Tampered Image Detection using SVM Classifier

by

Dhwani Patel
201511060

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

MASTER OF TECHNOLOGY
in
INFORMATION AND COMMUNICATION TECHNOLOGY
to

DHIRUBHAI AMBANI INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY

April, 2017

Declaration

I hereby declare that

i) the thesis comprises of my original work towards the degree of Master of Technology in Information and Communication Technology at Dhirubhai Ambani Institute of Information and Communication Technology and has not been submitted elsewhere for a degree,

ii) due acknowledgment has been made in the text to all the reference material used.

Dhwani Patel

Certificate

This is to certify that the thesis work entitled Tampered Image Detection using SVM classifier has been carried out by Dhwani Patel for the degree of Master of Technology in Information and Communication Technology at *Dhirubhai Ambani Institute of Information and Communication Technology* under my/our supervision.

Dr. Maniklal Das
Thesis Supervisor

Acknowledgments

I take such golden opportunity to express a deep sense of appreciation towards my guide Dr. Manik Lal Das for providing immense motivation and guidance throughout my research work.Without his inestimable guidance and inspiration, this work would never have reached a path of success.

I would also like to thank respected faculty members of my evaluation committee Dr. Gagan Garg and Dr. Manish Narwaria of the DA-IICT for their valuable comments and encouragement throughout my research work.

I am also thankful to Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar for providing me an infrastructural assistance in the form of research lab which helped me a lot to complete my work incomparably.

Finally I express my sense of gratitude towards my parents and friends for supporting me throughout my research.

Dhwani Patel

Contents

Abstract

Steganography is the process of hiding confidential information in image such that contents of original image remain unaltered. Hence steganalysis algorithms used to detect such data embedding needs to be designed. In this work, features are designed to classify the given image as raw image (cover image) or image containing hidden data (stego image) embedded using LSB matching steganography algorithm. Finally, support vector machine classifier is trained using designed features. Two set of features are designed i.e one based on histogram of image and other based on information theoretic measure such as mutual information. Histogram of image is analyzed using short time Fourier transform and features based on centre of mass (COM) in frequency domain is designed. Statistical dependency between adjacent pixels in natural images is quantified using mutual information and novel features are designed based on that.

Corel database containing 10,000 images is used for evaluating the proposed algorithm. Using this database, 20,000 images are made of same size out of which 10,000 are cover images and 10,000 are stego images. 85.71 % classification accuracy on the test set is obtained which is a significant improvement over the previously reported techniques.

List of Principal Symbols and Acronyms

LSB	Least Significant Bit
ROC	Receiver Operating Characteristic
bpp	Bits Per Pixel
i.i.d	Independent and Identically Distributed
PMF	Probability Mass Function
SVM	Support Vector Machine
HCF	Histogram Characteristics Function
COM	Center of Mass
DFT	Discrete Fourier Transform
STFT	Short Time Fourier Transform
MI	Mutual Information
RBF	Radial Basis Function
JDF	Joint Distribution Function

List of Tables

List of Figures

CHAPTER 1
Introduction

Steganography is to hide secret information in an image such that the presence of that secret data cannot be detected. The sources of hiding secret data can be any digital image, text, any video or audio. Secret data hidden inside any image is then secretly transmitted to the receiver. This secret hiding of data is considered as manipulation of original images which can be easily done with the help of advanced computers and improved photo-editing software tools. Steganalysis goal is to expose the secret data that has been hidden using any steganographic technique. To detect tampering in images without having any prior knowledge regarding image and its content has emerged as an important part of the research field.

With the advancement in present day digital technology, the way of sharing, accessing and manipulating information has changed manifold thus giving rise to variety of different security issues. The task of modifying images has become usual practice due to advancement in digital technology and various photo-editing software. Thus in order to create any kind of digital forgery, it has became extremely mandatory to manipulate images. This manipulation occurring in images has destroyed the trust in digital image technology.

A kind of image forgery was observed in a Tunisian newspaper in which a photograph was modified in such a way that the crowd appeared larger as compared to original image [20]. Another case as seen in Figure 1.1 shows the photograph posted by Iran which was changed in such a way that four missiles were seen in an image but in originality, there was only three present. This manipulated image case was published by media people all over the world. The research in this thesis seeks to focus towards the very need of powerful detector that can detect digital forgery and provide some intuition towards this demanding problem.

1.1 Image Tampering

Image tampering is to add or remove some contents from an image such that no detectable copy of tampering is found and thus tampering in image is marked as the preconceived

(a) Original Image (b) Tampered Image

Figure 1.1: Copy and move forgery

alteration in images for mischievous purposes [5]. There are various methods related to tampering in images which can be classified as:

- **Copy-Move attack:** It is a kind of forgery, in which a portion of image is copied from one location and pasted to other location in order to conceal some contents of image or remove undesired image contents. Textured portions have similar color, dynamic range and noise variation property and hence are used for copy-move forgery which proves difficult for human eye to detect those imperfections in images. Blurring is done considering border portions of the tampered regions such that the effect of deformity amongst tampered regions to that of original regions is reduced.

- **Image Splicing:**This technique combines multiple images to create a new manipulated image. Thus spliced image is obtained by merging any photographic images.

- **Image-Retouching:**In this technique selective features of image are added or removed such that it results in an attractive image. This technique is proved less harmful forgery and hence it is mostly used by magazine editors.

1.2 Techniques to Counterattack Forgery

To counterattack tampering in images, two ways are considered that are the Active and Passive approach. In active approach, techniques like watermarking or signature identification are applied during the creation of image. This approach is helpful in detecting

authenticity for images as there are huge variety of images available on the internet having neither digital watermark nor signature.

In passive approach, any embedded watermark or digital signature is not required. The techniques commonly used to manipulate digital images are Copy and Move, image re-touching and splicing as shown in figure 1.2.

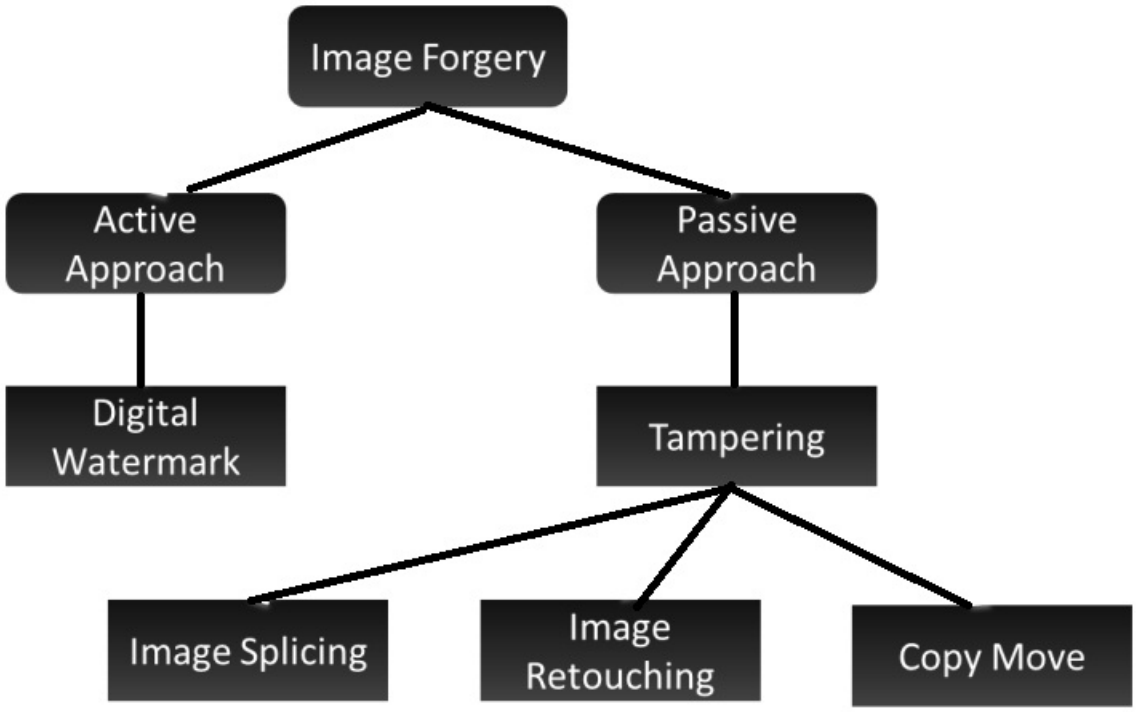


Figure 1.2: Categorization Of Forgery in images

1.3 Problem Definition

Image steganalysis algorithm is designed to classify query image as raw image (cover image) or image containing some hidden secret data (stego image). The raw image (cover image) is tampered using LSB matching steganography which provides us with a stego image. The features are designed to train SVM (Support Vector Machine)[13] model. It classifies whether an image is corrupted or not. Hence we discuss about various features designed and its accuracy of predicting the detection problem.

LSB matching is interpreted as the addition of noise. The addition of noise in spatial domain results in low pass filtering of image histogram. Thus stego histogram is less steeper as compared to cover histogram of image due to filtering that occurs after secret embedding. This results in decrease of COM of F(h),that represents fourier transform for histogram function after embedding secret data using LSB matching steganography.

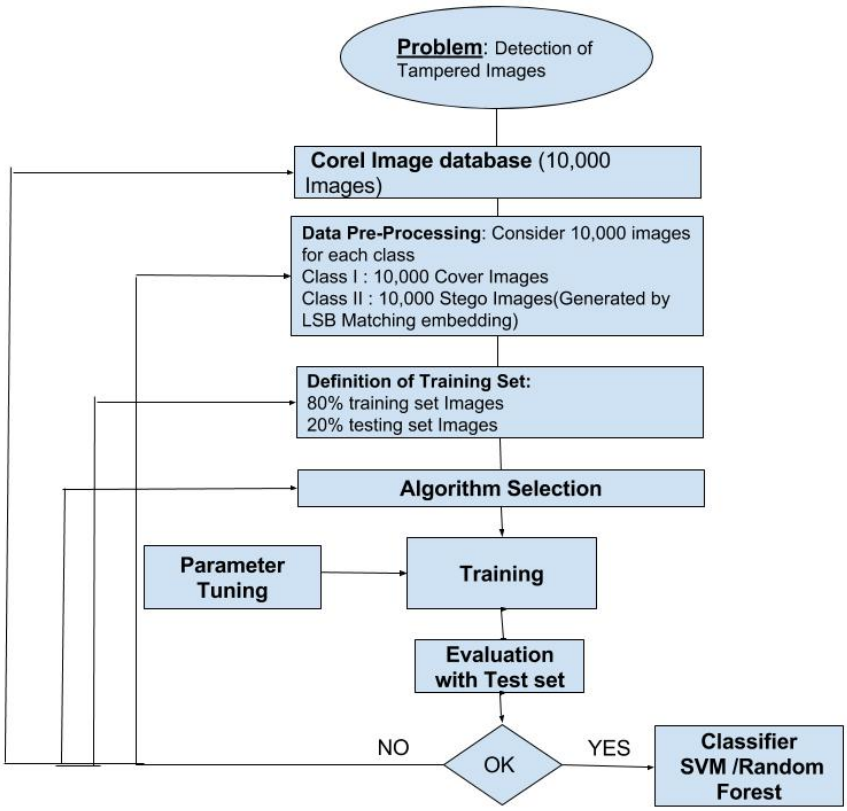


Figure 1.3: Tampered Image detection problem Flowchart

This idea is considered to construct a discriminating feature used in detecting tampering in images.

