Image Spam Detection

A Project

Presented to

The Faculty of the Department of Computer Science

San Jose State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Aneri Chavda

May 2017

The Designated Project Committee Approves the Project Titled

Image Spam Detection

by

Aneri Chavda

APPROVED FOR THE DEPARTMENTS OF COMPUTER SCIENCE

SAN JOSE STATE UNIVERSITY

May 2017

Dr. Mark Stamp	Department of Computer Science
Dr. Thomas Austin	Department of Computer Science
Dr. Aikaterini Potika	Department of Mathematics

ABSTRACT

Image Spam Detection

by Aneri Chavda

Electronic mail or email is one of the most common digital communication tools. Spam is unsolicited bulk emails. Spam text embedded inside images can be loosely defined as Image Spam or Image-based spam. Image spam can bypass standard signature based detection schemes. Image spam has thus become a threat to email based communication. In this research, we propose an approach to image spam detection classifiers that are built using a combination of image processing and machine learning techniques.

ACKNOWLEDGMENTS

I would like to acknowledge my family and friends who have motivated me and have contributed in shaping my journey. I would like to thank Dr. Mark Stamp for providing constant guidance for my project. Dr. Stamp helped me look at security domain from a different perspective.

My parents, Mr. Girish Chavda and Mrs. Urmi Chavda have been my pillars of strength. They have made me a better person and supported all my endeavors in life. I would like to thank them for believing in me and pushing me further to purse Master's. I would also like to thank my little brother Mr. Jeet Chavda for being there for me and constantly motivating me.

TABLE OF CONTENTS

CHAPTER

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

Electronic mail or email is one of the most common digital communication tool used currently. A survey conducted on Internet users in the US in 2010 indicated that 94% of them have used email as a communication tool. It also suggested that 62% of them use emails daily. These numbers have significantly grown since 2010 [1] . Rise in the amount of emails exchanged has attracted significant number of spammers. Email spam is unsolicited bulk email. Spam e-mails pose as a serious threat to the usefulness and popularity of email which contains advertisements, malware, phishing links, adult content, etc.

In the nascent stages, email spam was seen in the form of text emails. Several successful classifiers were developed to filter spam emails based on content, subject, header, etc. Lai and Tsai [2] explore 4 Machine Learning algorithms used to build classifiers using different parts of the email like content, body, header, etc. Machine Learning algorithms like Naive Bayes, Support Vector Machines (SVM), k-Nearest Neighbors (KNN), etc. have been widely used in email spam detection.

With stronger text based classifiers being developed to filter spam emails, spammers invented various other techniques like blank spam, image spam and backscatter spam to evade text based detection. Email spam sent in the form of images is commonly known as email image spam or image based spam. Spam text is embedded inside images making it easier to evade text based detection [3]. "Spam now accounts for 90.4 percent of all e-mail, according to a report released Monday from security vendor Symantec" [4].

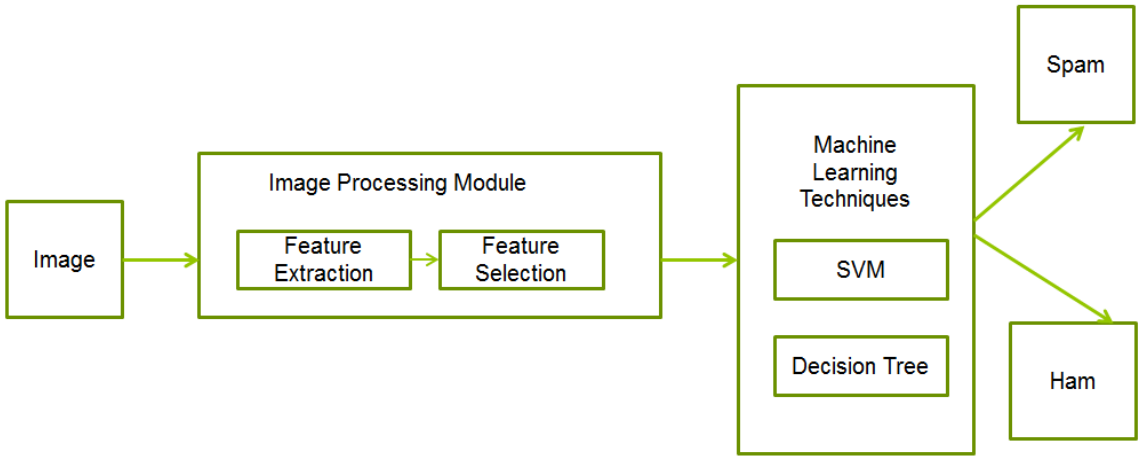


Figure 1: Proposed Architecture

Initial image spam was seen in the form of simple text converted to images. Optical Character Recognition (OCR) can be used to extract text inside an image. This text is then subjected to text based detection techniques. To evade OCR based detection scheme, spammers introduced obfuscation techniques in spam images which prevent OCR from reading the text embedded inside the images [5].

This opened a whole new avenue for research in image spam detection techniques. Image processing techniques were introduced for image spam detection. Detection schemes are now developed using multiple image properties as input to machine learning algorithms.

In this research, we will explore how image processing can be used in conjunction with machine learning algorithms. Fig. 1. shows the basic architecture of the proposed research. Since there are very limited public datasets available for image spam research, we developed a synthetic dataset. The aim of constructing this dataset was to challenge the detection scheme we developed.

In chapter 2 we give a brief overview about what is image spam, detection techniques and related work done in this domain. In chapter 3, we talk about image features used in the experiments. Chapters 4 and 5 give a brief overview about datasets and machine learning model used for the experiments. Chapter 6 details the environmental setup. In chapter 7, we discuss the process of generating the synthetic dataset and evaluate it. Chapter 8 dives into the experimental results of SVM Model with the datasets.

CHAPTER 2

Background

2.1 Types of Image Spam

Image Spam is an email spam technique developed to evade content based detection techniques. Image spam techiniques have evolved today. We can loosely classify them into 3 generations [6].

- First Generation Image Spam: The onset of image spam began with simple text embedded inside images. This was a successful effort to evade content based detection schemes. Combining Optical Character Recognition technique with content based filtering served as a good classifier for this class of image spam.
- Second Generation Image Spam: In the second generation of image spam, background images and noise were introduced in the image. This was an attempt make OCR filtering difficult.
- Third Generation Image Spam: This class of image spam introduced relevant images along with the text. This made OCR based detection difficult.

2.2 Spam Detection Techniques

- Content Based Filters: Content based detection schemes can be used to filter text based spam emails. They rely on the content/text inside the spam emails. String classifiers are built using keywords extracted from spam emails, headers, payload, etc. Machine Learning techniques have been used exhaustively to build these type of classifiers [1].
- Non-content based Filters: Non-content based detection schemes are used to detect more advanced forms email spams like image spam. These detection

schemes heavily rely on other properties of the emails like image properties.

2.3 Related Work

Since the onset of spam detection, machine learning techniques have been used exhaustively. Image spam has further widened this research area. A combination of Image Processing and Machine Learning techniques have resulted in strong image spam detection schemes.

Kumaresan et al. [7] used combination of 10 metadata features and 3 texture features to construct a feature vector for each image. Using SVM with particle swarm optimization, they achieved an accuracy of 90% on dataset [8].

Annapurna et al. [6] used 21 features to generate the feature vector. After conducting various experiments, with feature selection and feature elimination based on the weights associated with each feature, they constructed a strong SVM classifier. The experiments were conducted on 2 datasets [3, 8] and the accuracy achieved with each dataset was 97% and 99%. More features in comparision to [7] were used in this experiment like edges, noise, etc. Addition of these features helped improve the accuracy by 9% in dataset [8].

Soranamageswari et al. [9] proposed a similar architecture with Neural Networks. Color and image composition feature are extracted and fed to BPNN. They achieved an accuracy of 92.82% on the Spam Archive dataset [10].

Chowdhury et al. [11] extracted metadata features and visual features and fed it to BPNN. They presented a comparison of 3 machine learning algorithms; Naive Bayes, SVM and BPNN on the same dataset, with the same set of features. The results showed that despite of increased complexity, neural networks achieved greater

accuracy than the other two models.

From the results of the two papers, we can see that neural networks outperform traditional Machine Learning techniques like SVM and Naive Bayes. It can be attributed to how neural networks 'learn' from the data presented to it.

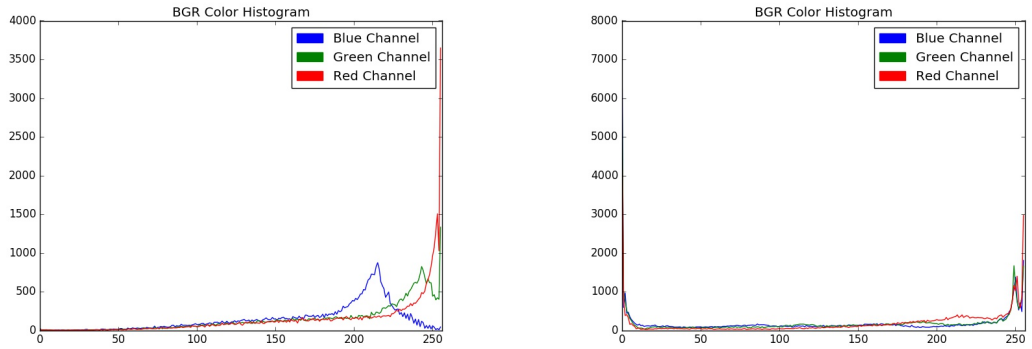
CHAPTER 3

Image Processing

3.1 Image Features

Image features are analogous to image properties. Since spam images are generated by computer, properties of spam images vary a lot from natural/ham images. For instance change in brightness in natural images is very high compared to that of spam images. Using advanced image processing techniques many such properties can be extracted from images. A total of 41 features were collected of which 21 are based on previous research [12]. Table 1 gives a brief overview of all the features. These features can be loosely classified in 5 domains.

