FEATURE-BASED STEGANALYSIS WITH IMAGE NOISE AND APPROXIMATE HISTOGRAMS

Benjamin Tyler

Department of Computer Science, Lake Forest College
Lake Forest, IL, 60045

Abstract

In this paper I explore using the noise within a JPEG image and the Gaussian approximations of the quantized DCT coefficient histograms of the image as features for the detection of steganographically embedded messages.

Index terms – Steganalysis

1. Introduction

Steganography is the process of hiding a message within another object. These cover objects look benign allowing the message to avoid detection. With the advent of the digital era it has become easy to embed messages using digital images as the cover objects. Detecting these messages can be difficult, especially if the image is compressed. Every steganographic algorithm leaves some sort of mark on the image. Steganalysis is where the marks left the image are discovered through different means. Feature based steganalysis searches for specific features of an image that may have obvious marks left by an embedded message. Areas such marks are most obvious are …

2. Related work

Work related to feature based steganography has been done before. This is where the 23 features that were used as a starting point originated. The 23 features were developed for detecting images using four types of steganographic algorithms. The algorithms are called F5, OutGuess, MB1 and MB2. Each algorithm has a different approach to embedding the message in the image. For example the MB1 algorithm focuses on removing artifacts from the edges of the 8x8 blocks created by JPEG compression. The two features I created were derived from previous work on determining image data paths and intrinsic image fingerprints. Part of the approach to determining intrinsic image fingerprints is analyzing the patterns of noise left by different compression algorithms. Another part of determining a datapath is approximating the original histogram of an image before compression was applied. Out of these I created two types features to be used with feature based steganalysis.

In this paper we propose combining the 23 features with features I created, to see if the detection of embedded messages could be at all improved. The idea behind our approach is that the features that allow the determination of intrinsic fingerprints can be applied to finding embedded messages within images.

The paper is organized with Section 3 laying out the approach taken in using the features. Section 4 discusses the results acquired in the experiments. Section 5 discusses the conclusions that can be drawn from the results. Section 6 presents ideas for future work on similar projects.

3. Proposed approach

We combine the concept of calibration with the feature-based classification to devise a blind detector specific to JPEG images. Specifically, our features include 23 features extracted from the calibration process [1], the 30 features extracted from the estimated noise, and the 36 Gaussian-fitting-based features extracted from the histogram of the quantized DCT coefficients across all frequencies.

Several advantages to investigate these three sets of features are that each set of features takes looks at different parts of the image. The 23 features focus on the differences between the JPEG compression blocks. The noise features focus on the image as a whole. The approximation of the histogram is directed at the DCT coefficients.

In the following, we explain each type of features in detail.

3.1. The 23 Features

The original image processing extracted six types of features, these features are the global histogram, individual histograms for 5 DCT modes, dual histograms for 11 DCT values (–5, …, 5), variation, L1 and L2 blockiness, and co-

occurrences. These features look primarily at the patterns that develop within and between the 8x8 blocks created in the DCT compression. The global histogram is the histogram of the entire image. The 5 DCT modes are histograms of the 5 lowest frequency DCT modes. The dual histograms are histograms of the occurrence of values between compression blocks. Variation is the overall entropy of the image. Blockiness is a measure of the discontinuity between adjacent compression blocks. Co-occurrence examines the distribution between paired, adjacent DCT coefficients. I treated these features as a single feature set, having already undergone testing with reliable results.

3.2. The Noise Features

The features derived from analyzing the noise of an image are the mean and standard deviation of the log transformed noise of the image. The noise is acquired by denoising the image then subtracting the denoised image from the original, leaving only the noise. I used averaging, median, Gaussian, and Wiener denoising filters. These noise types were groups into two feature sets, the Wiener denoised features and the other denoised features. These features are broken into two groups based on that a 3x3 and a 5x5 Wiener filters are used when the other filters were all 3x3. These two feature sets would then be tested individually, together and with other feature sets. The filters function on the three planes within an RGB image, with each the mean and standard deviation being calculated for each plane. Consequently, the number of features for the two feature sets is 12 for the Wiener filters and 18 for the other filters.

3.3. The Gaussian-Fitting-Based features

The approximation of the DCT histograms is done by finding the maximum value in each histogram bin, effectively the "top" point. These maximum values are then used as points for a fit function to fit the sum of three Gaussian curves to the points (eqn 1).