Ker[15] presented two different ideas of utilizing the histogram function(HCF) [15] that are :

- calibration by down sampled image
- computing adjacency histogram rather than intensity histogram

These improved features provided better results in detecting stego image.

The computation of DFT of histogram of image and downsampled image provide excellent frequency localization. The computation of STFT of image histogram and downsampled image histogram over small patches (narrow window size) of histogram of image provides excellent time localization(window size is infinitely short). The STFT of histogram of image is taken over patches which make it easy to capture localized variations caused due to tampering in images. This new feature compared to previously designed features, trains our model to achieve more reliable detection accuracy.

Mutual information is quantization of information of one random variable through the other. In any natural image, there exists dependency between neighboring pixels to some extent. Due to this high dependency among adjacent pixels the measure of mutual

information is high. The measure of mutual information is low in stego image due to reduced dependency. Hence these variations are captured through newly designed features that act as a discriminator for detecting steganography in images.

The above mentioned detectors proved to enable reliable detection of hidden data embedded inside cover Image.

1.4 Organization of the Report

In the 2^{nd} chapter, we will discuss steganography in images using LSB matching algorithm and effects of staganographic embedding on image histogram.Further down-sampling in images and variation in center of mass of HCF of image is studied. In the 3^{rd} chapter by analysis and experiments we designed steganalysis algorithm based on STFT and mutual information for robust tampered image detection rates. In 4^{th} chapter, we compare performance of our designed features and algorithm with already existing ones. Finally, 5^{th} chapter includes conclusion and future work.

1.5 Literature survey

LSB matching is interpreted as an additive noise. J.J. Harmsen et al. [10] presented that addition of noise in space domain provides us with low pass filtered intensity histogram.Thus stego histogram is less steep than cover histogram that occured due to filtering.Thus cover histogram has more variations due to which COM value is high. After steganographic embedding stego histogram is smoothened due to which fewer variations are seen hence COM reduces. This property is used as one of the distinguishing features for detecting stego image. It provides good results on color histograms while performs poorly on grayscale images. Ker[15] suggested two different ways of applying the histogram characteristic function (HCF) idea and that is based on using down sampled image instead of original image and considering adjacency in place of intensity histogram.

This improvement in features provided better results than before in detecting stego Work on grayscale images. The authors of [15] presented a method that used a high pass FIR filter. Here message length is recovered using maximum likelihood estimator[14]. This idea proved ineffective when applied on uncompressed images that come from the scanner. Holotyak et al. [11] and fridrich et al.[8] suggested a method based on classifying features that are derived by estimating stego image in the wavelet domain. Further Goljan et al.[9] presented a concept based on absolute moments of the noise residues.

The steganalyzers mentioned above proved to be poor detectors when a secret message was hidden using LSB matchings steganography done on grayscale images containing

good amount of high-frequency noise. This occurs because image noise masks the additive stego signal. Here it becomes very difficult to differentiate amongst the stego signal and already present noise in images. This naturally occurring noise in images is interpreted as a stego signal which wrongly proves that image has some hidden data. Thus new features were designed to address this issue based on the observation that the local maximum values in the histogram of grayscale images decreases and local minimum values increase after steganographic embeddings proposed by J. Zhang et al. [22].

Q. Zhang et al. [12] presented a detection method where unpredictable noise variation in image is detected to locate tampering in image.This is done using clustering method combined with unsupervised and supervised clustering. Every image is converted to HSV color space.Here the images are divided into blocks such that there is no overlapping with each image block is of different size and hence noise variance for each block is estimated. The results computed for each image block size proves that block with size 32X32 provides best detection results. But image blocks with size 16x16 and 64X64 pixels gives poor performance for this method.

V. Anand et al. [2] inorder to detect copy-move forgery suggested to combine dyadic wavelet transform (DyWT) method and scale invariant feature transform (SIFT). Firstly, an image is divided into four different bands that are HH, HL, LL, LH on which DyWT is applied. To get key features, SIFT is tried on LL parts as it proves to be maximum informative region when compared with other sub-bands. Thereafter having extracted the key features,various descriptor vectors are formed and similarities among various descriptor vectors are captured which easily locates the tampered regions.

G. Muhammad et al. [17], presented undecimated dyadic wavelets approach to detect copy move forgery in images. It proved more efficient for data analysis. The image is partitioned into two bands that are LL1 and HH1. The similarity between the image blocks is captured in form of a feature to detect tampering. In this method, the LL1 subband shows more similarity amongst the regions that are copied from one region and moved to other regions. But in HH1 band similarity captured is low due to noise unpredictability in the block that has been moved.

P. Deshpande et al. [7] mentioned techniques for tampered region detection in image. The first technique suggests to divide the image into sub-images and apply DWT transformation to each sub-image. No noise is detected, no filtering is considered and the moved region is located by pixel to pixel comparisons. This method proves best for the images containing regions those are pasted without any modifications to other location in that same image. In the second technique, feature vectors are designed for each subdivided image blocks and compared amongst each other. This method takes into consideration both rotation and noise removal issues and hence achieves best tamper detection rates.

Y. Cao et al.[4] discussed approach based on Circular Block along with DCT to detect tampered region in images with uniform background, images with high resolution, irregular duplicate regions and multiple copy-move regions. This method performs poor for low quality images.

V. Christlein et al. [5] introduced Same Affine Transformation Selection (SATS) method. The affine based transformation is applied to the copied portions to detect regions with tampering. But when copied regions are rotated not considering the image size then SATS method does not give reliably detection rates.

S.J. Ryu et al. [19] introduced copy, rotate and move (CRM) that works on Zernike moments concept. It helps to reduce of JPEG compression, any kind of blurring and effects due to additive white Gaussian noise. Here forgery is detected even on the rotated region. This is due to the fact that Zernike moments do not vary due to the rotation in regions. But the method is not able to detect tampering based on affine transform.

The techniques mentioned above are useful for detecting various types of image forgeries. Thus extensive survey is done to detect tampering in images hence providing future enhancements.

CHAPTER 2
Steganalysis in Images

The Steganalysis that is detecting tampering in images can be solved by designing various features to get reliable probabilistic detection rates. Features extracted from Corel Image Database are used to train SVM model.

2.1 Analysis for LSB Matching

Consider a grayscale image with pixel intensity values in the range 0,...,N-1 where N is 256. In our analysis, we have considered all cover and stego images to be gray scale images with pixel intensity values in the range 0..255. Here $p_{cover}(i,j)$ denotes intensity of cover image at location (i,j). We consider embedding of one secret bit per cover pixel using LSB matching steganography to get stego image $p_{stego}(i,j)$.

The LSB matching steganographic algorithm follows random increment or decrement of pixel intensity value hence removes the existence of asymmetry of odd or even pixels. It proves difficult to backtrack forged image to obtain hidden information.

$$p_{stego} = \begin{cases} p_{cover}+1, & \text{if } bit \neq \text{LSB}(p_{cover}) \text{ and } (k' > 0 \text{ or } p_{cover} = 0) \\ p_{cover}-1, & \text{if } bit \neq \text{LSB}(p_{cover}) \text{ and } (k' < 0 \text{ or } p_{cover} = 255) \\ p_{cover}, & \text{if } bit = \text{LSB}(p_{cover}) \end{cases} \tag{2.1}$$

where

p_{cover} and p_{stego} denotes pixel intensity value of cover and stego image at location (i,j) respectively.

bit is the secret bit to be hidden in image

k' is an i.i.d. random variable having uniform distribution on $\{-1,+1\}$.

Algorithm 1 LSB Matching Algorithm

Inputs: Query Image **image** and Hidding Bit matrix **bits**
for $i \leftarrow 1tosize(image,1)$ **do**
  for $i \leftarrow 1tosize(image,2)$ **do**
    if $mod(image(i,j),2) \neq bit(i,j)$ **then**
      $RandNum \leftarrow rand(1)$
      $Output(i,j) \leftarrow image(i,j) + 1 * (RandNum \geq 0.5) - 1 * (RandNum < 0.5);$
    else
      $Output(i,j) \leftarrow image(i,j)$
    end if
  end for
end for

2.1.1 Effects of Steganographic Embedding on Image Histogram

The intensity histogram for cover image is interpreted as:

$$h_{cover}(n) = |\{(i,j)|p_{cover}(i,j) = n\}| \tag{2.2}$$

where n is gray scale pixel intensity value in range 0..255 and $h_cover(n)$ denotes frequency of occurrence of pixel value n in cover image.

Now we analyze effect of using LSB matching steganography with an embedding rate $\rho = 1$ on the image histogram. The embedding rate of one means we are embedding one secret bit per pixel which means if bit is modified then pixel value will either be incremented or else decremented by one. It is observed that for 50 % of cases pixel values won't change and out of remaining 50 % , 25 % chances each for a pixel value to be incremented or decremented by one.

$$h_{stego}(n) = (1 - \frac{\rho}{2})h_{cover}(n) + \frac{\rho}{4}h_{cover}(n-1) + \frac{\rho}{4}h_{cover}(n+1) \tag{2.3}$$

LSB matching is interpreted as low pass filtering done on intensity histogram[22]. The obtained stego histogram is less steep than that of cover image histogram which is due to filtering of image histogram that attenuates energy in high frequency regions and also reduces sharpness in amplitudes of local extrema points in stego image histogram.