- Metadata Features: Properties like image size, height, width, aspect ratio, compression ratio, bit depth, image name, etc. are the basic set of properties of an image. A certain anomaly can be seen in computer generated images versus natural images. We used compression ratio and aspect ratio as 6 features.
- Color Features: Different histograms convey different properties of an image.
  - Color Histogram: A color histogram extracts the usage of Red, Green, Blue colors. Normally in a spam image, very few colors are used compared to natural images. Along with color histogram, RGB histograms can also be quantized and used for classification. Fig. 2. compares the RGB channels of ham and spam images.
  - Reg, Green, Blue Histograms: Mean, Variance, Skew and Kurtosis of each of these 3 histograms is calculated. All combined 12 features are extracted from the 3 histograms.
  - Hue, Saturation and Value (HSV) Histogram: HSV Histogram captures



(a) Ham (b) Spam

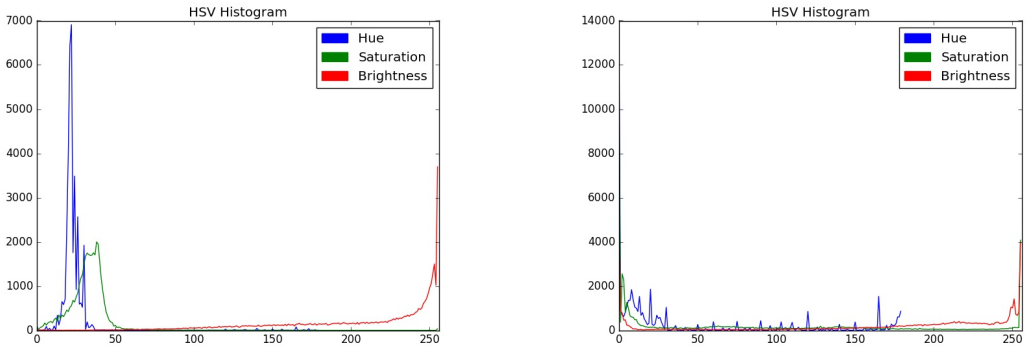
Figure 2: RGB Channels of Color Histogram

the following 3 aspects of the colors of an image.

* Hue: It defines how close the color is to red. Hue is measured between 0 to 1; 0 being red.
* Saturation: It defines how pure the color is. Higher values of saturation correspond to deeper/richer colors. While corresponds to 0 saturation.
* Intensity/Value: Intensity defines brightness. Higher values of intensity correspond to white.

– Hue, Saturation, Intensity Histograms: Mean, skew, variance nd kurtosis of each of these histograms are captured. This adds up to 12 features extracted from the 3 features. Fig. 3. compares the HSV channels of ham and spam images.

- Texture Features:
  – Local Binary Pattern (LBP) Histogram: This histogram is commonly used for texture classification. For each pixel, LBP helps quantify how similar or different each pixel is from its neighboring pixels. Spam images have relatively less information in this histogram.
- Shape Features:



(a) Ham (b) Spam

Figure 3: HSV Channels of HSV Histogram

  - Histogram of Oriented Gradients: This histogram is commonly used for object detection. It describes how the intensity of gradients change in the image.
  - Edges: Edges mark the change in contrast. Edge detection highlights boundaries of features in an image [12]. Fig. 4. shows canny edge detector output on a spam image and a ham image. Spam images in general contain a lot of text, resulting in an increased number of edges than ham images. Another observation we can make by looking at the images is that edges in spam images are smaller compared to that in ham images. Number of edges and average edge length have been considered as 2 features.

- Noise Features:
  - Entropy of Noise: Amount of noise in a spam image is less than a normal image. Entropy of noise histogram is measured as a feature.
  - Signal to Noise Ratio (SNR): For this paper SNR is the ratio of mean and standard deviation in grayscale image's histogram.

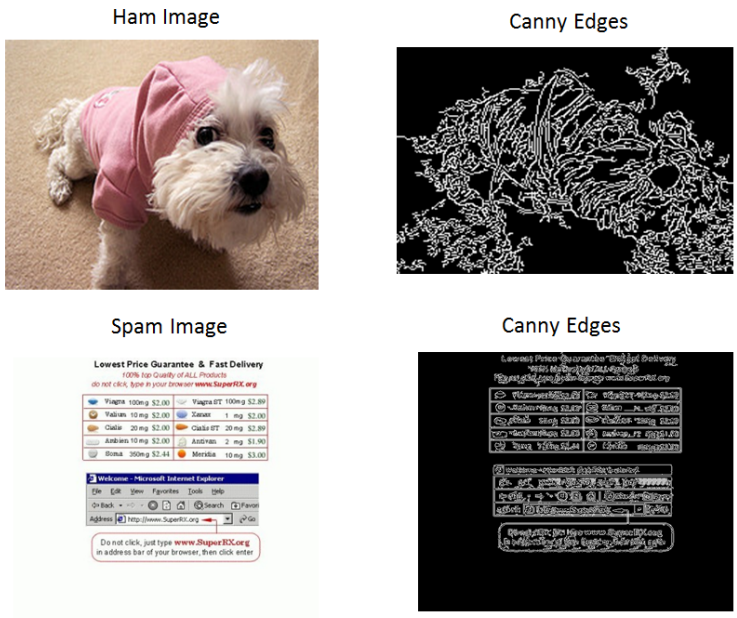


Figure 4: Canny Edge

3.2 Feature Selection

Once features have been extracted, these features have to be quantized. This process can be coined as preparing the data. Machine Learning algorithms require input in the form of feature vectors, wherein each feature is a number. Hence image features like canny edges, histograms, etc. have to be converted to numbers. Various statistical techniques like entropy of histograms, mean, variance, kurtosis, extracting number of edges from canny edge image, etc. are used.

Once feature vectors are constructed, multiple experiments can be conducted to select a subset of features to achieve greater accuracy.

Table 1: Feature set

Feature Domain	Feature	Description
Metadata Features	Height	Height of the image
	Width	Width of the image
	Aspect Ratio	Ratio of height and width
	Compression Ratio	How compressed is the image
	File Size	Size on disk
	Image Area	Area of the image
Color Features	entr-color	Entropy of color histogram
	r-mean	Mean of the red channel histogram
	g-mean	Mean of the green channel histogram
	b-mean	Mean of the blue channel histogram
	r-skew	Skew of the red channel histogram
	g-skew	Skew of the green channel histogram
	b-skew	Skew of the blue channel histogram
	r-var	Variance of the red channel histogram
	g-var	Variance of the green channel histogram
	b-var	Variance of the blue channel histogram
	r-kurt	Kurtosis of the red channel histogram
	g-kurt	Kurtosis of the green channel histogram
	b-kurt	Kurtosis of the blue channel histogram
	entr-hsv	Entropy of HSV Histogram
	h-mean	Mean of the hue channel of hsv histogram
	s-mean	Mean of the saturation channel of hsv histogram
	v-mean	Mean of the brightness channel of hsv histogram
	h-var	Variance of the hue channel of hsv histogram
	s-var	Variance of the saturation channel of hsv histogram
	v-var	Variance of the brightness channel of hsv histogram
	h-skew	Skew of the hue channel of hsv histogram
	s-skew	Skew of the saturation channel of hsv histogram
	v-skew	Skew of the brightness channel of hsv histogram
	h-kurt	Kurtosis of the hue channel of hsv histogram
	s-kurt	Kurtosis of the saturation channel of hsv histogram
	v-kurt	Kurtosis of the brightness channel of hsv histogram
Texture Features	lbp	Entropy of Local Binary Patterns histogram
Shape Features	entr-hog	Entropy of Histogram of gradients
	edges	Total number of edges in an image
	avg-edge-length	Average edge length
Noise Features	snr	Signal to Noise Ratio
	entr-noise	Entropy of noise

CHAPTER 4

Datasets

Three datasets have been used in this research. Two of these datasets are public datasets, images from actual spam and ham emails exchanged.

4.1 Dataset 1

This dataset was developed by writers of Image Spam Hunter [3] at Northwestern University. After cleaning the dataset, 920 spam images and 810 ham images were retained for the research. All the images are in jpg/jpeg format.

4.2 Dataset 2

Dredze et. al in their paper Learning Fast Classifiers [8], created an image spam corpus which is publicly available. After cleaning the dataset, 1089 spam and 1029 ham images were retained for research. All the images are in jpg/jpeg format.

4.3 Dataset 3

Since very few public image spam corpuses are available, a synthetic dataset was created. Another public corpus from Dredze et. al in their paper Learning Fast Classifiers [8] included only spam images. After cleaning the dataset, and retaining only jpg/jpeg images, and experimental dataset is constructed. The aim of creating this dataset is to defeat the detection schemes used earlier.

CHAPTER 5

Support Vector Machines (SVM)

5.1 SVM Model

SVM is a supervised learning algorithm, generally used for classification. SVM has been exhaustively used in email spam detection [2] and image spam detection [6]. In the training phase SVM constructs a separating hyper-plane. In this section, we give a brief overview of the SVM algorithm.

There are 4 key concepts of SVM algorithm as described by Stamp M., in ı Machine Learning with Applications in Information Security [13].

- **Separating Hyperplane:** In the training phase, SVM attempts to find a separating hyper-plane which divides labeled input data into two classes. In an ideal scenario, all the data of one class falls on one side of the hyperplane and other class falls on the other side.
- **Maximize Margins:** To construct an optimal hyperplane, only a subset of training data is required. These points are called the support vectors. The idea behind choosing an optimal hyperplane is to maximize the distance/margin between the support vectors of each class and the hyperplane. Fig.**??**. shows a separating hyperplane and support vectors for 2D data.
- **Work in higher dimensions:** Separating hyperplane is essentially a linear decision function. However, data of the input space is often not linearly separable. Hence, SVM converts the input data to a feature space higher dimension. Input data in this form is more spread out and linearly separable. Hence, classifying data becomes easier. This transformation is however an expensive task.

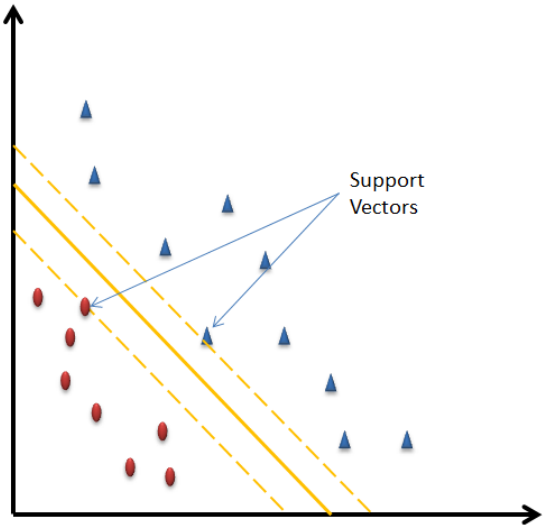


Figure 5: Separating Hyper-plane

- **Kernel Trick:** Kernel Trick is the mapping function used to transform input space to a linearly separable higher dimension. It makes a non-linear transformation an easy task. It doesn't actually perform the transformation to the higher dimension yet gives us the advantages of working in higher dimensions. Multiple Kernel functions are available like Linear Kernel, Polynomial Kernel, Radial Basis Function(RBF), etc.

