \in I} \mathcal{C}_i = \{1, \ldots, 63\}$. Thus, we divide the 63* elements into $|I|$ sets,

$$0 \underbrace{1}_{\mathcal{C}_1} \underbrace{2}_{\mathcal{C}_2} \underbrace{3}_{\mathcal{C}_3} \underbrace{4\ 5}_{\mathcal{C}_4} \underbrace{6\ 7}_{\mathcal{C}_5} \underbrace{8\ 9\ 10}_{\mathcal{C}_6} \ \underset{\cdots}{\cdots}\ \underbrace{27\ \cdots\ 63}_{\mathcal{C}_{|I|}}.$$

One way to partition the components is to use the constraint:

$$\sum_{s \in \mathcal{S}} h_{\mathcal{C}_j}[s] \approx \sum_{s \in \mathcal{S}} h_{\mathcal{C}_1}[s], \ \forall j \neq 1 \tag{5}$$

with $\mathcal{C}_1 = \{1\}$ and $\mathcal{S} = \{-d, \ldots, -1, 1, \ldots, d\}$ where $\{-d, \ldots, d\}$ covers the values of all possible indices. That is, we would like each $h_{\mathcal{C}_j}[s]$ to have approximately the same number of non-zero indices as are in the 1st DCT element.

As the number of nonzero indices is directly related to the quality factor, the quality factor will in turn effect the partition. The coefficient division for a number of test images with a quality factor of 80 are shown in Table 2.

The feature vector is a combination of the histograms for a set frequency grouping. Since the AC histograms are approximately symmetric and zero mean,[9] we will use the values $-p, \ldots, p$. Note that p is typically a small value ($p = 4$) as the number of indices with large magnitudes is very low.

The feature vector becomes,

$$\mathbf{x} = \left[h_{\mathcal{C}_1}[-p], \ldots, h_{\mathcal{C}_1}[0], \ldots, h_{\mathcal{C}_1}[p], \ldots, h_{\mathcal{C}_{|I|}}[-p], \ldots, h_{\mathcal{C}_{|I|}}[0], \ldots, h_{\mathcal{C}_{|I|}}[p]\right]^\top. \tag{6}$$

This feature vector will be used in the classifier built in Section 4.2.

3.3. F5 Spatial Steganalysis

The F5 algorithm has been analyzed in the method presented by Fridrich,[10] reviewed in Appendix B. In this method the image is first reconstructed in the spatial domain then cropped by removing four rows from the top of the image and four columns from the left. The cropping serves to break the original structure of the DCT blocks. The cropped image is then recompressed using the same quantization tables. It has been shown that the quantized DCT coefficient index histograms of the cropped-image are approximately equal to those of the image before embedding. By comparing the histograms of the indices before and after the spatial cropping, the relative number of modifications (and thus the size of the message) may be estimated.

In the process of estimating the message size, an approximation of the probability that an index in the kth DCT element is altered is calculated. This value is denoted β_k and will be used as a feature in the spatial classifier developed in Section 4.3.