A local extrema(maxima or minima) point, n*, in the histogram function is defined as:

$$(h_{cover}(n^*) - h_{cover}(n^* - 1))(h_{cover}(n^*) - h_{cover}(n^* + 1)) > 0 \tag{2.4}$$

From equation 2.3,for every local maximum, n* :

$$\begin{aligned} h_{stego}(n^*) &= (1-\frac{\rho}{2})h_{cover}(n^*)+\frac{\rho}{4}h_{cover}(n^*-1)+\frac{\rho}{4}h_{cover}(n^*+1) \\ &= h_{cover}(n^*)-\frac{\rho}{4}[(h_{cover}(n^*)-h_{cover}(n^*-1))+(h_{cover}(n^*)-h_{cover}(n^*+1))] \\ &< h_{cover}(n^*) \end{aligned} \tag{2.5}$$

Hence, for each local minima point we get $h_{stego}(n^*) > h_{cover}(n^*)$. Hence we conclude that after LSB matching embeddings, the local maximum value at any pixel intensity of histogram decreases and the local minimum value at any pixel intensity increase. This property provides a reliable discriminating feature that is used to locate tampering in the image.

- Following are the feature vectors used:

$$D_{cover} = \sum_{n^*} \mid 2h_{cover}(n^*) - h_{cover}(n^*-1) - h_{cover}(n^*+1) \mid$$
$$D_{stego} = \sum_{n^*} \mid 2h_{stego}(n^*) - h_{stego}(n^*-1) - h_{stego}(n^*+1) \mid$$

  These values represent the summation of absolute difference amongst each local extremum (local minima or local maxima) and its neighbors in image histogram.

  Result

  $D_{cover} > D_{stego}$ for any considered image after embedding secret message using LSB matching. This proves the fact that the local maximum value at any pixel intensity of an stego image histogram decreases and the local minimum value at any considered pixel intensity increases when compared with corresponding considered pixel intensity value in cover image histogram.

Algorithm 2 Find_D Algorithm

1: Inputs: Query Image **image**
2: Initialize D
3: Find Image histogram
4: $MaxIdx \leftarrow Findpeaks(imagehist)$
5: $MinIdx \leftarrow Findpeaks(-imagehist)$
6: $extremaIdx \leftarrow MergeMaxIdxandMinIdx$
7: **for** $i \leftarrow 1tolength(extremaIdx)$ **do**
8: $D \leftarrow D + abs(2 * imageHist(extremaIdx(i))) + abs(2 * imageHist(extremaIdx(i)+1)) + abs(2 * imageHist(extremaIdx(i)-1))$
9: **end for**

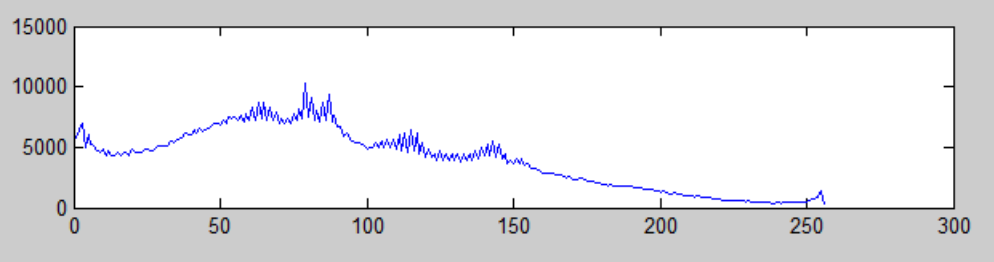


Figure 2.1: Cover Image Histogram

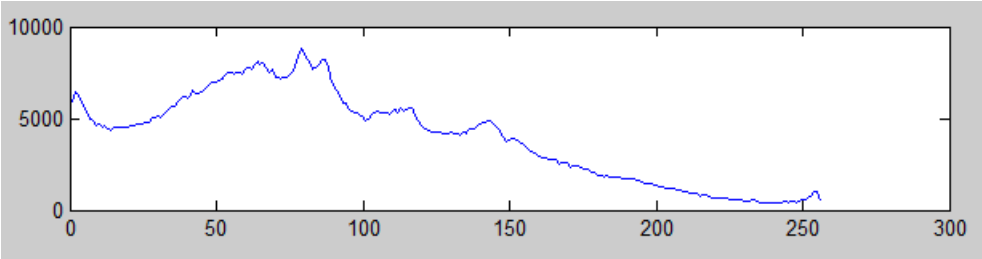


Figure 2.2: Stego Image Histogram

Experiments done:

The presented histogram plots in figure 2.1 and 2.2 are generated in Matlab. LSB matching embedding is done in cover image to generate corresponding stego image. It clearly shows that stego histogram is less steeper than cover histogram of an image and this is due to low pass filtering that occurred after LSB matchings embedding. In the plot at any considered point local maxima points in the stego image histogram are decreasing when compared with histogram of cover image. Similarly local minima in the histogram of stego image are increasing when compared with histogram of cover image. This idea defines our features used to train a classification model.

2.2 Histogram Characteristics function(HCF)

Harmsen's detector used to detect steganography in images uses histogram that captures how frequently a particular pixel intensity value occurs in any image [15].

$$h_{cover}(n) = |\{(i,j)|p_{cover}(i,j) = n\}| \tag{2.6}$$

Steganographic embedding is interpreted as additive noise $(modN)$ and $f_{\triangle}$ denotes mass function of noise as a random variable that takes value n with particular probability of,

$$\left| p_{cover}(i,j) - p_{stego}(i,j) = n(modN) \right| \tag{2.7}$$

The above equation shows that difference in intensity pixel values of cover image and stego image at particular location is noise.This is due to LSB matching embedding is modeled as additive noise. When two integer random variables are added it results in convolution of their mass functions. Let $H_{cover}[k], H_{stego}[k], F_{\triangle}$[k] be discrete Fourier transform(DFTs) of mass functions $h_{cover}, h_{stego}, f_{\triangle}$ respectively.

$$H_{stego}[k] = H_{cover}[k]F_{\triangle}[k] \tag{2.8}$$

Thus $H_{stego}[k]$ is the HCF of the stego image. The distribution of noise added due to LSB matchings is given as $f\triangle(0) = 0.5$, $f\triangle(1) = 0.25$ and $f\triangle(-1) = 0.25$ gives $F_{\triangle}(k) = cos^2(\frac{\pi k}{n})$ which is always < 1 being a monotone function.Therefore,

$$H_{stego}(k) < H_{cover}(k) \quad \forall k \tag{2.9}$$

Center of mass(COM) of histogram characteristics function is calculated as

$$COM(H[k]) = \frac{\sum_{i=0}^{n} iH[i]}{\sum_{i=0}^{n} H[i]} \tag{2.10}$$

where $n = \frac{N}{2}$ due to symmetry of DFTs of real signals and COM(H[k]) represents center of mass of Histogram characteristic function (HCF).

After steganographic embedding center of mass varies as

$$COM(H_s[k]) < COM(H_c[k]) \tag{2.11}$$

It is this property of COM that is used for detecting Steganography in images.

2.3 Calibration by Down-sampled Image

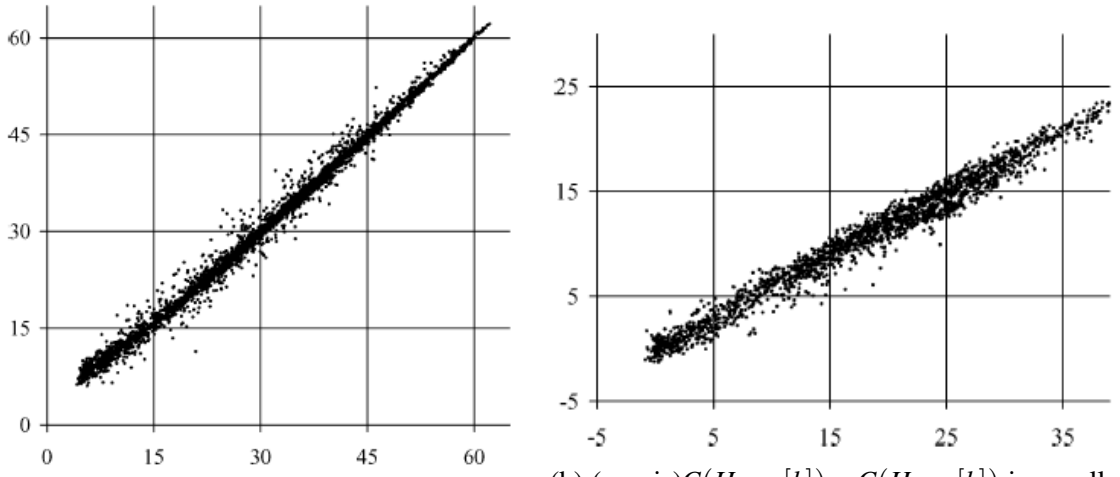
Here image is down-sampled by a factor of two in both scales using averaging filter. Let $p'_{cover}(i,j)$ and $p'_{stego}(i,j)$ be pixel intensities after down-sampling cover image and stego image respectively.Here $H'_{cover}[k]$ and $H'_{stego}[k]$ denotes HCF of down-sampled cover image and stego image respectively.

Observations:

1. It is observed that down-sampling operation do not affect the COM of the HCF of cover images as shown in plot in figure 2.3. We conclude that for images without

hidden data:

$$C(H^{'}_{cover}[k]) \approx C(H_{cover}[k]) \tag{2.12}$$



(a) (x-axis) $C(H_{cover}[k])$ is close to (y-axis) $C(H[k])$.

(b) (x-axis)$C(H_{cover}[k]) - C(H_{stego}[k])$ is usually greater than (y-axis) $C(H^{'}_{cover}[k]) - C(H^{'}_{stego}[k])$.

Figure 2.3: Scatterplot from 2000 JPEG images[15]

2. LSB matching process introduces the noise in the down-sampled cover image which results in reducing the COM of HCF but it is observed that it does so in lesser extent i.e.,

$$C(H_{cover}[k]) - C(H_{stego}[k]) > C(H^{'}_{cover}[k]) - C(H^{'}_{stego}[k]) \tag{2.13}$$

Thus this adding and rounding operations in down-sampling process tries to even out added noise due to LSB matchings. Thus combining equation 2.12 and 2.13 we get a new detector,

$$C(H[k]) < C(H^{'}[k]) \tag{2.14}$$

Hence downsampling idea provides us with the new feature i.e., $C(H[k])/C(H^{'}[k])$ for detecting presence of LSB matching steganography.