5.1.1 Training Phase

Training phase involves generating an equation for the separating hyper plane. It is done by solving a Lagrangian Duality problem. Given a set of input data $X_0, X_1...., X_n$, with labels $z_0, z_1...., z_n$, where $z_i \in \{-1, 1\}$, the training phase solves the Lagrangian Duality Problem for Select Kernal function K and C as follows -

$$\text{Maximize } L(\lambda) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j z_i z_j (X_i \cdot X_j)$$

$$\textbf{Subject to constraints } \sum_{i=1}^{n} \lambda_i z_i = 0 \; and \; C \geq \lambda_i \geq 0 \; for \; i = 1, 2..n.$$

5.1.2 Testing Phase

In the testing phase, we classify a point by determining on which side of the hyperplane the point lies on.

5.2 Feature Selection

In a multidimensional input space, the cost of converting the input space to a higher dimension increases. Though SVM is a classification algorithm, SVM also calculates weights for each feature and ranks them based on their contribution to classification. The idea behind feature selection is reduction of dimensionality. The cost of converting an input space to higher dimension/applying kernel transformations is high. So instead, using the ranks for each feature, we select only top k features for testing phase.

5.2.1 Recursive Feature Elimination

Recursive Feature Elimination (RFE) is a feature selection technique used to get rid of the features that contribute the least to SVM classification. Here, we use RFE to fine tune the SVM classification for unraveling ham and spam images. RFE assigns weights to features and ranks them according to the amount of contribution towards the classification. The feature with the least rank is eliminated and the process is repeated till the desired number of features are eliminated. RFE works only with Linear Kernel of SVM.

5.2.2 Univariate Feature Selection

Univariate feature selection uses univariate statistical properties to rank each feature. The top k ranked features are then selected.

5.3 Scoring Metrics

When any data point is scored, the result is one of the following 4 outcomes-

1. True Positive(TP): The scored sample is a spam, and it is rightly classified as spam.
2. True Negative(TN): The scored sample is a ham, and it is rightly classified as ham.
3. False Positive(FP): The scored sample is ham, and it is wrongly classified as spam.
4. False Negative(FN): The scored sample is spam, and it is wrongly classified as ham.

In the real world, we want to reduce the FP rate as low as possible and increase TP and TN. We measure SVM scores in the form of accuracy. Accuracy can be defined as-

$$\textbf{Accuracy} = \frac{TP + TN}{P + N}$$

where P = total positive samples and N = total negative samples.

5.3.1 Confusion Matrix

Confusion Matrix visualizes the four cases for given dataset. Fig. 6. shows a confusion matrix.

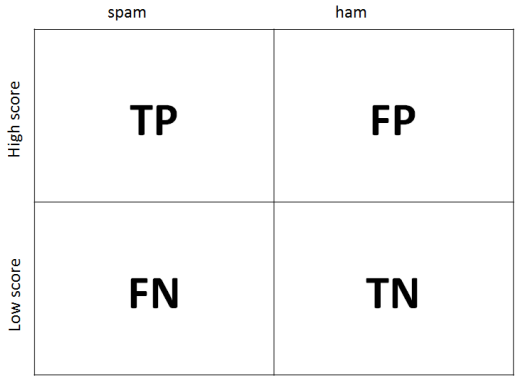


Figure 6: Confusion Matrix

5.3.2 Receiver Operating Characteristic(ROC) Curve

For any binary classifier, ROC curve is constructed by plotting True Positive Rate(TPR) versus False Positive Rate(FPR) for varying threshold values. True Positive Rate(TPR) is also called sensitivity, True Negative Rate(TNR) is also called specificity. FPR = 1 - specificity. TPR and TNR can be defined as follows-

$$\textbf{TPR} = \frac{TP}{TP + FN} \textbf{ and } \textbf{TNR} = \frac{TN}{TN + FP}$$

Area Under the Curve (AUC) for an ROC is used as a scoring metric. An AUC of 1.0 is perfect accuracy and AUC of 0.5 is like flipping a coin. Fig. 7. shows an example of an ROC curve.

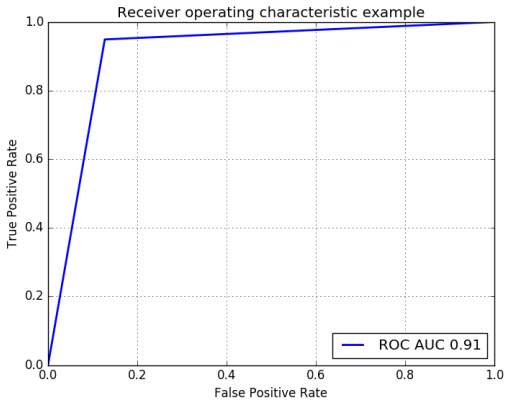


Figure 7: ROC Curve

CHAPTER 6

Environmental Setup

All the experiments were conducted in a Windows 7 Machine with 8 GB RAM and 256 GB SSD. We chose Python as our primary language. Python 3.5.0 with open-cv were primarily used for all image processing tasks. Table 2 lists all the python packages used and their purpose.

Table 2: Python Packages

Library	Purpose
Open-cv	Image Processing for feature extraction
PIL	Image Processing for feature extraction
Scikit-learn	SVM and preprocessing
Numpy	Mathematical computations like mean, var
Matplotlib	Charting

CHAPTER 7

Synthetic Dataset Generation

The aim of generating this dataset was to challenge the existing detection scheme. Image properties between ham and spam images vary. We used Image Processing techniques on spam images, to make it look more like a ham image. A public corpus; Spam Archive, from Dredze et. al in their paper Learning Fast Classifiers [8] included only spam images. We used this corpus and overlayed it on the ham images from Dataset 1. The resulting spam images were harder to detect.

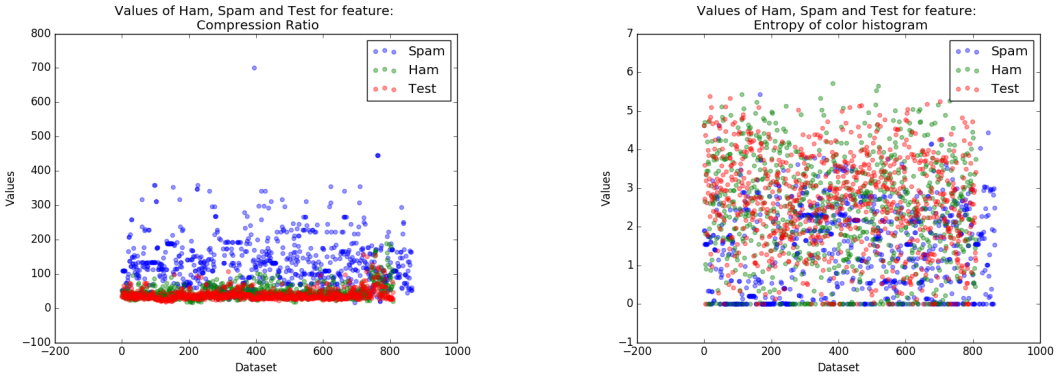
There are two key steps to generating this test dataset-

- First, we resize the spam image to the dimensions of ham image. This resizing helps align the file properties of the test dataset to that of ham set.
- Second, we overlay spam images on top of ham images. A general observation we made with the spam images, was that spam images have a light (white/yellow) background. Eliminating the background, we picked up only the content of the spam image and overlaid it on ham image. Doing so, helped us align many image properties like color histogram and edges with that of ham images.

Fig. 8. shows an example of the generated dataset. We can see that the test image(generated image), has the ham image as the background and the content of the spam image as the foreground.

Fig. 9. shows scatterplots of compression ratio and color entropy values for ham, spam and test images. It is easy to note from these scatterplots how the properties of ham and test image align. Appendix A lists scatter plots of rest of the features.

Figure 8: Test dataset example



(a) Compression Ratio (b) Entropy of color histogram

Figure 9: Feature value comparison scatter plots

CHAPTER 8

Support Vector Machines (SVM) - Experiments

SVM has been widely used in text based detection techniques [1]. In this section we will analyze how SVM can be used in image spam detection. SVM is a supervised classification algorithm. SVM generates a separating hyper-plane at the end of the training phase which separates our data into two classes [14].

8.1 Experiments

Fig. 10. shows the flow of train and test phases for the SVM Detection Model. First the ham and spam data is split into train and test sets. Train and test sets are exclusive i.e. there is no overlap between the two. 41 features are then extracted from the datasets. We then train the SVM Classifier with scaled train data. Test set is then passed to the SVM Classifier for detection. Additionally, in the train phase, feature selector is added to perform dimensionality reduction based on feature weights.

To analyze the weight of each feature we calculated SVM scores for each feature individually. Fig. 11. shows SVM scores for individual features for all the three datasets. It is easy to note from the three graphs that the SVM AUC scores for individual features for test dataset has gone down significantly compared to those of Dataset 1 and 2.

8.1.1 Dataset 1

All 41 features were extracted and scaled for dataset 1. We used 6% of ham and spam images as training set and rest for testing. A total of 55 spam and 48 ham images were used as train objects. Remaining 865 spam images and 762 ham images

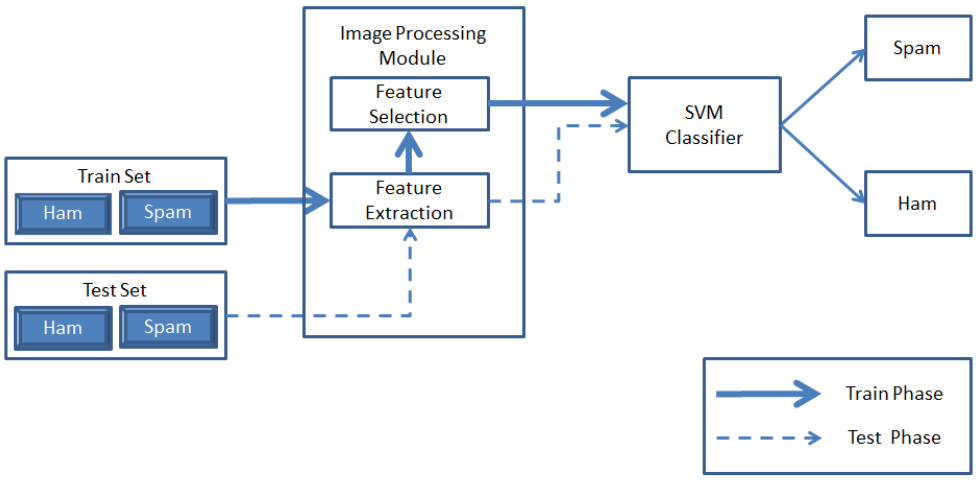


Figure 10: SVM Detection Model

were used for testing. Table 3 shows the accuracies and False Positive Rates for each of the three SVM Kernels. We achieved best results for Linear Kernel. Fig. 12. shows the ROC curve and confusion matrix for linear kernel.