4. EXPERIMENTS

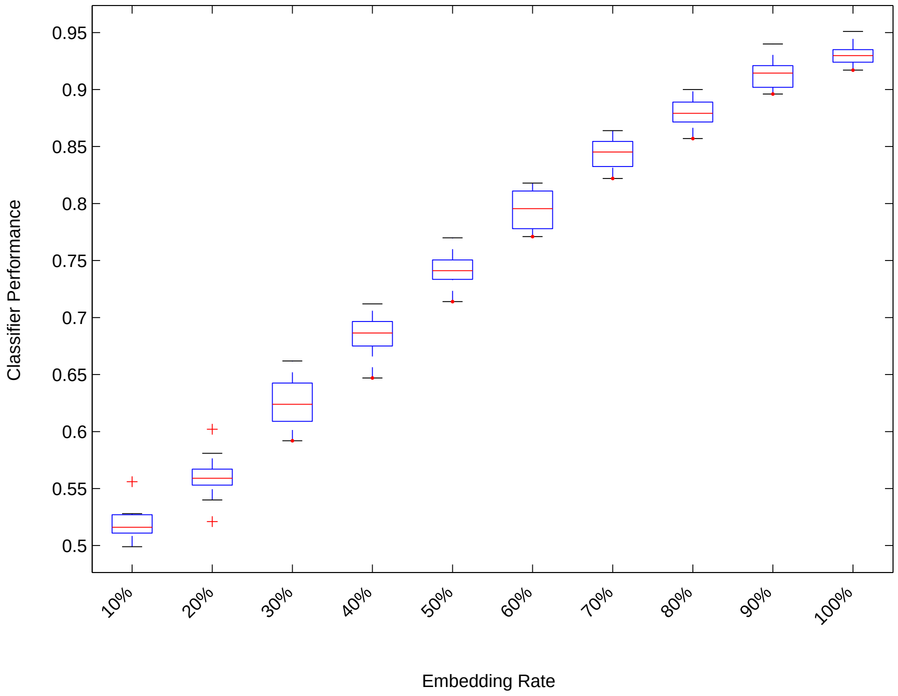
4.1. Test Set

The test set consists of approximately 2200 color JPEG images collected from Philip Greenspun's photography page†. All images are larger than 800x800 pixels and a 512x512 subimage is extracted from the center of each image and saved as a BMP. The set of original images is formed by using the F5 program (release 11+) to compress the BMP images with a quality of 80 and no secret message. The stegoimages are created with a quality of 80 and embeddings of $10, 20, \ldots, 100\%$ of the individual image capacities. For the test set the average capacity is 4.74kB.

*We do not consider the DC coefficient, as it is not used for embedding.

†http://www.photo.net/philg/digiphotos/

Figure 4. Classification Results



4.2. Coefficient Classification

LS-SVMLab 1.5[‡] is used to perform the Kernel Fisher Discriminant analysis. The training sets and testing sets are normalized to zero mean and unit variance (based only on training set statistics). The Gaussian Kernel is used in the analysis.

The data used in testing is as follows,

- 150 original images training
- 150 stego images training
- 500 original images testing
- 500 stego images testing

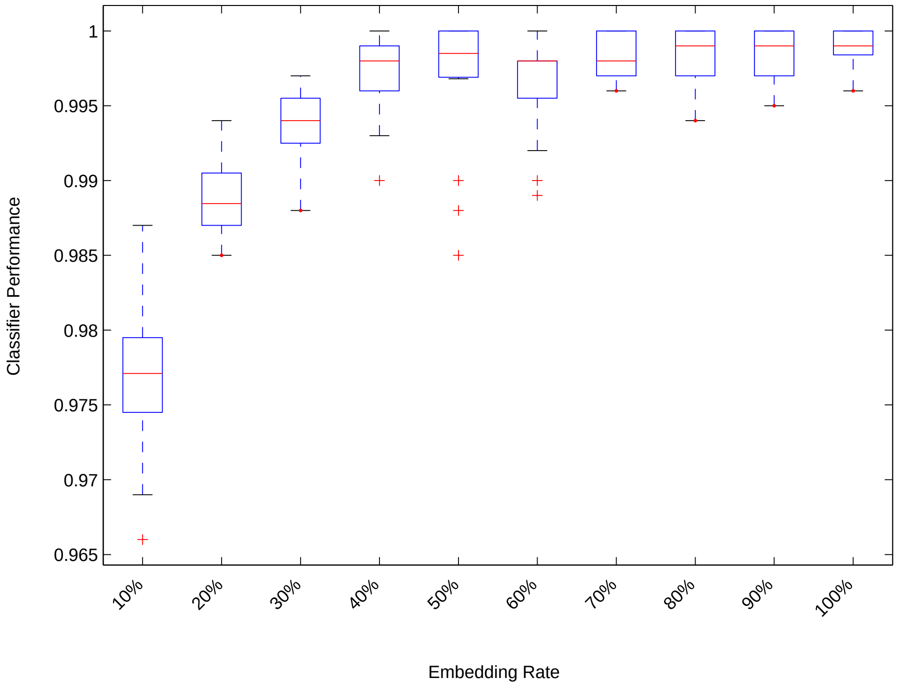
From each of these images the feature of Equation (6) is found using $p = 4$ and the frequency partition of Table 2. In the training of the classifier, two parameters need to be optimized: the σ^2 of the kernel and the regularization value γ of (15). Both parameters are tuned using the gridsearch provided by LS-SVMLab, using a 10 fold cross validation as the cost function.

The performance of the classifier is shown in Figure 4. For each embedding rate, the classifier was trained and tested 15 times, the box plots show the extreme values as well as the mean and confidence intervals.

As can be seen the classifier performs well at high embedding rates, with performance falling off sharply as the message size decreases.

