Robust Image Authentication Based on HMM and SVM Classifiers

Mohammad F. Hashmi, Aaditya R. Hambarde, and Avinash G. Keskar, Member, IAENG

Abstract— **In the modern era of digital publishing and imaging, many a time, images are retouched and manipulated to increase and enhance the aesthetic beauty of the images. Several images are morphed before publishing in order to incorporate extra information. The problem is altogether intensified by the presence of various methods of image capture. There exist a variety of cameras with different resolutions and encoding. Due to the availability of image editing tools, tampering with images has become relatively easy. Many times the forged image is compressed and resized before publishing. Hence detecting image forgery in such cases is a challenging task. Different methods of forgery are resizing, blurring, scaling, rotation, addition of noise to the image, addition of some vital data to the image or removal of some data from the image etc. However most methods attempt to detect a particular type of image forgery. In this paper comprehensive technique has been presented for detection of any type of image forgery. Feature extraction techniques like DCT, LBP, Curvelet transform, Gabor filter etc. are used to represent the image in transformed domain. HMM and SVM are the machine learning methods used to classify the image into either of the two classes (Authentic or Forged). CASIA image database was used for training and testing the system.**

Index Terms—**A Image Forgery, Feature Extraction, Classifiers, Hidden Markov Model (HMM), Support Vector Machine (SVM)**

I. Introduction

IMAGE forgery is a major issue today in publishing and printing. Accepting the authenticity of images is difficult since many of the digital images are forged before publishing them. For instance, many a time number of listeners in political rallies are changed in order to show many attendees. Also images presented as evidence in criminal cases may be tampered. Hence images presented as evidences cannot be considered valid any longer. It is becoming more and more important to detect image forgery and to establish the validity of the image. Digital image forensics refers to the science of establishing the validity of a digital image with the help of various methods. Digital image forensics can broadly be divided in to two categories: source identification and forgery detection. Source identification method endeavors to determine the camera that was used to capture the digital image. This method is related with the concept of camera signature etc. Forgery detection on the other hand attempts to determine whether the image has been tampered with and tries to determine the regions in the image where the tampering has been done. There has been substantial work in detecting copy-move forgeries in an image. Image forgery, wherein a part of image is copied and pasted in another part of the same image or a different image is known as copy-move forgery. Copy-move forgery is usually performed to hide some substantial information in the image or to enhance some information. However image forgery is not limited to copy-move forgery. Different types of image forgeries are resizing, blurring, scaling, rotation, addition of noise to the image, addition of some vital data to the image or removal of some data from the image etc. Fig. 1 shows different types of forgeries possible in images. Various forgery detection algorithms are in place for detection of copy-move forgery [12]. Most of them are based on the similarity of copied and pasted parts. In a similar manner there are many methods to detect addition of noise in the image etc. SIFT methods detect a copy-move forgery even when the copied part is scaled and/or rotated and then pasted. However a versatile method which could detect many types of forgeries was not developed until now [11]. The objective of this work is to develop a comprehensive technique for detection of any type of forgery using machine learning algorithms. Machine learning techniques have recently been used to detect image forgery. Artificial Neural Networks (ANN) has been one of the most popular machine learning techniques for detection of image forgery [5]. Such learning techniques however fail to identify tampering in images captured with a source that is different than those they are trained with. The machine learning techniques used in this work are the Hidden Markov Model (HMM) classifier and the Support Vector Machine (SVM) classifier. HMM is a probabilistic model generally used for data classification. HMM has the virtue of imbibing the variability as well as similarity between features of two images. SVM was used for a two class classification in case HMM was not able to classify the image. It was found that HMM when employed along with SVM gave better classification results than when HMM alone was employed. Feature extraction is also a major part of forgery detection. During the course of work, various feature extraction techniques like DCT, Gabor Filter, LBP (Local Binary Patterns) and Curvelet Transform were employed. The

Manuscript received August 31, 2014. This work was supported in part by the Center of Excellence of Department of Electronics Engineering, VNIT Nagpur.

M.F.Hashmi is a doctoral student at Visvesvaraya National Institute of Technology (phone: 91-9665610484 ; e-mail: farooq78699@gmail.com).