Table 3: Dataset 1 - SVM Results

Kernel	Accuracy	FPR
Linear	0.97	0.06
RBF	0.96	0.07
Poly	0.95	0.08

8.1.2 Dataset 2

All 41 features were extracted and scaled for our in-house generated test dataset. We used 45% of ham and spam images as training set and rest for testing. A total of 490 spam and 463 ham images were used as train objects. Remaining 599 spam images and 566 ham images were used for testing. Table 4 shows the accuracies and False Positive Rates for each of the three SVM Kernels. We achieved similar results

for Linear and rbf kernels.

Table 4: Dataset 2 - SVM Results

Kernel	Accuracy	FPR
Linear	0.98	0.02
RBF	0.98	0.02
Poly	0.95	0.10

8.1.3 Dataset 3

All 41 features were extracted and scaled for our in-house generated test dataset. We used 30% of ham and spam images as training set and rest for testing. A total of 243 spam and 243 ham images were used as train objects. Remaining 567 spam images and 567 ham images were used for testing. Table 5 shows the accuracies and False Positive Rates for each of the three SVM Kernels. We achieved best results for Linear Kernel.

Table 5: Test Dataset - SVM Results

Kernel	Accuracy	FPR
Linear	0.70	0.38
RBF	0.64	0.34
Poly	0.56	0.78

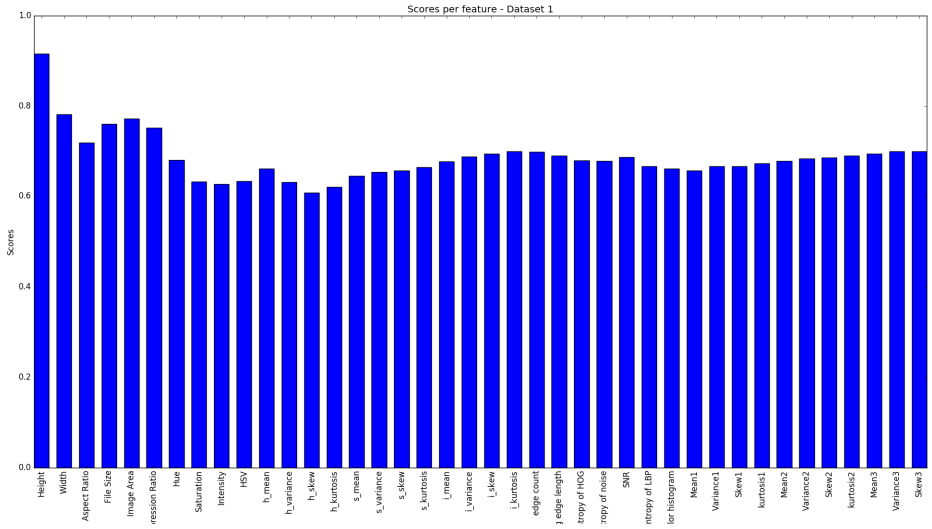
8.2 Feature Selection

8.2.1 Recursive Feature Elimination

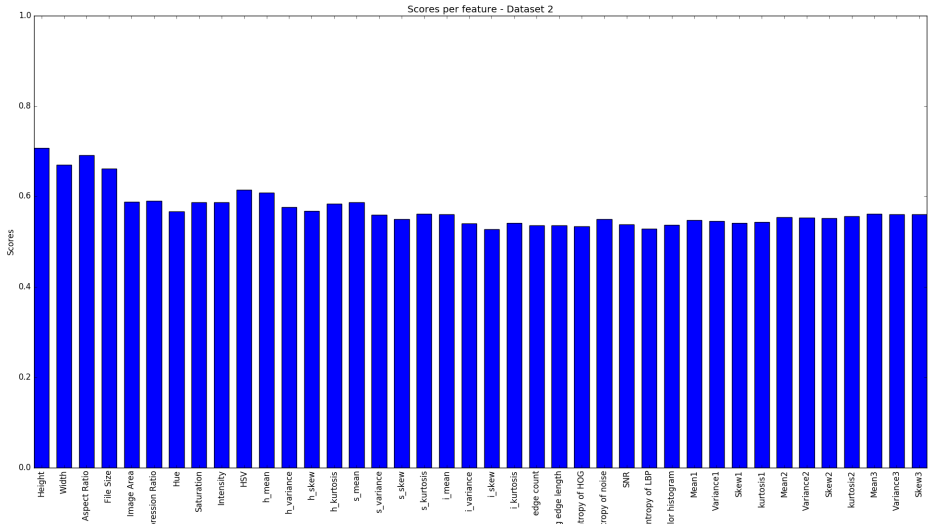
Recursive Feature Elimination (RFE) is a feature selection technique. It is used to get rid of the features that contribute the least to the classification. Here, we use RFE to fine tune the SVM classification for unraveling ham and spam images. RFE assigns weights to features and ranks them according to the amount of contribution towards the classification; and eliminates the least ranked features to enhance the

accuracy of SVM.

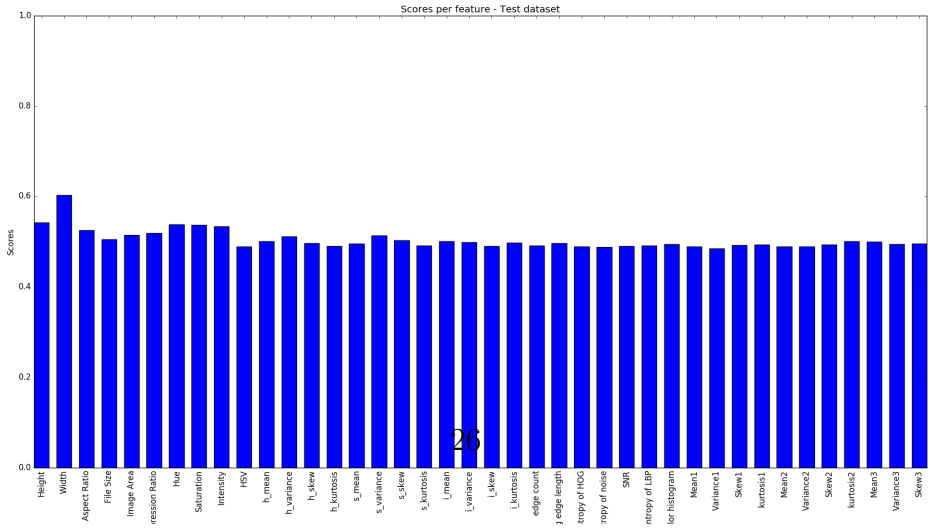
8.2.2 Univariate Feature Selection



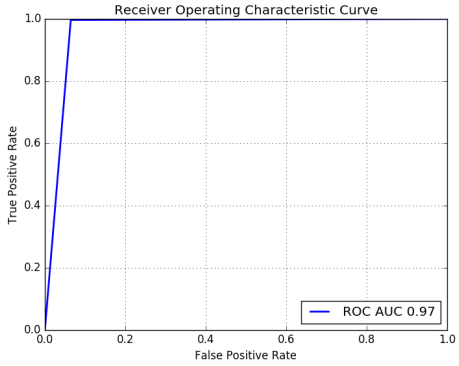
(a) Dataset 1 AUC scores for Individual Features



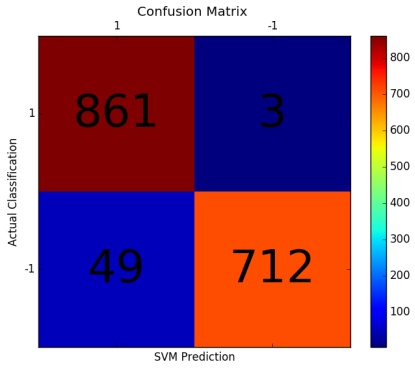
(b) Dataset 2 AUC scores for Individual Features