[‡] Available from: http://www.esat.kuleuven.ac.be/sista/lssvmlab/

Figure 5. Spatial Method Results



4.3. Spatial Method Classification

A second classifier is constructed as in the previous section with the exception that the classifier feature is created using features from the spatial F5 method of Section 3.3. The feature is created as,

$$\mathbf{x} = [\beta_1 \ \beta_2 \ \beta_3]^\top . \tag{7}$$

The motivation for using only the first three estimates is that these DCT elements typically have the greatest number of non-zero values and produce statistically reliable estimates of β.[10] Results shown in Figure 5.

As can be seen the additional processing required for the spatial method pays off in far greater classification accuracy for low payload embeddings.

5. CONCLUSIONS

This work presents two methods for the detection of F5 steganography in JPEG images. The first method places an emphasis on the efficiency of the detection process while the second focuses on accuracy. Both methods use the powerful Kernel Fisher discriminant as the classification mechanism.

In the first method, only the indices of the quantized DCT coefficients are used in the feature construction. Thus to classify an image, only the entropy decoding step of the JPEG decompression is required. Furthermore, the processing of the indices is in the form of the computationally efficient histogram operation. The resulting classifier is able to accurately detect F5 steganography at high embedding rates, but fails as the payload decreases.

In the second method, spatial processing is used to create a feature for the classifier. Using existing steganalysis techniques a feature is generated for the Kernel Fisher discriminant. This method produces excellent detection rates, even at lower embedding rates.

The need for systems that accurately detect information hiding is well understood and the majority of present systems rightfully focus on this goal. Additionally, the real-world utility of detections systems depend heavily on their computational efficiency. This work in this paper shows that it is possible to create detection schemes tailored for each of these important characteristics.

APPENDIX A. KFD APPENDIX

A.1. Fisher Linear Discriminant

This derivation follows that of Mika.[5] Let χ be a set of training examples, where $\chi = \{\mathbf{x}_1, \ldots, \mathbf{x}_\ell\}$ with $\mathbf{x} \in \Re^N$. The training examples consist of two classes, χ_1 and χ_2, where $|\chi_i| = \ell_i$. Also let $\mathbf{y} \in \{-1, 1\}^\ell$ be the corresponding training labels. The kernel matrix is defined as,

$$K = \begin{bmatrix} \mathcal{K}(\mathbf{x}_1, \mathbf{x}_1) & \cdots & \mathcal{K}(\mathbf{x}_1, \mathbf{x}_\ell) \\ \vdots & \ddots & \vdots \\ \mathcal{K}(\mathbf{x}_\ell, \mathbf{x}_1) & \cdots & \mathcal{K}(\mathbf{x}_\ell, \mathbf{x}_\ell) \end{bmatrix},$$

and a specific column of the kernel matrix is,

$$K_{\mathbf{x}_n} = \begin{bmatrix} \mathcal{K}(\mathbf{x}_1, \mathbf{x}_n) \\ \vdots \\ \mathcal{K}(\mathbf{x}_\ell, \mathbf{x}_n) \end{bmatrix}.$$

The goal is to find $\mathbf{w} \in \mathcal{F}$ that does the following:

1. Maximizes the interclass distance between the projected points
2. Minimizes the intraclass variance of the projected points

The solution is restricted to lie in a subspace spanned by the images of the training points,

$$\mathbf{w} = \sum_{\mathbf{x} \in \chi} \alpha_{\mathbf{x}} \mathbf{\Phi}(\mathbf{x}). \tag{8}$$

The mean of each class in feature space is,

$$\mathbf{m}_i = \frac{1}{\ell_i} \sum_{\mathbf{x} \in \chi_i} \mathbf{\Phi}(\mathbf{x}). \tag{9}$$

Letting $\boldsymbol{\alpha} = [\alpha_{\mathbf{x}_1} \; \ldots \; \alpha_{\mathbf{x}_\ell}]^\top$ and $\boldsymbol{\mu}_i = \frac{1}{\ell_i} \sum_{\mathbf{x} \in \chi_i} K_{\mathbf{x}}$, the projected mean is written as,

$$\begin{aligned} \mathbf{w}^\top \mathbf{m}_i &= \frac{1}{\ell_i} \sum_{\mathbf{z} \in \chi} \sum_{\mathbf{x} \in \chi_i} \alpha_{\mathbf{x}} \langle \mathbf{\Phi}(\mathbf{z}), \mathbf{\Phi}(\mathbf{x}) \rangle \\ &= \frac{1}{\ell_i} \sum_{\mathbf{z} \in \chi} \sum_{\mathbf{x} \in \chi_i} \alpha_{\mathbf{x}} \mathcal{K}(\mathbf{z}, \mathbf{x}) \\ &= \boldsymbol{\alpha}^\top \boldsymbol{\mu}_i. \end{aligned} \tag{10}$$