A.R.Hambarde is an under-graduate student at Visvesvaraya National Institute of Technology. (e-mail: adihambarde1993@gmail.com).

A.G.Keskar is a Professor at Visvesvaraya National Institute of Technology, Nagpur. He is a senior member of IEEE, IAENG, FIETE, LMISTE, FIE. (e-mail: avinashkeskar@yahoo.com).

images released by the Institute of Automation, Chinese Academy of Sciences (CASIA) and those sourced from the internet were used for testing the proposed method [22].

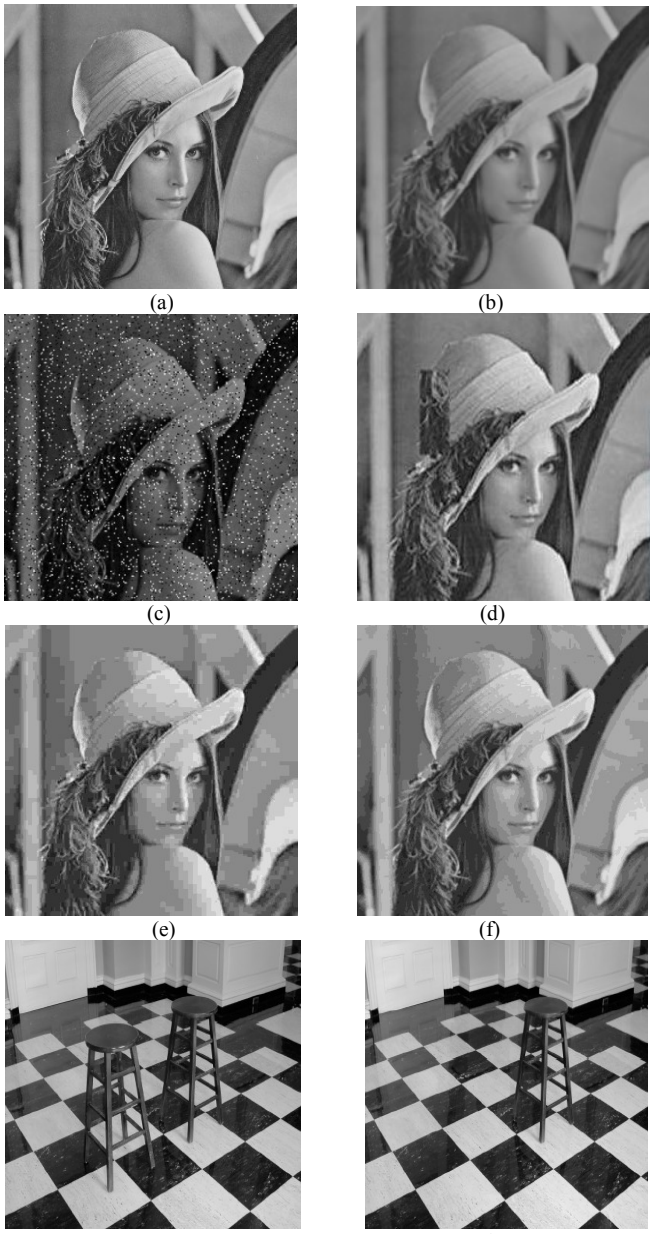

Fig. 1. Different types of attacks (a) Authentic image (b) Gaussian blurring (c) Noise Addition (AWGN) (d) Copy-Move attack (e) JPEG Compression at ratio 32:1 (f) JPEG Compression at ratio 20:1 (g) Authentic Image (h) Image Splicing