(c) Test Dataset AUC scores for Individual Features

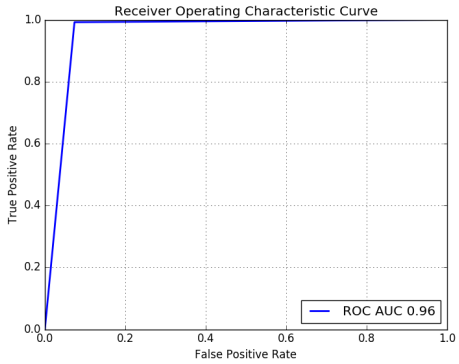


(a) ROC Curve

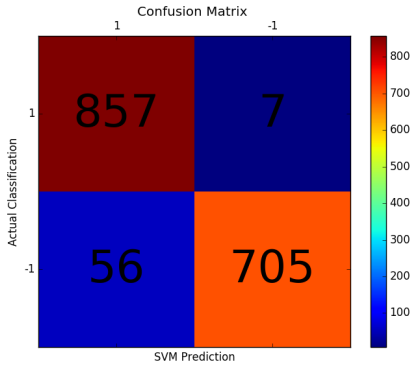


(b) Confusion Matrix

Figure 12: Dataset 1 - Linear Kernel

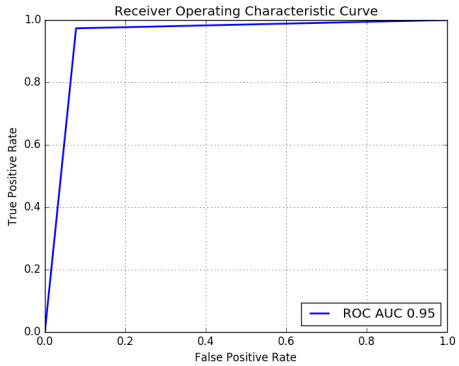


(a) ROC Curve

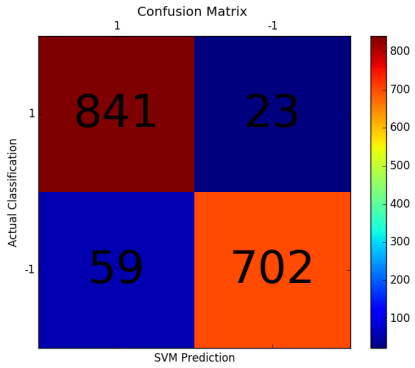


(b) Confusion Matrix

Figure 13: Dataset 1 - RBF Kernel

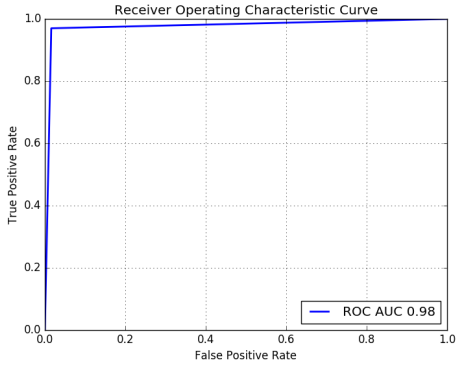


(a) ROC Curve

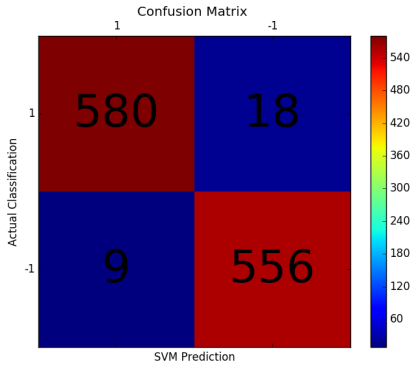


(b) Confusion Matrix

Figure 14: Dataset 1 - Polynomial Kernel

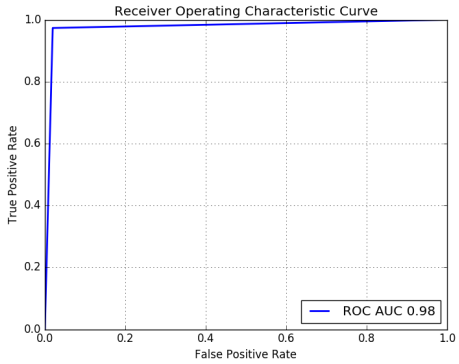


(a) ROC Curve

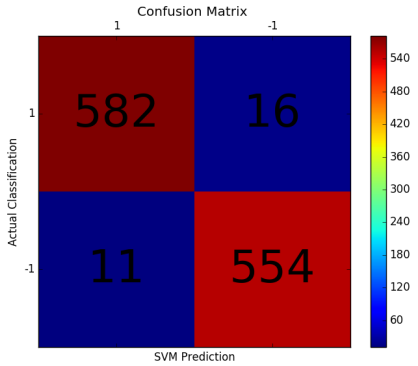


(b) Confusion Matrix

Figure 15: Dataset 2 - Linear Kernel

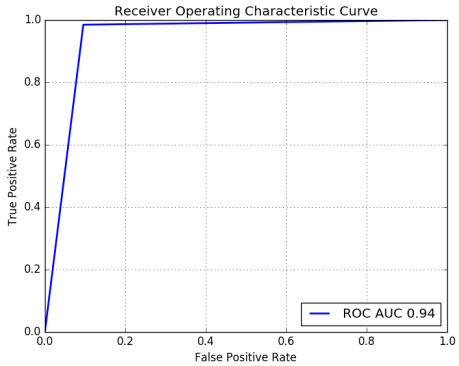


(a) ROC Curve

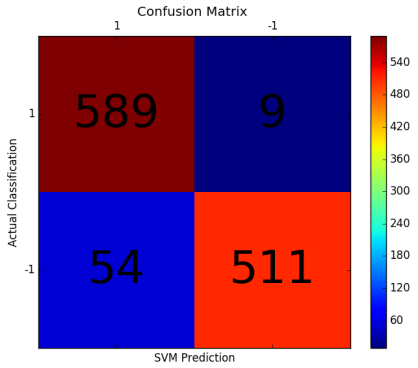


(b) Confusion Matrix

Figure 16: Dataset 2 - RBF Kernel

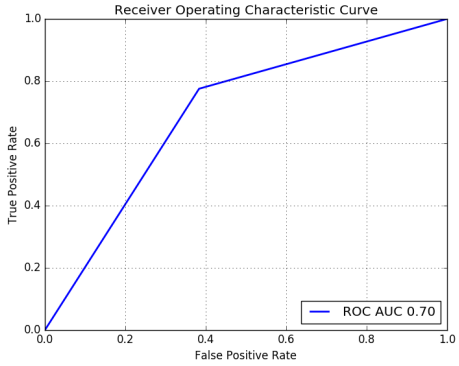


(a) ROC Curve

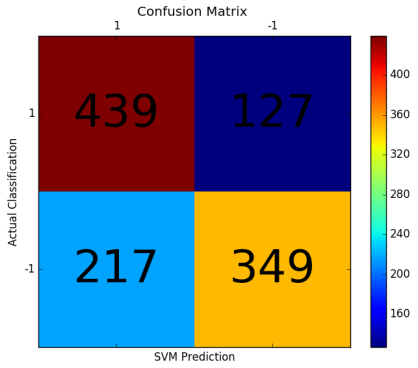


(b) Confusion Matrix

Figure 17: Dataset 2 - Polynomial Kernel

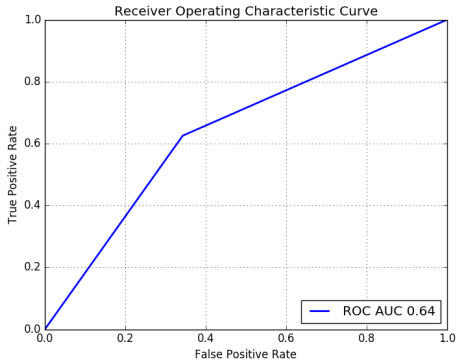


(a) ROC Curve

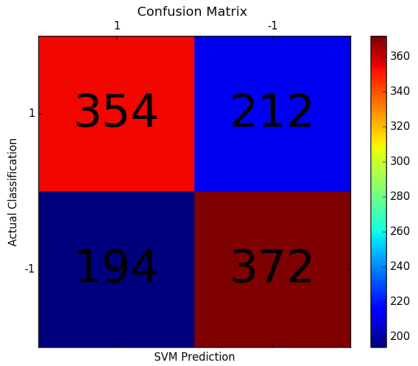


(b) Confusion Matrix

Figure 18: Test Dataset - Linear Kernel

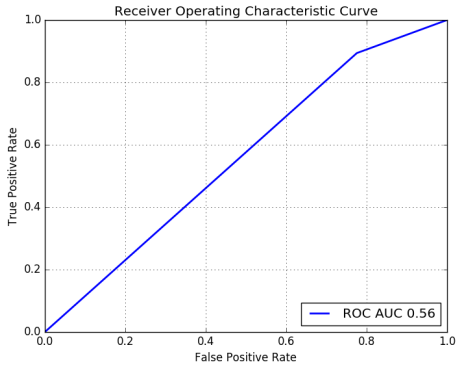


(a) ROC Curve

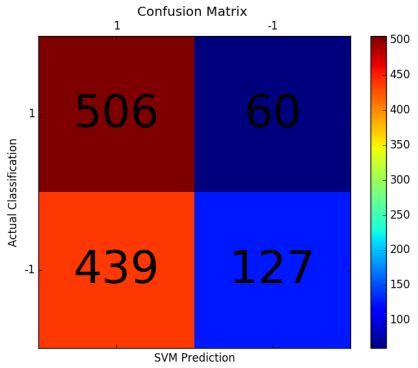


(b) Confusion Matrix

Figure 19: Test Dataset - RBF Kernel



(a) ROC Curve



(b) Confusion Matrix

Figure 20: Test Dataset - Polynomial Kernel

CHAPTER 9

Conclusion and Future Work

LIST OF REFERENCES

[1] S. Dhanaraj and V. Karthikeyani, "A study on e-mail image spam filtering techniques," in *2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering*, Feb 2013, pp. 49–55. [Online]. Available: http://ieeexplore.ieee.org/xpls/icp.jsp?arnumber=6496446&tag=1

[2] C.-C. Lai and M.-C. Tsai, "An empirical performance comparison of machine learning methods for spam e-mail categorization," in *Fourth International Conference on Hybrid Intelligent Systems, 2004. HIS'04.* IEEE, 2004, pp. 44–48. [Online]. Available: http://ieeexplore.ieee.org.libaccess.sjlibrary.org/stamp/stamp.jsp?arnumber=1409979

[3] Y. Gao, M. Yang, X. Zhao, B. Pardo, Y. Wu, T. N. Pappas, and A. Choudhary, "Image spam hunter," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing.* IEEE, 2008, pp. 1765–1768. [Online]. Available: http://www.cs.northwestern.edu/~yga751/ML/ISH.htm

[4] L. Whitney, "Report: Spam now 90 percent of all e-mail," *CNET News*, vol. 26, 2009. [Online]. Available: https://www.cnet.com/news/report-spam-now-90-percent-of-all-e-mail/

[5] S. Assassin, "The apache spamassasin project," Aug 2005. [Online]. Available: http:/spamassasin.apache.orgs

[6] A. Annadatha and M. Stamp, "Image spam analysis and detection," *Journal of Computer Virology and Hacking Techniques*, vol. 23, pp. 1–14, 2016. [Online]. Available: http://dx.doi.org/10.1007/s11416-016-0287-x

[7] T. Kumaresan, S. Sanjushree, K. Suhasini, and C. Palanisamy, "Image spam filtering using support vector machine and particle swarm optimization." [Online]. Available: http://research.ijcaonline.org/nciprc2015/number1/nciprc8006.pdf

[8] M. Dredze, R. Gevaryahu, and A. Elias-Bachrach, "Learning fast classifiers for image spam." in *CEAS*, 2007. [Online]. Available: https://www.cs.jhu.edu/~mdredze/datasets/image_spam/

[9] M. Soranamageswari and C. Meena, "Statistical feature extraction for classification of image spam using artificial neural networks," in *2010 Second International Conference on Machine Learning and Computing*, Feb 2010, pp. 101–105. [Online]. Available: http://ieeexplore.ieee.org/document/5460761/

[10] G. Fumera, I. Pillai, and F. Roli, "Spam filtering based on the analysis of text information embedded into images," *Journal of Machine Learning Research*, vol. 7, pp. 2699–2720, Dec 2006. [Online]. Available: http://www.jmlr.org/papers/v7/fumera06a.html

[11] M. Chowdhury, J. Gao, and M. Chowdhury, *Image Spam Classification Using Neural Network*. Australia: Springer International Publishing, 2015, pp. 622–632. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-28865-9_41

[12] M. Nixon, *Feature extraction and image processing*. Academic Press, 2008. [Online]. Available: http://vlm1.uta.edu/~patjang/cvpr2012/Books/feature-extraction-image-processing-second-edition.pdf

[13] M. Stamp, "*Machine Learning with Applications in Information Security*," unpublished manuscript.