The features discussed above are used to train SVM model for solving the detection problem. Below mentioned algorithm is designed for finding relevant features of an image inorder to detect whether it is tampered image or not.

Features used for Classification:

$$C(H[k]) = \frac{\sum_{i=0}^{n} iH[i]}{\sum_{i=0}^{n} H[i]}$$

$$D = \sum_{n^*} \mid 2h(n^*) - h(n^* - 1) - h(n^* + 1) \mid$$

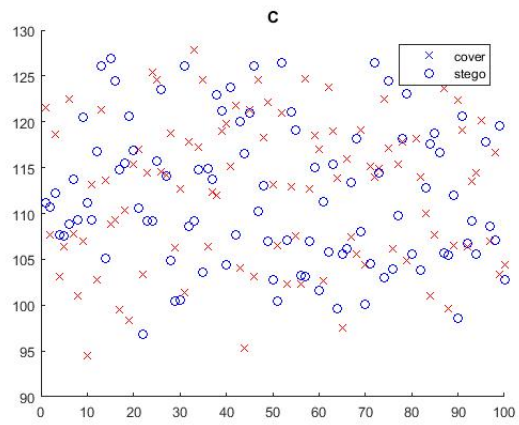
$$Dratio = D/D\_downsampled$$

$$Cratio = C/C\_downsampled$$

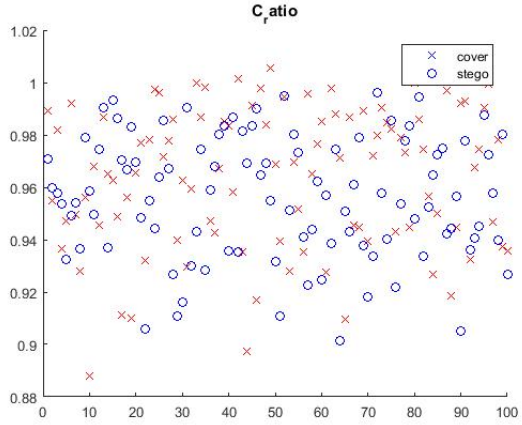
Algorithm 3 Find Stego features

Inputs: Given Image
Initialize $D = 0, D\_downsampled = 0$
Find Image histogram h
Find peaks(*Maximas and Minimas*)*from Image histogram*
$n \leftarrow$ [MaxIdx, MinIdx]
for $i \leftarrow 1$ *to length*(n) **do**
    $D \leftarrow D + abs(2 * h(n)) - abs(2 * h(n+1)) - abs(2 * h(n-1))$
end for
Downsample the Image and Compute Histogram h^* and peaks (n^*)
for $i \leftarrow 1$ *to length*$(n*)$ **do**
    $D\_downsampled \leftarrow D\_downsampled + abs(2 * h^*(n^*) - abs(2 * h^*(n^* + 1)) - abs(2 * h^*(n^* - 1))$
end for
$Ratio1 \leftarrow D/D\_downsampled$
$Ratio2 \leftarrow C/Ctilda$
$Features \leftarrow [D, D\_downsampled, C, Ctilda, Cratio, Dratio]$

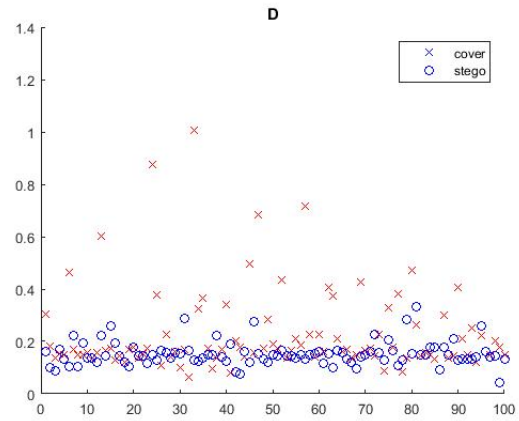
Comparisions and Conclusion The plot obtained in fig 2.4(a)and (b) is for COM classifier where all crosses and circles are mixed and hence no clear separation is observed. Similarly, Cratio feature also gives no clear distinction between stego and cover images. The plot obtained from Matlab in fig 2.4 (d) shows that values for Dratio (before embedding-cross) is higher than Dratio values (after embedding-circles). So here all circles(stego) are seen lower as compared to crosses(cover). This distribution helps us to locate a boundary separating all cover images and stego images. Thus the feature Dratio gives reliable performance for classification of tampered images.



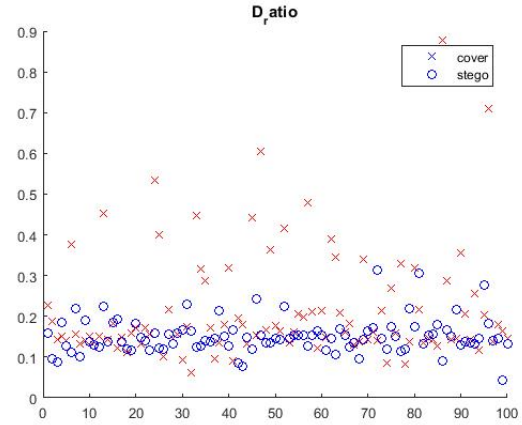
(a) Values of COM(H[k]) (crosses) before and (circles) after embedding for 100 images



(b) Values of Cratio (crosses) before and (circles) after embedding for 100 images



(c) Values of D (crosses) before and (circles) after embedding for 100 images



(d) Values of Dratio (crosses) before and (circles) after embedding for 100 images

Figure 2.4: Plotting Classification features for 100 images

2.4 Classifiers

2.4.1 Support Vector Machine (SVM)

The goal of SVM is to design a hyperplane that classifies all training vectors in two classes.The best choice is hyperplane that leaves the maximum margin from both classes. The support vector machine searches for the closest point, which it calls the support vectors.The name support vector machine is because points are like vectors and that the best line depends on or is supported by the closest points.Once the closest points are located, the SVM draws a line connecting them. It draws this connecting line by doing vector subtraction. The support vector machine then declares the best separating line to be the line that bisects and is perpendicular to the connecting line. We have used radial basis function (RBF) kernel in our experiments done.The RBF kernel on two samples x and y,

represented as feature vectors in some input space, is defined as,

$$K(x,y) = e^{-\|x-y\|^2/2\Sigma^2} \tag{2.15}$$

where $\|x-y\|^2$ is squared Euclidean distance between the two feature vectors and $\sum = 7$

2.4.2 Random Forest

Develops lots of decision tree based on random selection of data and random selection of variable.It Provides class of dependent variable based on many decision tress.The number of trees used for training in our case is 30.

In general, the more trees in the forest the more robust the forest looks like. In the same way in the random forest classifier, the higher the number of trees in the forest gives the high accuracy results.

The advantage of using random forest algorithm is that this can be used for both classification and the regression task.This classifier will handle the missing values.When we have more trees in the forest, random forest classifier wonâĂŹt over-fit the model.Even we can model the random forest classifier for categorical values also.

2.5 Experimental Results

Downloading 10,000 images from Coral Image Database,we divide them into two classes class 1 has 10,000 images and class 2 has 10,000 images. Class 1 has all images as cover image which are left unchanged but class 2 images converted to stego image by embedding bits using LSB matching algorithm. We choose only 8000 images out of 10,000 corel images to train SVM model.Rest of 2000 images are kept as test images. Now any random image is taken (from 2000 Test images or any other random image) and provide it as input to classifier. It classifies and detects whether the image is stego or cover with accuracy 75 %

The higher accuracy guarantees a best detector which differentiates between stego and cover image. The ROC curve in figure 2.5 shows highest probability of detection when all features(C and D with ratio features) are used to train a model. The ratio (Cratio and Dratio) features try to capture variations in values of C(COM feature) and D(Difference vector) in original image to that of down-sampled image. The bar plots in figure 2.6 compare the performance of varied featured used to train SVM model.

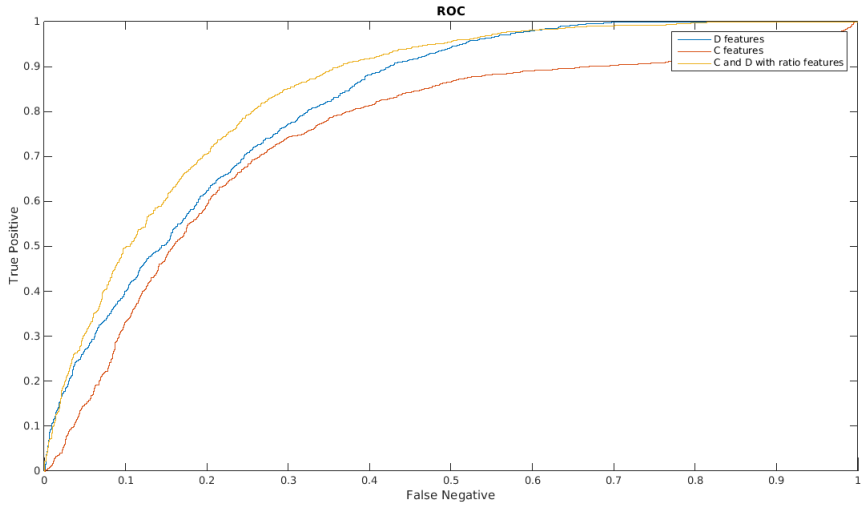


Figure 2.5: ROC plot generated from 10,000 images for classification features COM,D and ratio

Features	SVM	Random Forest
D, D_downsampled	74.16	69.51
C, C_downsampled	71.95	72.01
C,C_downsampled,D,D_downsampled,Cratio,Dratio	77.82	79.10

Table 2.1: Results of Tampered Image detection rates with COM,D vector and ratio features