II. Related Work

In the past, researchers have tried to develop active methods of image authentication. For example Saxena et al. [1] devised a watermarking scheme using DCT. Also a lot of work has been previously done in blind forgery detection, particularly copy-move forgery detection. Al-Qershi et al. [31] addressed key issues in developing a robust copy-move forgery detector. Popescu et al. [15] devised a method for forgery detection by dividing the image into several blocks, applying the PCA transform (for dimension reduction) and detecting the forgery by detecting similarity between the blocks. Hao-Chiang Hsu et al. proposed copy-move forgery detection by detecting the similarity using feature extraction by Gabor filter [7]. Leida Li et al. [14] devised a method for copy-move forgery detection using Local Binary Patterns. A similar effort based on Harris feature points and LBP was put in by Zhao et al. [3]. M.Qiao et al. [9] used Curvelet statistics for detection of copy-move forgery after dividing the image into several overlapping blocks. S. Khan and A. Kulkarni devised a method for copy- move forgery detection using the multi-resolution characteristic of DWT [26]. In their work, DWT was used for reducing the dimensions of the image and then analysing it. Fridrich et al. [32] analysed the DCT coefficients of image blocks, performed lexicographical sorting and outputted the copied regions by detecting similarity between the blocks. Irene Amerini et al. proposed a SIFT based method for copy-move attack detection [25]. Local visual features like SURF, SIFT, GLOH are robust to several geometrical transformations like rotation, occlusions, clutter and scaling. Hence they are being extensively used for image forgery detection. Mahdian et al. [30] and Qazi et al. [10] present a comprehensive study of blind methods for forgery detection. Birajdar et al. [27] presented a detailed study on passive methods for forgery detection. Use of machine learning techniques for image forgery detection is relatively new. E.S.Gopi et al. [5] used Auto Regressive coefficients as feature vectors and ANN for training the system.HMM and SVM were used majorly for speech recognition, signature verification, license plate detection and classification etc. Fujii et al. [24] used HMM for gesture recognition. P.Sutthiwan et al. [13] used multi-size block based DCT transformation along with SVM classification for detection of image forgery. Jia Li et al. studied image classification using 2-D HMM [28].

III. Proposed Technique

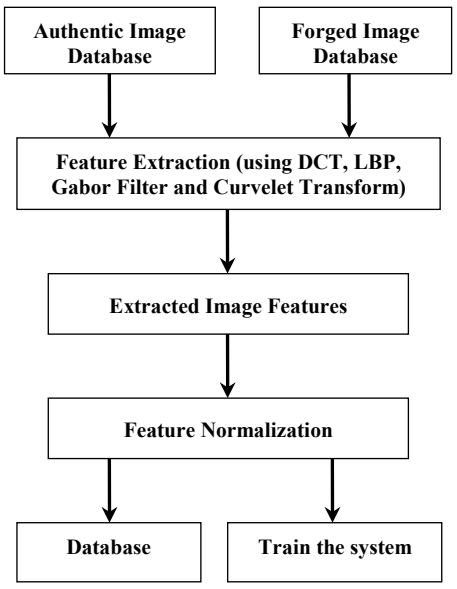

Fig. 2. Training Phase/ Learning Phase of the technique

Objective of the proposed technique is to authenticate the image. The image is represented by its model in the transformed domain in terms of its feature vector. Feature

extraction techniques such as DCT, LBP, Gabor Filter and Curvelet Transform were used. The feature vectors are normalized by dividing them by the maximum of their values and multiplying them by 50. Since HMM can process only non-zero values, 1 is added to all feature vector values. A feature vector set is thus generated for each image. The system is then trained.