[14] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995. [Online]. Available: http://dx.doi.org/10.1007/BF00994018

APPENDIX

Feature value comparison scatter plots for test dataset

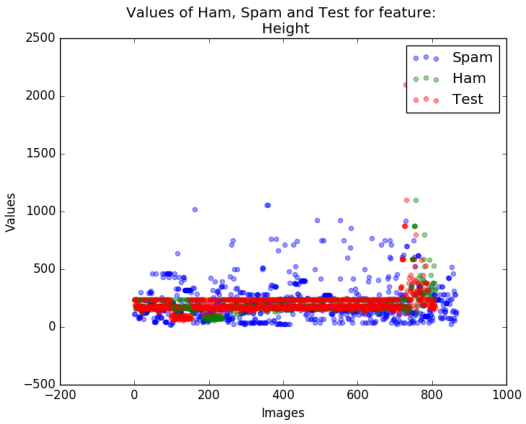


Figure A.21: Height

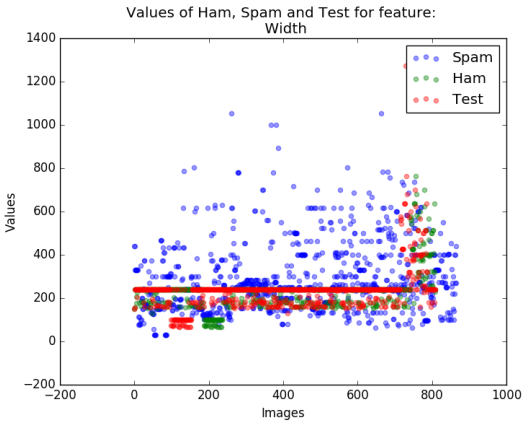


Figure A.22: Width

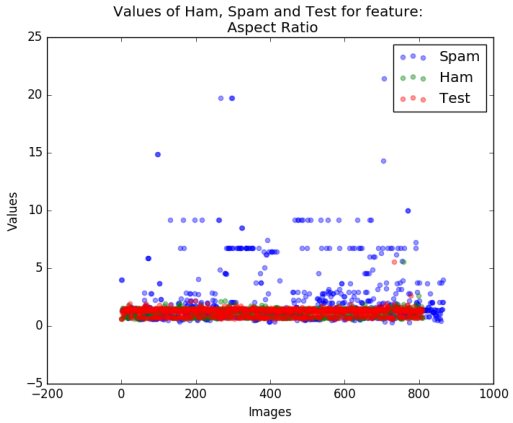


Figure A.23: Aspect Ratio

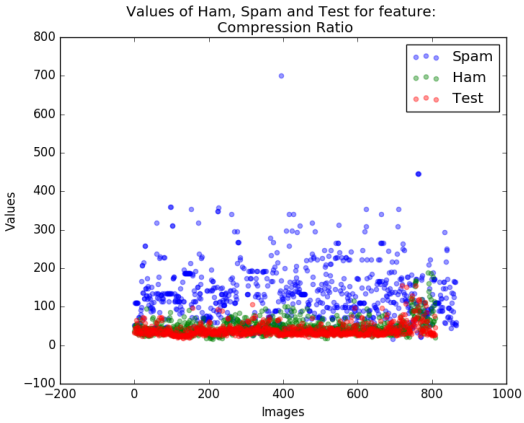


Figure A.24: Compression Ratio

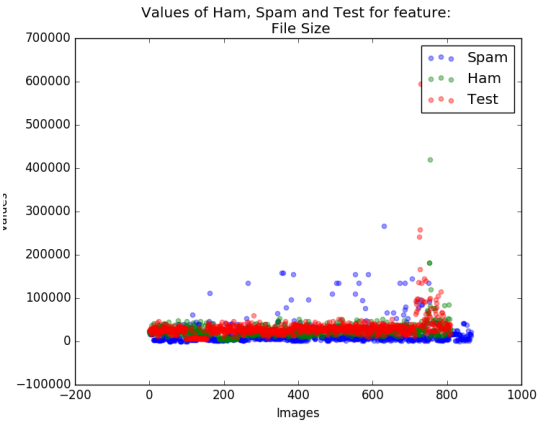


Figure A.25: File Size

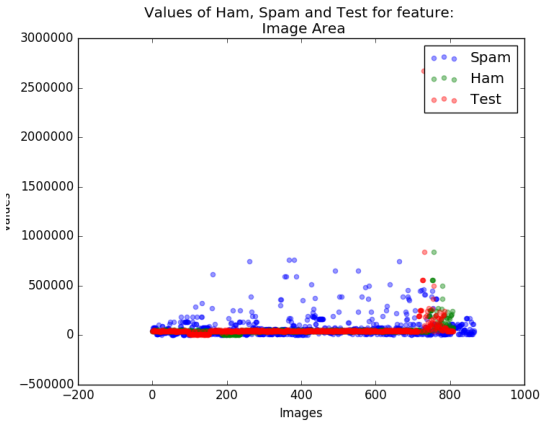


Figure A.26: Image Area

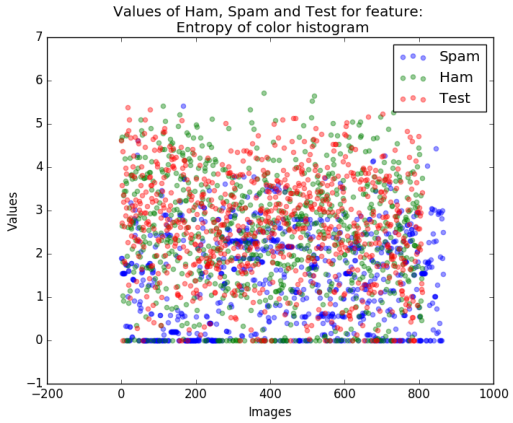


Figure A.27: Entropy of color histograms

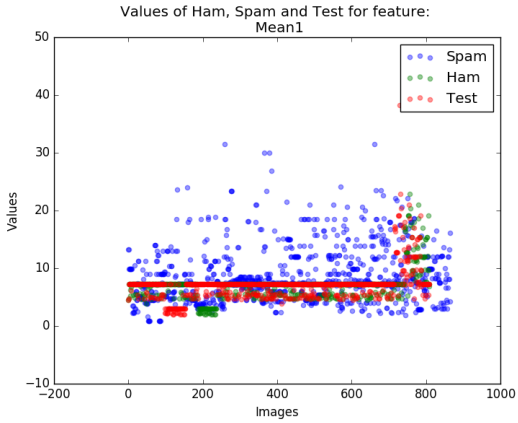


Figure A.28: Red channel mean

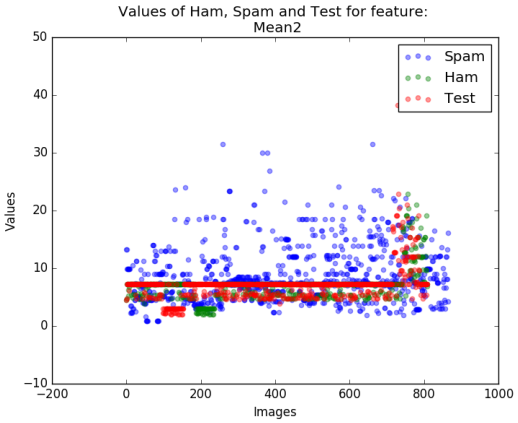


Figure A.29: Green channel mean

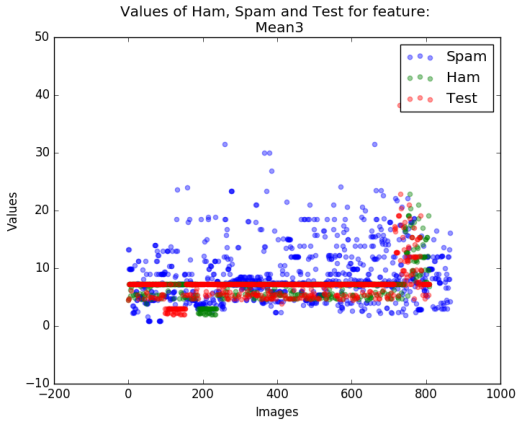


Figure A.30: Blue channel mean

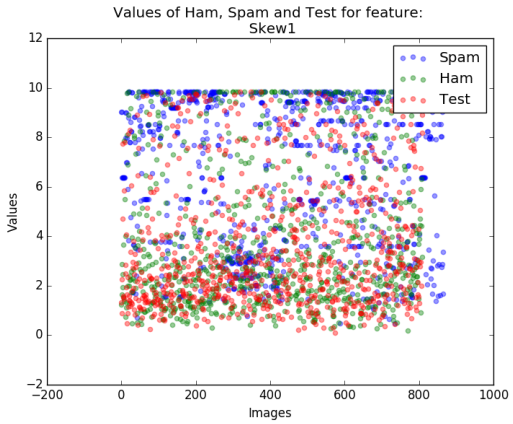


Figure A.31: Red channel Skew

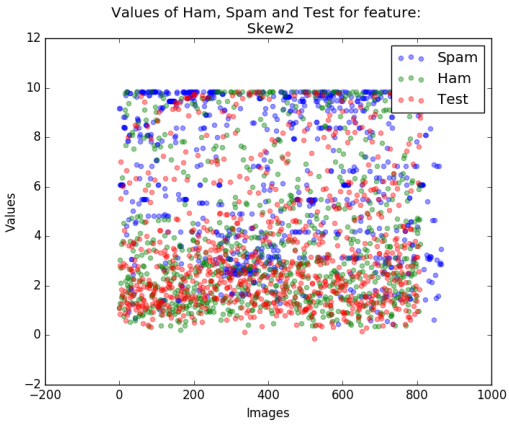


Figure A.32: Green channel skew