The squared distance between the projected class means is then,

$$\begin{aligned} \mathbf{w}^\top S_I \mathbf{w} &= \mathbf{w}^\top (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^\top \mathbf{w} \\ &= \boldsymbol{\alpha}^\top (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^\top \boldsymbol{\alpha} \\ &= \boldsymbol{\alpha}^\top M \boldsymbol{\alpha}, \end{aligned} \tag{11}$$

where,

$$M \triangleq (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^\top.$$

The variance of each class is,

$$S_N = \sum_{i=1,2} \sum_{\mathbf{x} \in \chi} \left(\mathbf{\Phi}\left(\mathbf{x}\right) - \mathbf{m}_i \right) \left(\mathbf{\Phi}\left(\mathbf{x}\right) - \mathbf{m}_i \right)^\top . \tag{12}$$

The variance of the projected points,

$$\begin{aligned} \mathbf{w}^\top S_N \mathbf{w} &= \mathbf{w}^\top \left[\sum_{i=1,2} \sum_{\mathbf{x} \in \chi_i} \left(\mathbf{\Phi}\left(\mathbf{x}\right) - \mathbf{m}_i \right) \left(\mathbf{\Phi}\left(\mathbf{x}\right) - \mathbf{m}_i \right)^\top \right] \mathbf{w} \\ &= \boldsymbol{\alpha}^\top \left[\sum_{i=1,2} \sum_{\mathbf{x} \in \chi_i} \left(K_{\mathbf{x}} - \boldsymbol{\mu}_i \right) \left(K_{\mathbf{x}} - \boldsymbol{\mu}_i \right)^\top \right] \boldsymbol{\alpha} \\ &= \boldsymbol{\alpha}^\top \left(K \left(I - \mathbf{v}_1 \mathbf{v}_1^\top - \mathbf{v}_2 \mathbf{v}_2^\top \right) K^\top \right) \boldsymbol{\alpha} \\ &= \boldsymbol{\alpha}^\top N \boldsymbol{\alpha}, \end{aligned} \tag{13}$$

where,

$$N \triangleq K \left(I - \mathbf{v}_1 \mathbf{v}_1^\top - \mathbf{v}_2 \mathbf{v}_2^\top \right) K^\top .$$

Here, I is the identity matrix, $\mathbf{v}_1 = \{v_1, \ldots, v_{\ell_1}\}$ is a vector with $v_i = 1/\sqrt{\ell_1}$ if the ith training example belongs to the first class and zero otherwise. Likewise $\mathbf{v}_2 = \left\{v'_1, \ldots, v'_{\ell_2}\right\}$ is a vector with $v'_i = 1/\sqrt{\ell_2}$ if the ith training example belongs to the second class and zero otherwise.

The goal is to maximize (11) and minimize (13), so the Rayleigh coefficient is constructed,

$$J(\boldsymbol{\alpha}) = \frac{\boldsymbol{\alpha}^\top M \boldsymbol{\alpha}}{\boldsymbol{\alpha}^\top N \boldsymbol{\alpha}}. \tag{14}$$

As is shown in[5] the solution to this maximization is equivalent to solving the following quadratic optimization problem,

$$\min_{\boldsymbol{\alpha}, b, \boldsymbol{\xi}} \left|\left|\boldsymbol{\xi}\right|\right|^2 + \gamma \left|\left|\boldsymbol{\alpha}\right|\right|^2 \qquad \text{subject to: } K\boldsymbol{\alpha} + \mathbf{1}b = \mathbf{y} - \boldsymbol{\xi}. \tag{15}$$

Here the term $\gamma \left|\left|\boldsymbol{\alpha}\right|\right|^2$ is the regularization constraint.

APPENDIX B. SPATIAL F5 METHOD

For a complete discussion of this spatial method see Fridrich.[10] For an unaltered image let $h_k^*\left[d\right]$ be defined as the number of quantized DCT coefficient indices in the kth DCT element, with an absolute value of d. Likewise for a stegoimage, let $H_k^*\left[d\right]$ be the number of indices of the kth DCT element, with an absolute value of d. The probability that a index in the kth DCT element is changed is denoted β_k.