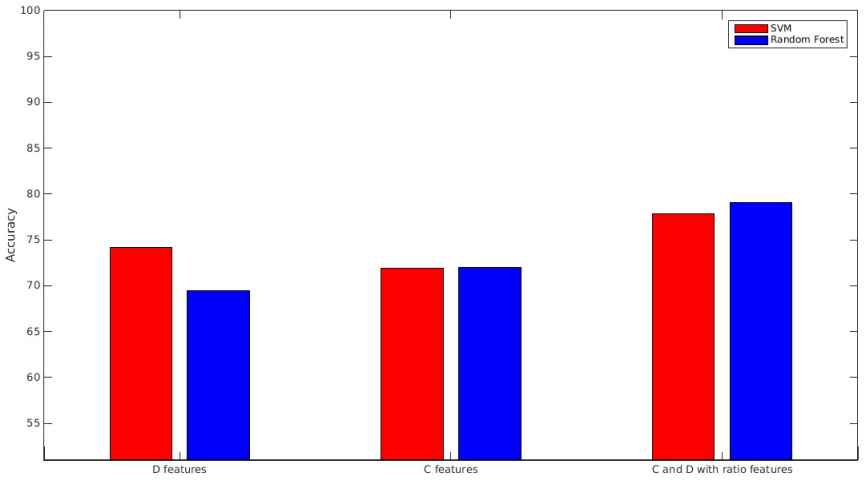


Figure 2.6: Bar graph plot comparing classification features

CHAPTER 3
Steganalysis based on STFT and Mutual Information

3.1 Short Time Fourier Transform (STFT)

Fourier Transform(FT) reveals which frequency components are present in function. However,Fourier transform cannot provide simultaneous time and frequency localization. The Short Time Fourier Transform (STFT) [21] computes Fourier Transform of Signal in sliding window fashion and hence analyses signal better in time and frequency domain simultaneously. It first divides the signal into small time intervals and thereafter calculates the Fourier transform of each interval thereby providing simultaneous time and frequency localization. An appropriate window function $W(t)$ is chosen for dividing the signal into various segments. Examples of $W(t)$ are rectangular window, hamming window, Blackman window and so on.

STFT of function $x(t)$ can be calculated using window W(t) as

$$STFT_x(\hat{t}, f) = \int_t [f(t).W(t-\hat{t})].e^{-j2\pi ft}dt \tag{3.1}$$

where $\hat{t}$ and f is time parameter and frequency parameter respectively.

Selection of appropriate Window and its support plays a critical role in analyzing signal in time and frequency domain simultaneously. A highly localized window in time domain will have worst localization in frequency domain and vice versa as per Fourier principles. Hence there's a trade off between both frequency and time localization.

- **When window W(t)is infinitely long**, i.e $W(t) = 1$ for all t, then STFT is nothing but FT which provides good frequency localization, but worst time localization.
- **When window W(t) is infinitely short:** $W(t) = \delta$, it provides good localization in time domain but worst localization in frequency domain.

The STFT of histogram of stego image and cover image is estimated patch wise. Due to

narrower window size STFT provides excellent time localization which makes it easy to get more precise information as compared to computing DFT of histogram of stego image and cover image. Thus variations that occur due to image tampering can be more precisely captured using STFT concept.

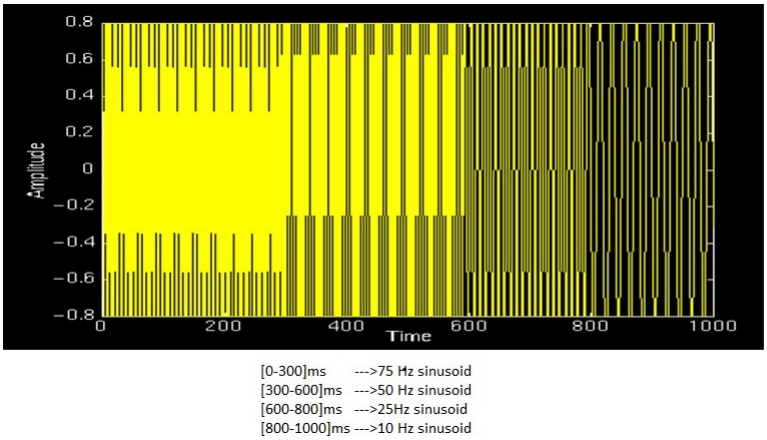


Figure 3.1: Signals with different frequency at different time [3]

The ROC plot in fig 3.3 including STFT of histogram of stego and cover image improves our detection rates because STFT works on small patches of histogram of images and captures variations easily. Hence we get improved ROC after using STFT of image histogram. Even for very low false positive value in ROC curve we get high true positive value (probability of detection) as compared to other features.

Conclusions and comparisons: In previous analysis we computed DFT of histogram of image and down-sampled image that provided excellent frequency localization and detecting accuracy of 75 %. Now we compute STFT of histogram of image and down-sampled image over small patches (narrow window size) of histogram of image and this

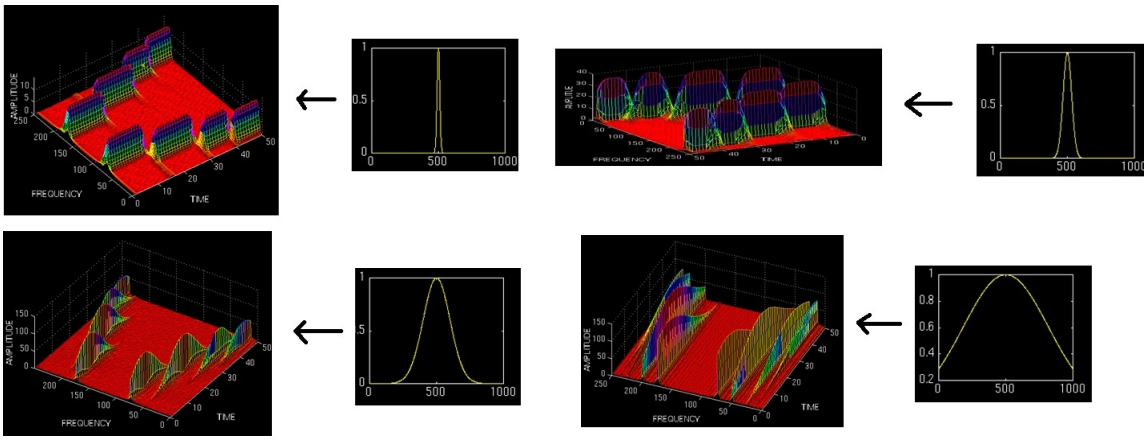


Figure 3.2: STFT computation[3]

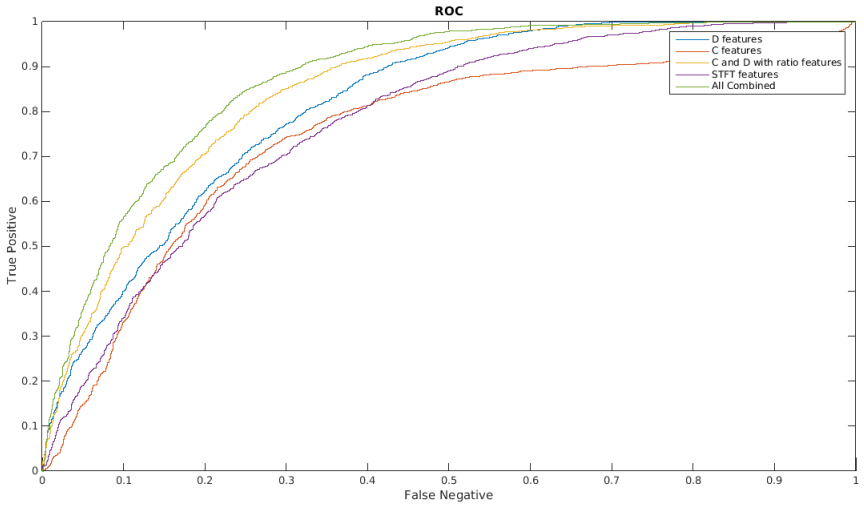


Figure 3.3: ROC plot of all features including STFT feature

Features	SVM	Random Forest
D, D_downsampled	74.16	69.51
C, C_downsampled	71.95	72.01
C,C_downsampled,D,D_downsampled,Cratio,Dratio	77.82	79.10
STFT	70.21	70.89
C,D,ratio and STFT	79.63	79.86

Table 3.1: Results of Tampered Image detection rates with ratio and STFT features

provides excellent time localization. The STFT of histogram of image is taken over patches which make it easy to capture localized variations caused due to tampering in images. Adding STFT feature to previously designed features trains our model to achieve detection accuracy of 80% as shown in fig 3.4.

3.2 Entropy and Mutual Information

Entropy is an information theory term which to measure uncertainty about events which are probabilistic in nature. Entropy is also used to measure predictability of a random event or random variable. Higher the entropy, higher the randomness. Entropy can also be said as expected number of bits to send a message over the communication channel. The entropy of an event which is sure to happen is zero.

Entropy H_X of a discrete random variable X with probability density function(pdf) P_X is,

$$H_X = -\sum_{k=1}^{N} P(x_k) \log_b P(x_k) \tag{3.2}$$

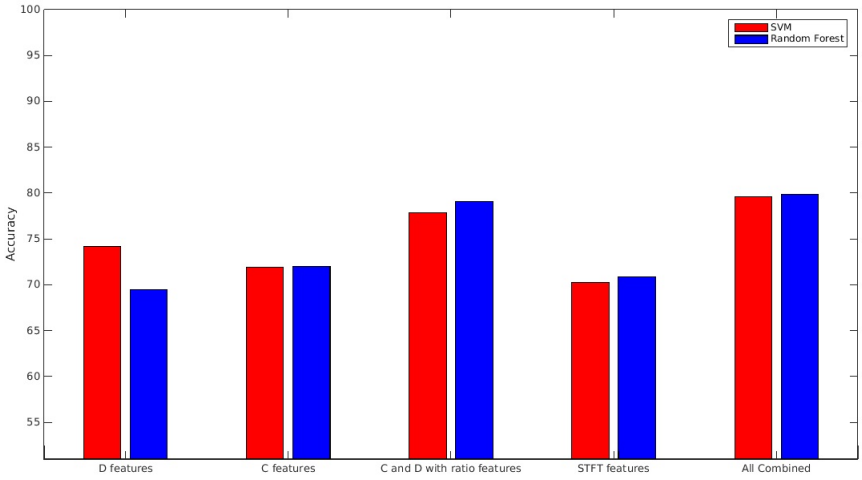


Figure 3.4: Bar graph plot comparing classification features with STFT feature

where b is the base of the logarithm used. Common values of b are 2, Euler's number e, and 10, and the unit of entropy is Shannon for b = 2, Nat for b = e, and Hartley for b = 10.When b = 2, the units of entropy are also commonly referred to as bits.