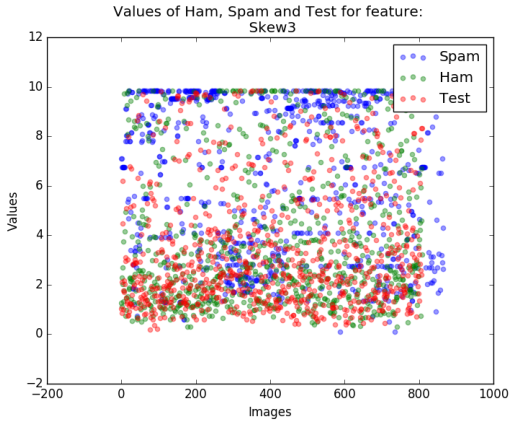


Figure A.33: Blue channel Skew

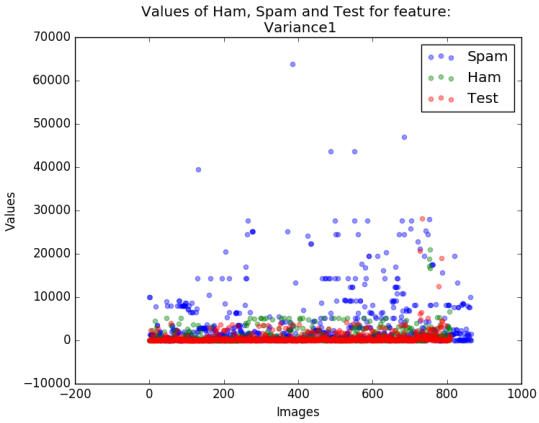


Figure A.34: Red channel Variance

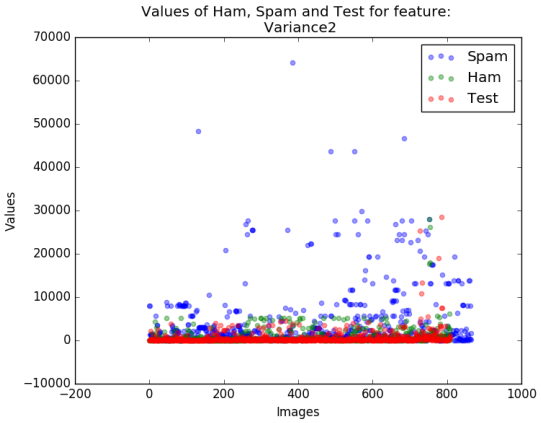


Figure A.35: Green channel Variance

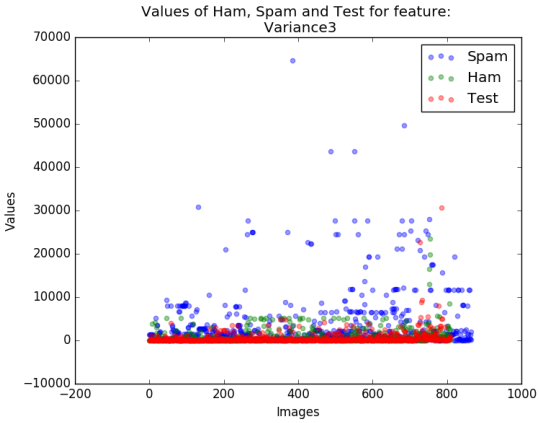


Figure A.36: Blue channel Variance

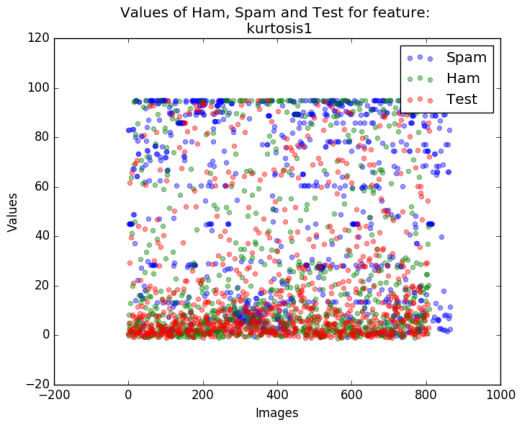


Figure A.37: Red channel Kurtosis

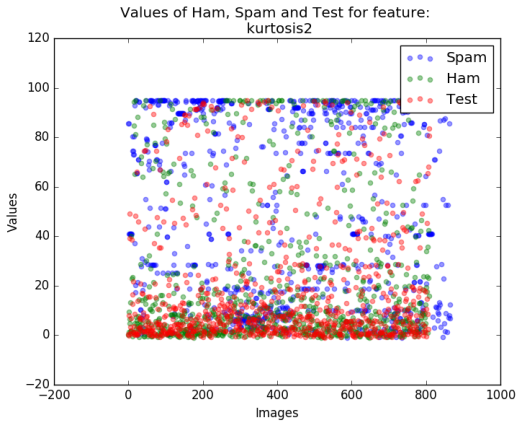


Figure A.38: Green channel Kurtosis

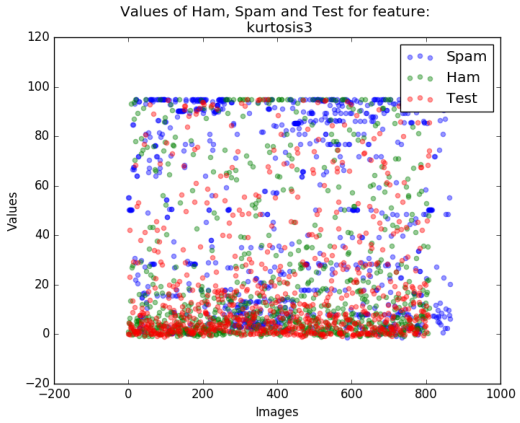


Figure A.39: Blue channel Kurtosis

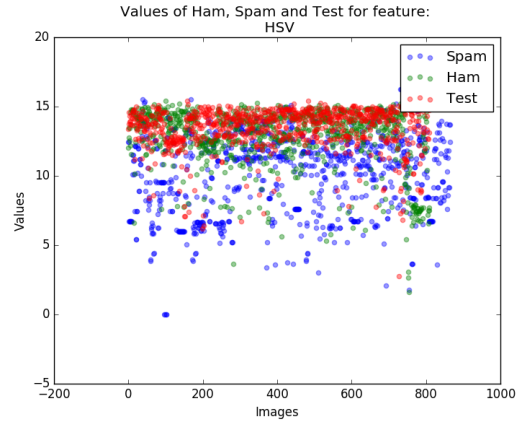


Figure A.40: Entropy of HSV

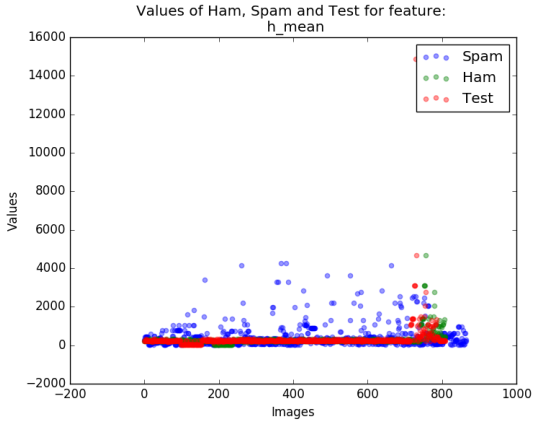


Figure A.41: Hue channel mean

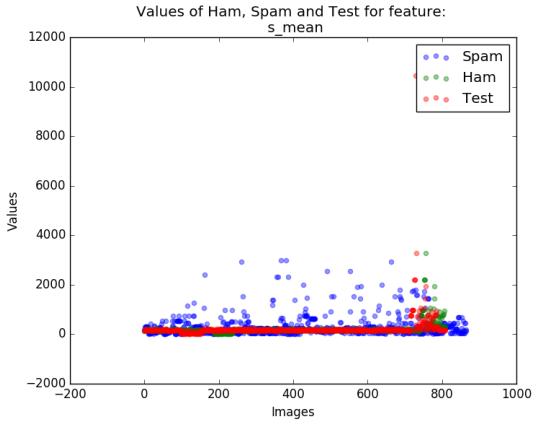


Figure A.42: Saturation channel mean

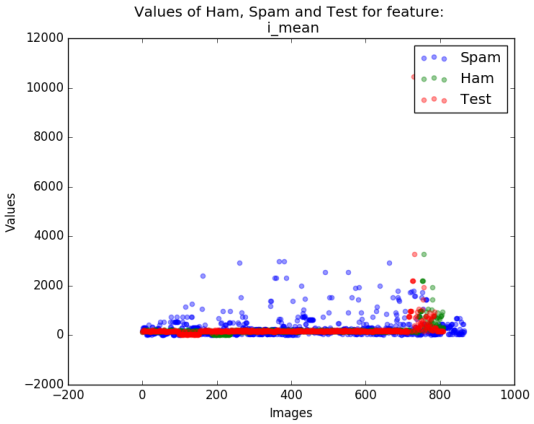


Figure A.43: Intensity channel mean

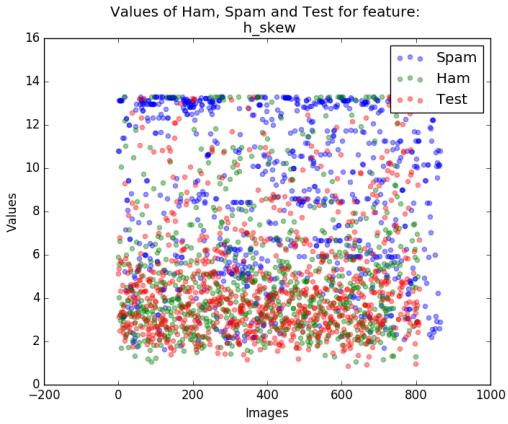


Figure A.44: Hue channel Skew

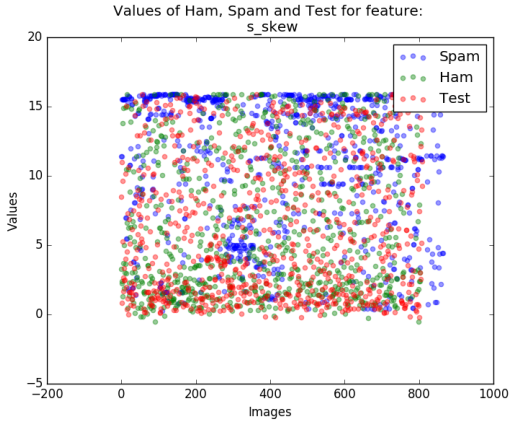


Figure A.45: Saturation channel Skew