Since F5 embeds by decreasing the absolute value of the index, the stegoimage histogram becomes,

$$H_k^*\left[0\right] = h_k^*\left[0\right] + \beta_k h_k^*\left[1\right] \tag{16a}$$

$$H_k^*\left[d\right] = (1 - \beta_k) h_k^*\left[d\right] + \beta_k h_k^*\left[d+1\right], \qquad \text{for } d > 0. \tag{16b}$$

For an unknown image, it is possible to estimate β_k if the original histogram $h_k^*\left[d\right]$ is known. As this is assumed to be unknown, an estimate is found. This estimate is denoted $\hat{h}_k^*\left[d\right]$ and is created as follows.

First the image is cropped in the spatial domain by four pixels in each direction. This serves to break the 8×8 block structure of the DCT. When the image is compressed again (using the same quantization table) index histogram of the cropped image has been found to approximate the original, i.e. $\hat{h}_k^*\left[d\right] \approx h_k^*\left[d\right]$.

Since the most prominent changes in embedding occur in the histograms of $H_k^*\left[0\right]$ and $H_k^*\left[1\right]$ the analysis is restricted to these values. The value of β_k may be estimated through the solution of the following minimization,

$$\beta_k = \arg\min_{\beta} \left[H_k^*[0] - \hat{h}_k^*[0] - \beta\hat{h}_k^*[1]\right]^2 + \left[H_k^*[1] - \beta\hat{h}_k^*[1] - (1-\beta)\hat{h}_k^*[1]\right]^2. \tag{17}$$

Here the terms of the minimization are those of (16a) and (16b). The solution of this minimization is,

$$\beta_k = \frac{\hat{h}_k^*[1]\left(H_k^*[0] - \hat{h}_k^*[0]\right) + \left(H_k^*[1] - \hat{h}_k^*[1]\right)\left(\hat{h}_k^*[2] - \hat{h}_k^*[1]\right)}{\hat{h}_k^2[1] + \left(\hat{h}_k^*[2] - \hat{h}_k^*[1]\right)^2}. \tag{18}$$

ACKNOWLEDGMENTS

The support of the Center for Integrated Transmission and Exploitation (CITE) and the Information Directorate of the Air Force Research Laboratory, Rome, NY is gratefully acknowledged.

REFERENCES

1. A. Westfeld and A. Pfitzmann, *High Capacity Despite Better Steganalysis (F5-A Steganographic Algorithm)*, vol. 2137 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, Germany, 2001.
2. H. Farid, "Detecting steganographic messages in digital images," Tech. Rep. TR2001-412, Dartmouth College.
3. B. Scholkopf and et. al, "Input space versus feature space in kernel-based methods," *IEEE Trans. on Neural Networks* **10**, pp. 1000 – 1017, Oct. 1999.
4. C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery* **2**(2), pp. 121–167, 1998.
5. S. Mika, G. Ratsch, J. Weston, B. Scholkopf, A. Smola, and K. Muller, "Constructing descriptive and discriminative nonlinear features: Rayleigh coefficients in kernel feature space," *IEEE Trans. on Pattern Analysis and Machine Intellignece* **25**, pp. 623 – 628, May 2003.
6. T. Van Gestel, J. Suykens, G. Lanckriet, A. Lambrechts, B. De Moor, and J. Vandewalle, "Bayesian framework for least squares support vector machine classifiers, gaussian processes and kernel fisher discriminant analysis," *Neural Computation* **15**, pp. 1115 – 1148, May 2002.
7. J. Suykens, T. Van Gestel, J. D. Brabanter, B. De Moor, and J. Vandewall, *Least Squares Support Vector Machines*, World Scientific, Singapore, 2002.
8. A. Westfeld and A. Phitzmann, "Attacks on steganographic systems," in *Proceedings* 3^{rd} *Information Hiding Workshop*, pp. 61–75, (Dresden, Germany), Sept. 28-Oct. 1 1999.
9. E. Y. Lam and J. W. Goodman, "A mathematical analysis of the DCT coefficient distributions for images," *IEEE Trans. on Image Processing* **9**, pp. 1661–1666, Oct. 2000.
10. J. Fridrich, M. Goljan, and D. Hogea, "New methodology for breaking steganographic techniques for JPEGs," in *Proc. SPIE Electronic Imaging 5022*, (Santa Clara, CA), Jan. 21–24, 2003.