For two random variables X and Y, their joint uncertainty is given by joint entropy as,

$$H_{XY} = -\sum_{x}\sum_{y} p(x,y)\log p(x,y) \tag{3.3}$$

To measure how much we can predict a random variable X given another random variable Y, Conditional entropy $H_{X|Y}$ is used.

$$H_{X|Y} = -\sum_{x}\sum_{y} p(x,y)\log p(x|y) \tag{3.4}$$

Mutual Information: Mutual information(MI)[18] is used to measure how much two random variables are statistically dependent. Suppose message X is sent over the noisy communication channel C, and message Y is received by the receiver. Then $I(X;Y)$ i.e mutual information between X and Y is known as capacity of channel. I(X;Y) tells with how much certainty, we can decode the noisy message Y to get original message X.

Suppose random variables X and Y are jointly distributed as $p(x,y)$ and marginally distributed as $p(x)$ and $p(y)$ respectively, then Mutual information $I(X;Y)$ can be calculated as

$$
\begin{aligned}
I(X;Y) &= -\sum_x \sum_y p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \\
&= H_X - H_{X|Y} \\
&= H_Y - H_{Y|X} \\
&= H_X + H_Y - H_{XY}
\end{aligned} \tag{3.5}
$$

Mutual information is used to measure the information that X and Y combinedly share with each other such that knowing one of these variable improves the predictability of the other random variable. For Example, if X and Y are statistically independent random variables then knowing X won't provide any information about Y and hence mutual information $I(X;Y)$ is zero. If Y is deterministic function of random variable X i.e $Y = g(X)$. then knowing X, we can predict random variable Y with zero error. Hence here H(Y|X) will be zero and I(X;Y) will be maximum. ' Also it can be proved that mutual information is symmetric i.e commutative quantity between two random variables.

$$
I(X;Y) = I(Y;X) \tag{3.6}
$$

Venn Diagram: Joint entropy and mutual information can be pictorially understood by Venn diagram shown in figure .

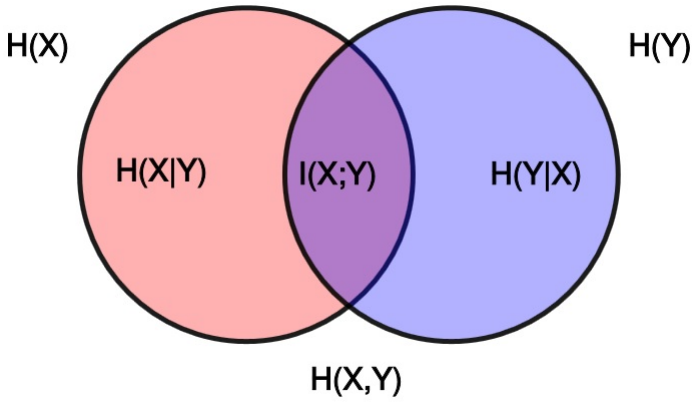


Figure 3.5: Mutual Information Venn Diagram[1]

Application of Mutual Information is not limited to just communication field. It has been used successfully in following different applications where statistical dependence between two random variables needs to be quantized.

- To determine Similarity between different clusters of a data.

- Used to measure similarity between phrases in search engines.
- To measure channel capacity of communication channel.
- Used as similarity measure to register two images in image registration problem.

3.3 Feature Extraction based on Mutual Information

Mutual information is quantization of information of one random variable through the other. In any natural image there exists dependency between neighboring pixels to some extent. This dependency restricts further to classify clearly amongst cover and stego images. In cover image measure of mutual information is high as compared to stego images because dependency between adjacent pixels is high. Whereas, in stego image mutual information is low due to addition of noise in stego images.

The difference between adjacent neighboring pixels is considered based on which mutual information is computed. For many operations, distinguishing objects depends on whether pixels are connected or not and way they are connected to each other. The two popular neighborhood pattern used by image processing community are 4 and 8 Connected Neighborhood.

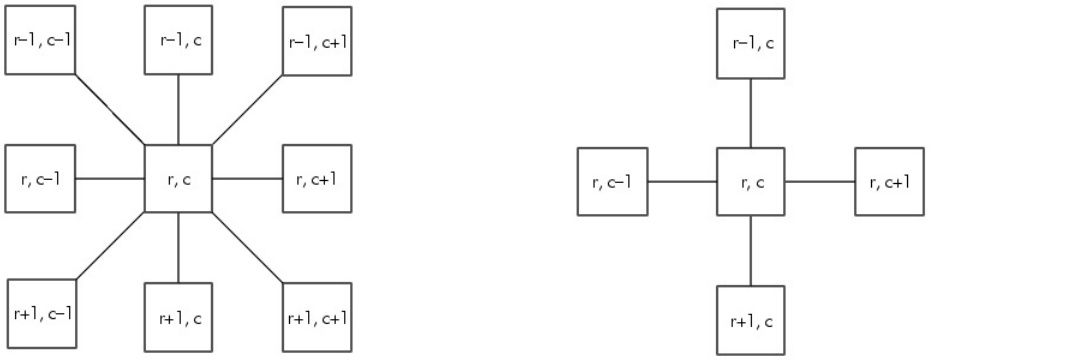
3.3.1 4- and 8- Connected Neighbors

In 4-connected neighborhood, only the pixels surrounding the central pixel in four directions i.e top, left, left right are considered as a part of neighbors. While in 8-connected neighborhood, all the pixels that touch the central pixel in 8 directions which includes diagonal neighbors in addition to 4-connected neighbors are considered.

8-connected neighborhood is more practical since it includes all neighboring pixels and is more generally used in practical applications than 4-connected neighborhood.

Hence for computing features, we will use 8-connected neighborhood.

Comparisons and conclusions: The pixels in every cover image(original image) is highly dependent on each of its neighboring pixel values. So adjacent pixels values in any cover image matrix is highly correlated. Due to this high dependency among adjacent pixels the measure of mutual information is high. The stego image is cover image with some steganographic embedding (hidden data) inside it. Becuase data embedded in the stego image behaves as independent additive noise ,the adjacent pixels values in stego image matrix are highly uncorrelated. Hence, dependency of adjacent pixels reduces in



(a) An 8- Connected Neighborhood (b) An 4- Connected Neighborhood

Figure 3.6: 4- and 8- Connected Neighbors

stego image as compared to cover image. Due to reduced dependency, measure of mutual information is low in stego image. Hence this variations are captured in form of features which acts as a discriminator for detecting steganography in images.

The ROC plot in fig 3.7 including mutual information of image as designed feature improves our steganographic detection rates because mutual information quantizes information of one random variable through the other. Due to this variation among pixels are captured by features designed based on mutual information concept. Hence we get improved ROC such that for very low false positive value in ROC curve we get high true positive value (probability of detection).

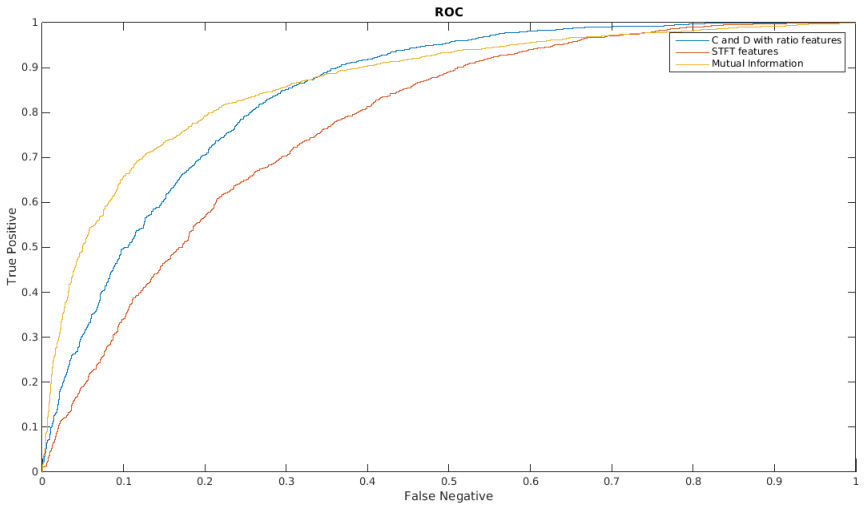


Figure 3.7: ROC plot of all features including MI(Mutual Information)

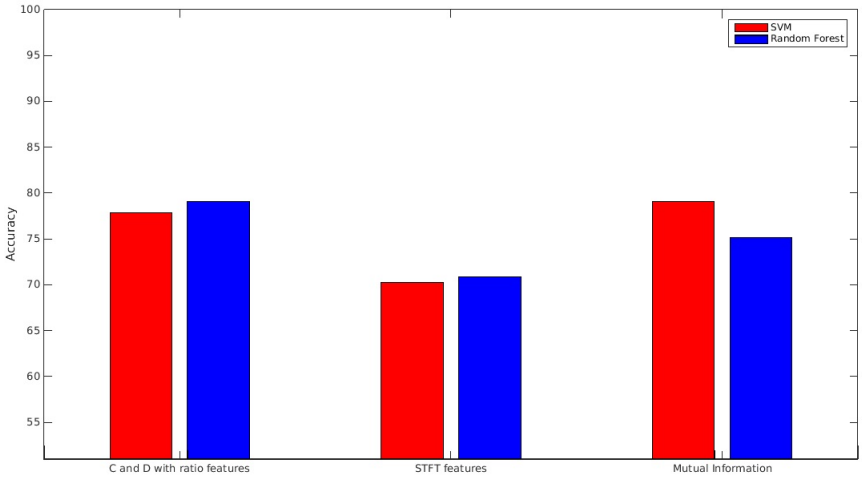


Figure 3.8: Bar graph plot comparing classification features with MI features

Features	SVM	Random Forest
C,C_downsampled,D,D_downsampled,Cratio,Dratio	77.82	79.10
STFT	70.21	70.89
Mutual Information	79.11	75.12

Table 3.2: Results of Tampered Image detection rates with STFT and MI features

3.4 Features Designed

In previous work by A. D. Ker, Center of Mass of DFT of Histogram was used to capture low pass filtering caused by LSB Matching. When window W(t)is infinitely long, i.e W(t) = 1 for all t, then STFT is nothing but FT which provides good frequency localization, but worst time localization. When window W(t) is infinitely short, it provides good localization in time domain but worst localization in frequency domain. So DFT gives global information in intensity domain and lacks local intensity changes in histogram. To improve this, we have used Short time Fourier Transform (STFT) to capture local changes in the histogram.