Equation 1

$$\alpha_1 e^{\left(-((x-\gamma_1)/\nu_1)^2\right)} + \alpha_2 e^{\left(-((x-\gamma_2)/\nu_2)^2\right)} + \alpha_3 e^{\left(-((x-\gamma_3)/\nu_3)^2\right)}$$

This resulting curve approximates the histogram. The coefficients of the component three curves are then used as features. The features would consist of three feature subsets, the alpha coefficients, the gamma coefficients, and the nu coefficients. The alpha, gamma and nu features could then be tested individually, together or with other types of features. The approximated histograms can be of the DCT of the entire image or select DCT frequencies. I used both the entire image and the lowest five individual frequencies of the DCT. The single frequency features are listed as "approx" in table 1 and the features based on the full DCT of an image are listed as "DCT".

4. Experimental Results

In my experiments I used a data set created from cropped and scaled images I took. The data set created for each of the embedding algorithms varied a little as some images were unable to have a message embedded in them. For the final data sets I had 499 un-embedded images, 502 images embedded with the F5 algorithm [3], 439 images embedded with the OutGuess algorithm [4], and 394 images embedded with the MB1 algorithm [5]. The maximum length message allowed by each algorithm used when was embedding. These algorithms were chosen based on the previous work done with them using only the 23 features. This would allow me to see if my features allowed for improvement. These images were used as a training database for the classify function built into MatLab. To test this training I had a set of test images. This test set was composed of 50 un-embedded images, 48 images embedded with the F5 algorithm, 40 images embedded with the OutGuess algorithm, and 47 images embedded with the MB1 algorithm. All features were extracted from these data and testing sets, after which, test cases and the training data required for the case were

assembled. The tests checked the success of the classification function at distinguishing between each of the individual steganographic algorithms and then between non-embedded and embedded images as a whole. In table 1 the individual algorithms are appropriately referenced and the test between any embedding and none is called "stego". For each algorithm the test cases were classified based on the training data and an ROC curve generated to determine the successful detection of embedded images (fig1,2). Some features were not used due to difficulty in processing. Specifically the alpha features from the approximation of the DCT histogram tended to be too large for MatLab to effectively process. Presented in table 1 are areas under the generated ROC curves for the given cases. These areas give the percentage of correct detections. Not every possible case is presented in table 1, the results are meant to be representative. The classifier function took training data and testing data along with labels for the data. the labels would tell the classifier which image data in the training set was an original or steganographically embedding image. The classify function would then use the training data to assign the image data in the testing set a label. The resulting list of labels would be used with the perfcurve function. This function takes the list of labels from the classification, the real labels for the data that was classified and a score which weights the labels. The score signifies which label would be considered a positive result. The perfcurve function then calculates the ROC curve. The area under this curve is equivalent to the percentage of correct classifications. Table 1 presents this effective percentage of detection for several cases of individual and combined feature sets. The first column gives the percentage when distinguishing between original images and those with a message embedded using the F5 algorithm. The second and third columns give similar results, but with the OutGuess and MB1 algorithms respectively. The fourth column is the percentage in correctly classifying between original images and images upon which any of the three steganographic algorithms had been used. The last column provides a basic measure of the overall effectiveness of the feature combination at detecting embedded messages across all four types of detection tested. This score is generated by simply adding the numbers from the previous columns together. Rows of note would be the first row, where only the 23 original features are used for detection. The best detection results were when the 23 features, along with all the noise features and two types of approximated features were used. This result can be seen in the last row. The second best result was generated by using only the 23 features with all the noise features.