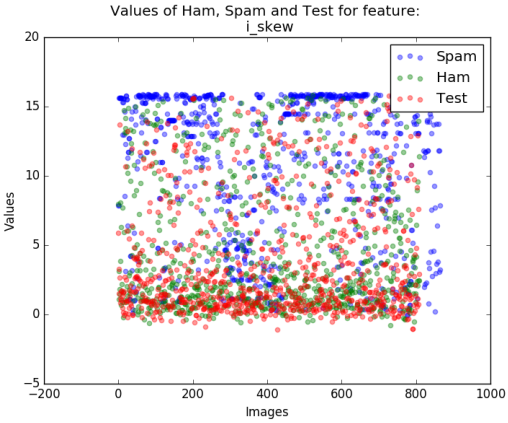


Figure A.46: Intensity channel skew

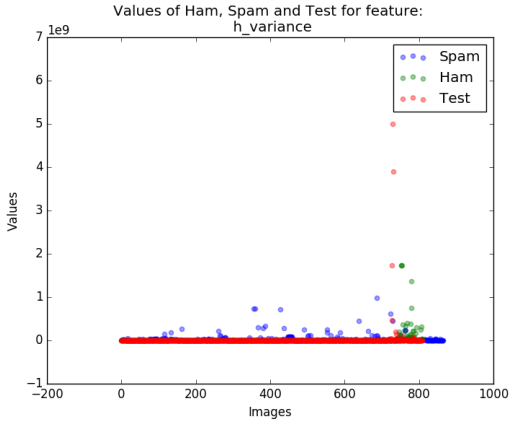


Figure A.47: Hue channel Variance

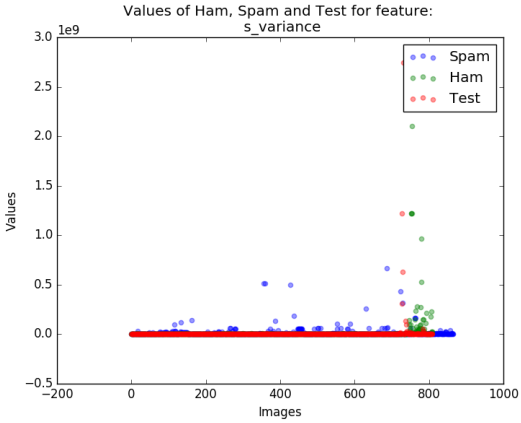


Figure A.48: Saturation channel Variance

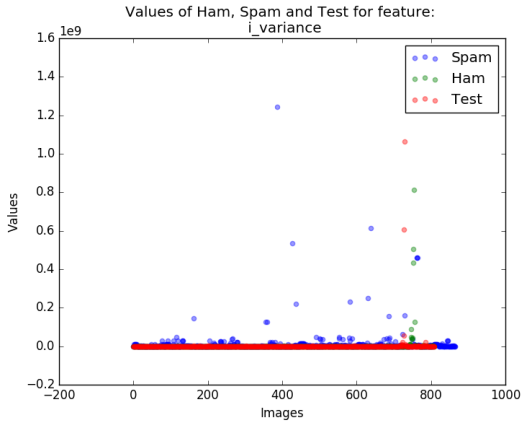


Figure A.49: Intensity channel Variance

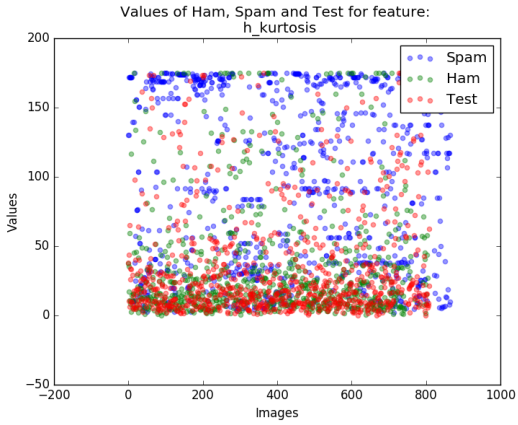


Figure A.50: Hue channel Kurtosis

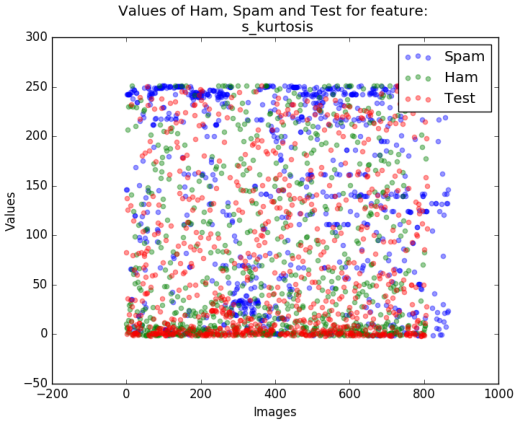


Figure A.51: Saturation channel Kurtosis

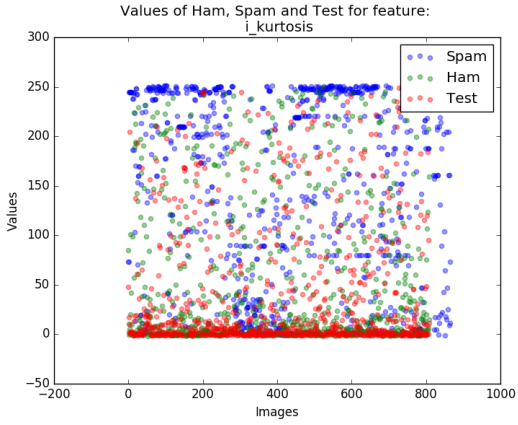


Figure A.52: Intensity channel Kurtosis

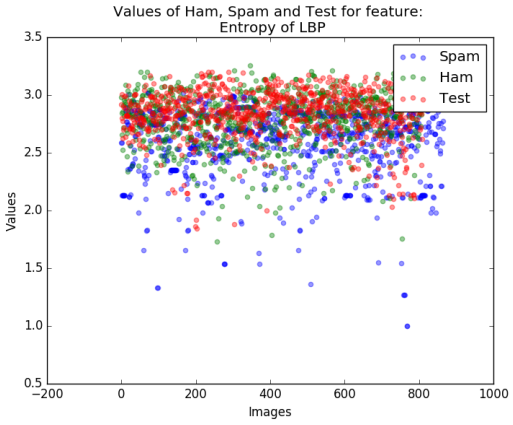


Figure A.53: Entropy of Local Binary Pat-tern

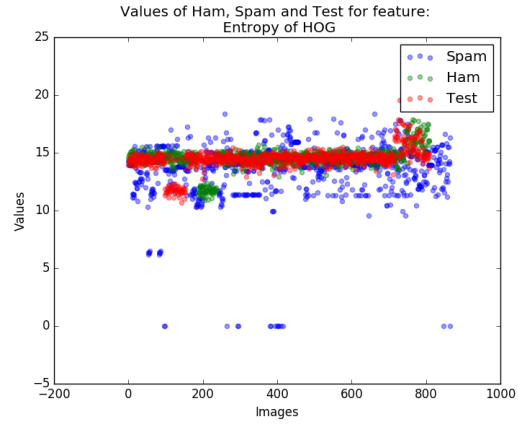


Figure A.54: Entropy of Histogram Of Gradients

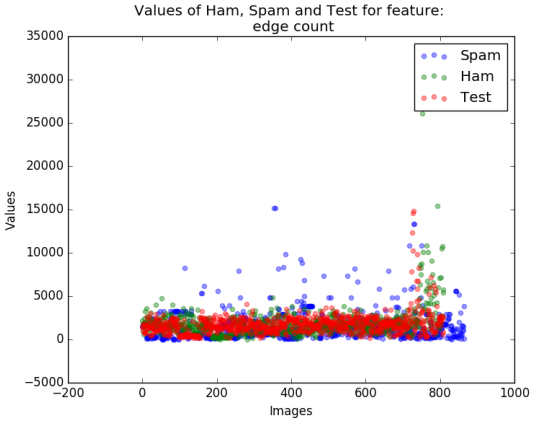


Figure A.55: Edge Count

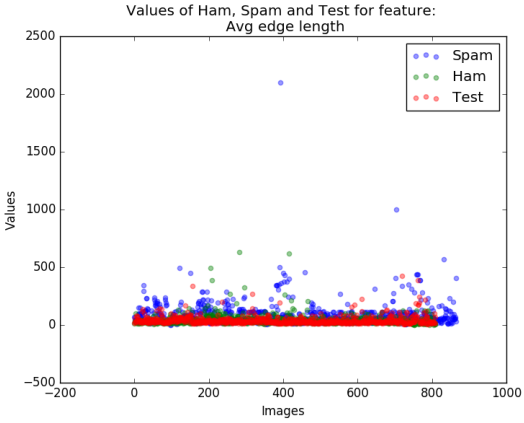


Figure A.56: Average Edge Length

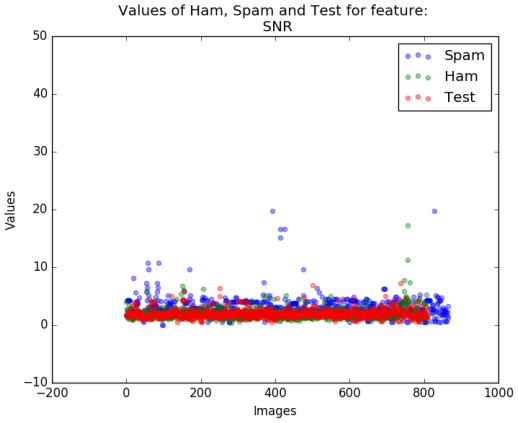


Figure A.57: Signal to Noise Ratio

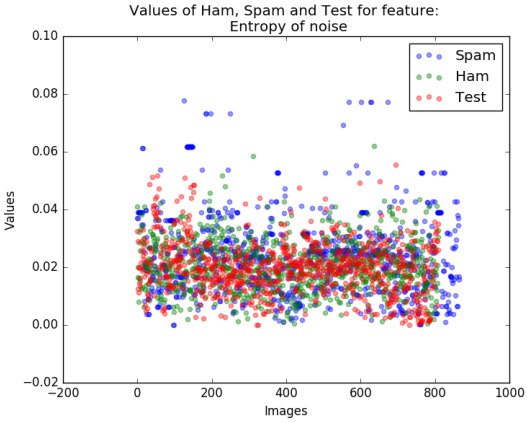


Figure A.58: Entropy of Noise