Here STFT is computed by dividing histogram into small segments of specified window size and thereafter calculating DFT over each window size.
Number of windows used is 8
Window length is $256/8 = 32$

Now further we compute COM of STFT of Histogram over each window and hence we get eight COM's for 8 different windows. Also, we have considered COM of STFT of down-sampled image histogram as down sampling averages out noise in an image. Hence variations in an image are captured using the ratio $C/C_downsampled$.

Hence eight features are used based on COM of STFT of image histogram and other eight features based on ratio. In total 16 features are used to train SVM model based on STFT.

Similar to this, We have used mutual information to capture statistical independence caused by LSB matching algorithm. Here mutual information is computed over difference values of neighboring pixels in 6 different directions. The difference value matrix obtained is considered as gradient matrix.

Gradients using different Kernels : To compute gradients we consider following kernels:

First Order Difference (Horizontal Direction) : $\begin{bmatrix} 1 & -1 \end{bmatrix}$

First Order Difference (Vertical Direction) : $\begin{bmatrix} 1 \\ -1 \end{bmatrix}$

Second Order Difference (Horizontal Direction) : $\begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$

Second Order Difference (Vertical Direction) : $\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$

Diagonal Right : $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}$

Diagonal Left : $\begin{bmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \end{bmatrix}$

Hence six different types of gradient matrices can be obtained for any given image. Further Mutual information of two random variables which in our case are two gradient matrices (one kernel at a time)is computed. Hence six MI features can be obtained for six different kernels applied to an image.

Calculate Mutual Information:

$$\begin{aligned} I(X;Y) &= -\sum_x \sum_y p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \\ &= H_X - H_{X|Y} \\ &= H_Y - H_{Y|X} \\ &= H_X + H_Y - H_{XY} \end{aligned} \tag{3.7}$$

So while computing MI the normalized image histogram (1D histogram) is used as marginal probability density function (PDF). Further, 2D image histogram is computed which is used as joint probability density function(JDF) to calculate MI.

2D Histogram:A 2D histogram shows the relationship of intensities at the exact position between two images. The 2D histogram is mostly used to compare 2 channels in a multi-channel images, where the x-axis represent the intensities of the first channel and the y-axis the intensities of the second channel. It counts the occurrence of combinations of intensities.

Similarly, down-sampled image histogram is considered and six different types of gradients are applied over down-sampled image. Further Mutual information of two random variables which in our case are two gradient matrices (one kernel at a time)is computed. Hence six MI features are obtained for six different kernels applied to the down-sampled image.

As down-sampling cancels out noise in an image, variations can easily be captured between image and down-sampled image by taking ratio of MI of an image to that of MI of down-sampled image. Hence we get six ratio features to train our model.

Total Features used in training: We have computed C (COM of DFT of image histogram), D (sum of absolute difference between local extrema and its neighbors in image histogram), C_downsampled, D_downsampled (for down-sampled image), Cratio and Dratio based on DFT of image histogram as all 6 features.

Further, we computed C_stft (COM of STFT of image histogram) and a ratio (between COM of STFT of image and down-sampled image) based on STFT of image histogram as all 16 features.

Further, based on six gradients we have got six MI features each for image and down-sampled image and ratio as all 18 features. Hence 40 features have been designed and used to train SVM model to detect tampering in images.

CHAPTER 4
Experimental Results and Conclusions

4.1 Comparison on Designed Features :

Detection rates are reduced when LSB matching (+1 or -1 embedding) is used as LSB matching is interpreted to be an additive noise process which results in poor detection of images with high-frequency noise. This happens because noise occurring at higher frequency is wrongly interpreted as a hidden data. To defeat this problem a focused steganalysis algorithm is to be designed.

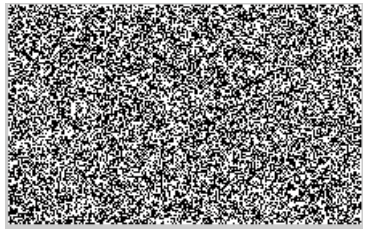
Experiments conducted:

- Consider any cover image (fig 4.1(a)) and bits to be hidden inside the cover image shown as bit image (fig 4.1(b)) in MATLAB.
- Using LSB matching hidden message in form of bits is embedded in the cover image that gives us a new image called stego image.The stego image has hidden message inside it.
- Plot histogram of cover image (fig 4.2) and stego image(fig 4.3) obtained after embedding using LSB matching) obtained in MATLAB.

  Observation:

  - The stego Histogram is less steep than that of cover histogram as shown in fig 4.2 and fig 4.3. Thus LSB matching proves to be interpreted as low pass filter of the image histogram with kernel $[\rho^*/4, 1-\rho^*/2, \rho^*/4]$. This low pass filtering lessens energy in high frequencies and modifies the amplitude values of local extrema (local minima and local maxima in the Histogram).
  - Attenuation of local extrema due to LSB matching in image where Local maxima of an images in gray-scale or color histogram decrease and the local minima increase. Clearly seen that local maximum values at considered pixel in-

(a) Cover Image



(b) bit image

(c) Stego image using bits

Figure 4.1: Data Embedding in Image using LSB Matching

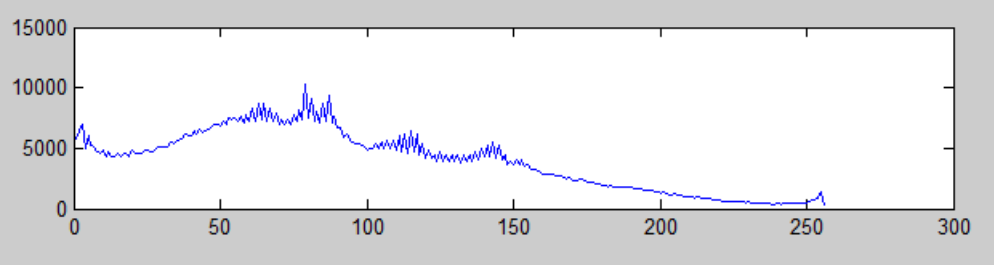


Figure 4.2: Histogram of Cover Image

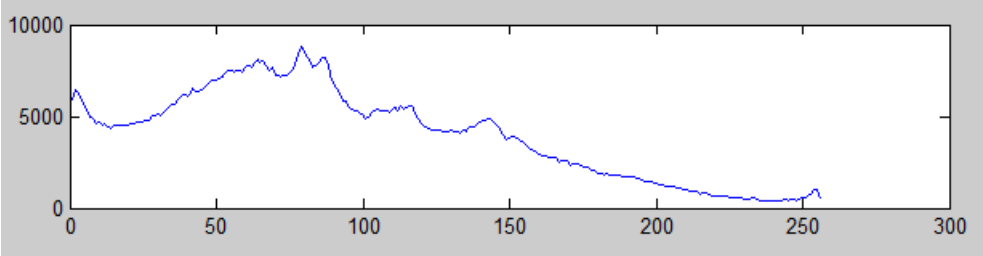


Figure 4.3: Histogram of Stego Image

tensity is lessened and local minimum value is increased in stego histogram when compared with cover histogram.

- The cover image has more variations as compared to stego image. Due to higher variation COM of cover image is higher than stego image. Thus COM acts as an discriminator for steganographic detection.

$$COM(H[k]) = \frac{\sum_{i=0}^{n} iH[i]}{\sum_{i=0}^{n} H[i]} \tag{4.1}$$

After steganographic embedding center of mass varies as

$$C(H_{stego}[k]) < C(H_{cover}[k]) \tag{4.2}$$

- Following are the feature vectors used:

$$D_{cover} = \sum_{n^*} \mid 2h_{cover}(n^*) - h_{cover}(n^* - 1) - h_{cover}(n^* + 1) \mid$$
$$D_{stego} = \sum_{n^*} \mid 2h_{stego}(n^*) - h_{stego}(n^* - 1) - h_{stego}(n^* + 1) \mid$$

These represents the summation of absolute difference amongst each local extremum (local minima or local maxima) and its neighbors in histogram.

Result

$D_{cover} > D_{stego}$ for any considered image after embedding secret message using LSB matching. This proves the fact that the local maximum value at any pixel intensity of an stego image histogram decreases and the local minimum value at any considered pixel intensity increases when compared with corresponding considered pixel intensity value in cover image histogram.

- The down sampling of an image using averaging concept cancels noise in image. The variations are captured through designed D feature for histogram of image and histogram of down-sampled image in form of Dratio (D/D_downsampled) feature. Similarly variations while computing COM of histogram of image and COM of histogram of down-sampled image are captured through Cratio (C/C_downsampled) feature.

- Downloading 10,000 images from Coral Image Database.We divide them into two classes each having 10,000 images. Class 1 has all images as cover image which are left unchanged but class 2 images converted to stego image by embedding bits using LSB matchings. We choose only 8000 images out of 10,000 corel images to train a model i.e. SVM classifier. Rest of 2000 images are kept as test images. The higher

accuracy guarantees the best detector which differentiates between stego and cover image .

Now any random image is taken (from 2000 Test images or any other random image) and provide it as input to the classifier. SVM model classifies and detects whether the image is stego or cover with accuracy >=85 %.

- In previous analysis we computed DFT of histogram of image and down sampled image that provided excellent frequency localization and detecting accuracy of 75 %. The DFT computation of image histogram is similar to STFT computation with window size infinitely long. Now we compute STFT (Short time Fourier transform) of the histogram (1-dimensional signal) of image and down sampled image over small patches (narrow window size) of image histogram and this provides excellent time localization. The STFT of the histogram of image is taken over patches which make it easy to capture localized variations caused due to tampering in images. Adding STFT feature to previously designed features trains our model to achieve detection accuracy of 80%.

- The pixels in every cover image(original image) is highly dependent on each of its neighboring pixel values. So adjacent pixels values in any cover image matrix are highly correlated. Due to this high dependency among adjacent pixels the measure of mutual information is high. The stego image is a cover image with some steganographic embedding (hidden data) inside it. This steganographic embedding is modeled as independent additive noise. Hence, dependency of adjacent pixels reduces in stego image as compared to cover image. Thus the measure of mutual information is low in stego image due to reduced dependency. Hence these variations are captured in form of features which acts as a discriminator for detecting steganography in images.