Table 1: Experimental Results

Features used	F5	OutGuess	MB1	Stego'ed	Score total
23	.8258	.9875	.9681	.8837	3.6651
Approx Gamma	.6108	.5550	.4002	.5593	2.1253
Approx Nu	.4783	.4800	.6091	.5115	2.0789
DCT Gamma	.5921	.5025	.6396	.6085	2.3427
DCT Nu	.5437	.5225	.5445	.5426	2.1553
Filter Noise	.7658	.9900	.3840	.7270	2.8668
Wiener Noise	.7008	.9900	.4991	.7056	2.8955
All Noise	.5940	.9900	.7143	.6696	2.9679
23 and Approx Gamma	.8163	.9750	.9681	.8826	3.642
23 and DCT Gamma	.8258	.9875	.9681	.8837	3.6651
23 and Filter Noise	.9088	.9900	1	.9541	3.8529
23 and Wiener Noise	.8454	.9900	.9681	.9096	3.7131
23 and All Noise	.9188	.9900	1	.9678	3.8766
23, Approx Nu, DCT Gamma, Filter Noise	.8783	.9900	1	.9615	3.8561

23, Approx Nu, DCT Gamma, All Noise	.9396	.9900	1	.9715	3.9011

Figure 1 – ROC curves of 23 features

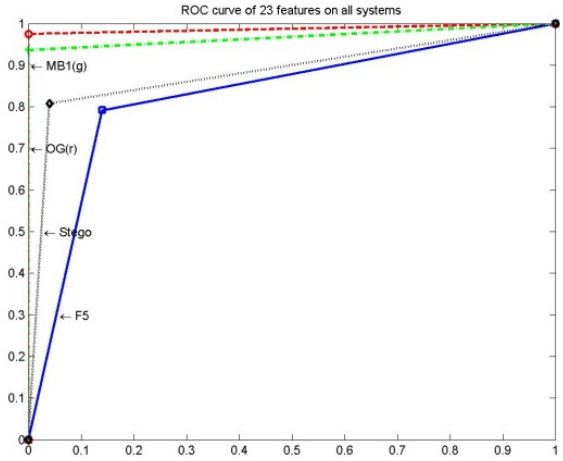
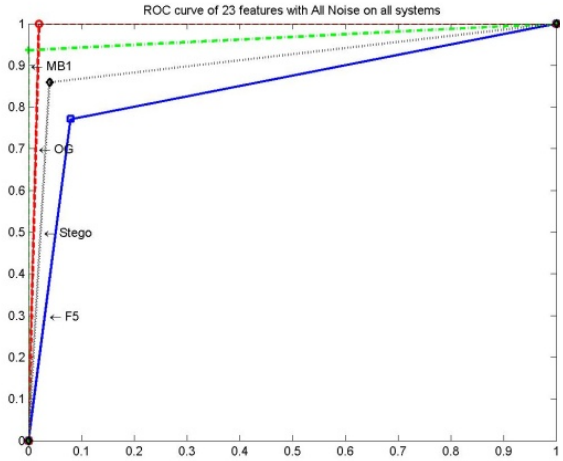


Figure 2 – ROC curves of 23 and Noise features



5. Conclusion

Detection of steganograpically embedded images increased dramatically when the original 23 features were combined with the noise features. A slight additional improvement could be made by including the Approximate nu and either DCT gamma or DCT nu features. By themselves neither the noise nor approximated histogram features were as accurate as the 23 features. The noise features by themselves achieved about 75% detection. The approximate histogram features by themselves were inaccurate, being no more precise than flipping a coin. When the approximate histogram features were added to other features the result was the same or slightly worse than without the approximate histogram features. This means that the primary features and not the approximate histogram features were driving the detection.

6. Future work

Possibilities for extending the work done here could involve seeing the effect of normalizing the feature vectors. Testing could be done to determine which specific features in the noise feature set are the most powerful and eliminate the extraneous features.

7. References

[1] J. Fridrich, “Feature-Based Steganalysis for JPEG Images and its Implications for Future Design of Steganographic Schemes,” in *6th Information Hiding Workshop*, 2004.

[2] W. S. Lin, S. K. Tjoa, H. V. Zhao, and K. J. Ray Liu, ”Digital Image Source Coder Forensics Via Intrinsic Fingerprints,” in *IEEE Transactions on Information Forensics and Security*, vol. IV, pp. 460-475. September 2009.

[3] A.Westfeld. “F5—A Steganographic Algorithm,” in *Information Hiding: 4th International Workshop*, volume 2137 of Lecture Notes in Computer Science, pages 289–302, 2001.

[4] N. Provos, “Defending Against Statistical Steganalysis,” in *10th USENIX Security Symposium*, August 2001.

[5] P. Sallee, "Model-Based Steganography" in *International Workshop on Digital Watermarking*, October, 2003.