4.2 ROC Plots and Comparison of Designed Features

All experiments results are obtained considering two image sets of 10,000 images each from Corel Image Database of 10,000 images. A set of 5000 images are considered as cover images and remaining set of 5000 images are converted to stego images. Each image is embedded with a randomly generated message. A training set of 8000 images and test set of 2000 images is formed. The message embedding rate is $\rho = 1$.

Fig 4.4 demonstrates significant improvements in performance over the existing methods. For example, with a false positive rate of 50% the obtained detection rates using only

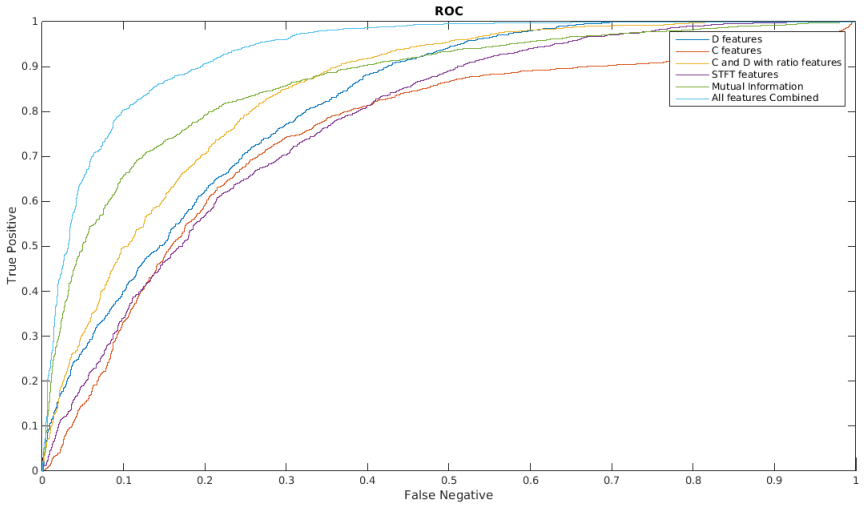


Figure 4.4: ROC curves comparing designed features

COM and D feature are 82% and 90 % respectively. While using COM and D feature with ratio features the detector rates are 93 %. Using all features combined the detection rates are 99 %.

The ROC plot in fig 4.4 including mutual information of image improves our steganographic detection rates because mutual information quantizes information of one random variable through the other. Hence we get improved ROC such that for very low false negative value in ROC curve we get high true positive value (probability of detection) in MI as compared to other features.

4.3 Bar Graph with Accuracy Comparison

The bar graph shown in figure 4.5 shows that with how much accuracy percentage our model is able to detect steganographic embedding for different features used to train the SVM classifier/Random Forest Classifier. The accuracy of 79% is obtained using the Mutual information to detect tampering in an image. Further, combining C and D features with ratio features gets an accuracy of 77% with SVM classifier. Finally, when our model is trained with all combined features the detection accuracy obtained is 85.71%.

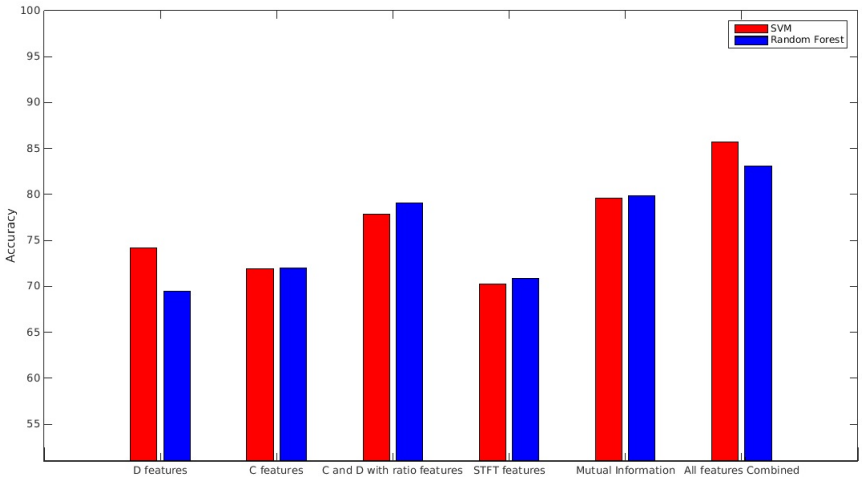


Figure 4.5: Bar graph with accuracy comparision

Features	**SVM**	**Random Forest**
D, D_downsampled	74.16	69.51
C, C_downsampled	71.95	72.01
C,C_downsampled,D,D_downsampled,Cratio,Dratio	77.82	79.10
STFT	70.21	70.89
C,D,ratio and STFT	79.63	79.86
Mutual Information	79.11	75.12
C, D,ratio,STFT and MI	85.71	83.09

Table 4.1: Results of Tampered Image detection rates using MI features

CHAPTER 5
Conclusion

Detection of steganography in cover images that already have high frequency noise is difficult. This is because LSB matching is considered as additive noise process and hence high frequency noise is confused with actual steganographic embedding.

We have implemented steganalysis algorithm by training support vector machine (SVM) using designed features. We used COM of image histogram as a discriminator for steganographic detection. The COM of cover image is more as compared to stego image because variations are more in cover image than stego image. It was noted previously that addition of noise after embedding results in low pass filtering of the histogram of an image. The local maximum value at any considered pixel intensity of an stego image histogram decreases and the local minimum value increases compared with the cover image histogram.

The STFT of histogram of stego image and cover image is estimated patch wise.Due to narrower window size, STFT provides excellent time localization which makes it easy to get more precise information as compared to computing DFT of the histogram of stego image and cover image. Thus variations can be more precisely captured using STFT concept. Further, mutual information concept was introduced because it quantizes information. The measure of mutual information is low in stego image due to reduced dependency. Hence these variations were captured in form of features which act as a discriminator for detecting steganography in images.

Experimental comparisons were performed using bar plots and ROC that clearly differentiated among best features. The experimental results reliably show the improved performance using STFT and mutual information feature. The SVM model developed here is more generalized to work reliably with any kind of dataset. The detectors designed will also be able to detect other types of steganography.

Further new features that quantify data embedding can be designed to improve detection accuracy. Till now, we have used SVM for classification but deep learning methods such as neural networks [16]could be explored further. The steganography is done in images considering embedding rate of $\rho = 1$ per pixel. Hence further experiments could be done by modifying embedding rate [6] lesser than 1.

References

[1] Information diagram. https://en.wikipedia.org/wiki/Information_diagram.

[2] V. Anand, M. F. Hashmi, and A. G. Keskar. A copy move forgery detection to overcome sustained attacks using dyadic wavelet transform and sift methods. In *Asian Conference on Intelligent Information and Database Systems*, pages 530–542. Springer, 2014.

[3] P. Bebis. Short time fourier transform(stft). https://www.google.co.in/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwj1_LaD0OjUAhWDto8KHZDjDnoQFgghMAA&url=https%3A%2F%2Fwww.cse.unr.edu%2F~bebis%2FCS474%2FLectures%2FShortTimeFourierTransform.ppt&usg=AFQjCNGO_wY9crEmjjasDbDTSf8d2bDl7g.

[4] Y. Cao, T. Gao, L. Fan, and Q. Yang. A robust detection algorithm for copy-move forgery in digital images. *Forensic science international*, 214(1):33–43, 2012.

[5] V. Christlein, C. Riess, and E. Angelopoulou. On rotation invariance in copy-move forgery detection. In *Information Forensics and Security (WIFS), 2010 IEEE International Workshop on*, pages 1–6. IEEE, 2010.

[6] J.-F. Couchot, R. Couturier, C. Guyeux, and M. Salomon. Steganalysis via a convolutional neural network using large convolution filters for embedding process with same stego key. *arXiv preprint arXiv:1605.07946*, 2016.

[7] P. Deshpande and P. Kanikar. Pixel based digital image forgery detection techniques. *International Journal of Engineering Research and Applications*, 2(3):539–543, 2012.

[8] J. Fridrich, D. Soukal, and M. Goljan. Maximum likelihood estimation of length of secret message embedded using±k steganography in spatial domain. In *Electronic Imaging 2005*, pages 595–606. International Society for Optics and Photonics, 2005.

[9] M. Goljan, J. Fridrich, and T. Holotyak. New blind steganalysis and its implications. In *Electronic Imaging 2006*, pages 607201–607201. International Society for Optics and Photonics, 2006.

[10] J. J. Harmsen and W. A. Pearlman. Steganalysis of additive-noise modelable information hiding. In *Electronic Imaging 2003*, pages 131–142. International Society for Optics and Photonics, 2003.

[11] T. Holotyak, J. Fridrich, and S. Voloshynovskyy. Blind statistical steganalysis of additive steganography using wavelet higher order statistics. 2005.

[12] Y. Ke, Q. Zhang, W. Min, and S. Zhang. Detecting image forgery based on noise estimation. *International Journal of Multimedia and Ubiquitous Engineering*, 9(1):325–336, 2014.

[13] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy. Improvements to platt's smo algorithm for svm classifier design. *Neural computation*, 13(3):637–649, 2001.

[14] A. D. Ker. Improved detection of lsb steganography in grayscale images. In *International workshop on information hiding*, pages 97–115. Springer, 2004.

[15] A. D. Ker. Steganalysis of lsb matching in grayscale images. *IEEE signal processing letters*, 12(6):441–444, 2005.

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

[17] G. Muhammad, M. Hussain, and G. Bebis. Passive copy move image forgery detection using undecimated dyadic wavelet transform. *Digital Investigation*, 9(1):49–57, 2012.

[18] J. P. Pluim, J. A. Maintz, and M. A. Viergever. Mutual-information-based registration of medical images: a survey. *IEEE transactions on medical imaging*, 22(8):986–1004, 2003.

[19] S.-J. Ryu, M.-J. Lee, and H.-K. Lee. Detection of copy-rotate-move forgery using zernike moments. In *International Workshop on Information Hiding*, pages 51–65. Springer, 2010.

[20] R. Sekhar and A. Chithra. Recent block-based methods of copy-move forgery detection in digital images. *International Journal of Computer Applications*, 89(8), 2014.

[21] M. Sonka, V. Hlavac, and R. Boyle. *Image processing, analysis, and machine vision*. Cengage Learning, 2014.

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