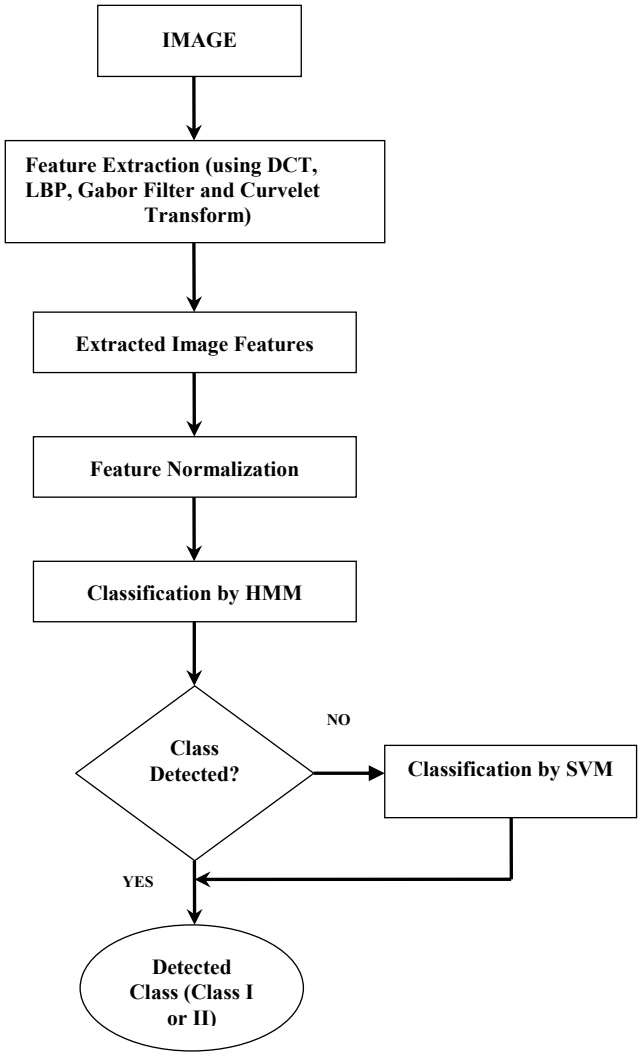

Fig. 3. Testing Phase of the technique

Training is performed using 1250 authentic and 1250 forged images. The images are randomly selected from the CASIA image database of authentic and forged colored images released by the Institute of Automation, Chinese Academy of Sciences. The trained HMM and SVM models are then used to detect image forgery. During testing, a comprehensive feature vector set of the test image is generated. Firstly, data classification is done by HMM. A search is performed for the best combination of states for the maximum a posteriori possibility. This is performed by what is called the 'log likelihood' estimator. When the likelihood value is infinite, the image has to be classified using SVM. Recall that the system has been already trained by SVM. The image is classified in to either Class I (Authentic image) or Class II (Forged image). Fig. 2 shows the training phase and Fig. 3 shows the testing phase of the technique.

IV. Feature Extraction Techniques

A. Discrete Cosine Transform (DCT)

DCT transforms the information available in spatial domain in to frequency domain in terms of DCT coefficients. The size of the DCT coefficient matrix is same as that of the image matrix. DCT does not in any way compress the image. It just reduces the number of coefficients required to represent the data. DCT separates the image into several spectral sub-bands of differing importance. DCT coefficients are divided in to three categories. They are:

i. Low Frequency Components
ii. Middle Frequency Components
iii. High Frequency Components

The low frequency components are associated with light conditions. The middle frequency components contain the useful information encapsulated in the image. The high frequency components are a representation of noise and small details/variations. Forgery in the image introduces variations/high frequency components in the image. Hence DCT can be effectively used for forgery detection [16]. Block-based DCT and DCT of the entire image are the two ways in which DCT can be implemented. Since forgery changes the local frequency distribution of the test image, these changes are reflected in the block based DCT array. However for the purpose of simplicity of computation the global DCT implementation has been used in the proposed technique. This also leads to faster response. Over a uniformly distributed data, the DCT may be said to approximate the Karhunen- Loeve Transform (KLT). Furthermore, unlike the KLT, the DCT does not require a training set. For u=0, 1.....M and v=0, 1.....N the 2-D DCT is given by [16]:

$$F(u,v) = \frac{1}{\sqrt{MN}}\alpha(u)\alpha(v)\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\cos\left(\frac{(2x+1)u\pi}{2M}\right)\cos\left(\frac{(2y+1)v\pi}{2N}\right) \quad (1)$$

Where $\alpha(\omega) = \frac{1}{\sqrt{2}}$ for $\omega = 0$

1 otherwise

Here f(x, y) represents the intensity function of the image. Uniform image sub-bands are produced by sorting of DCT coefficients of each image block. Let the coefficient sub-band be Si(x, y) and i here denotes the specific sub-band. The mean represents the average intensity value of the image and the standard deviation represents the deviation of the intensity values about the mean position. The mean and the standard deviation are calculated as:

$$\mu_i = \int\int |S_i(x,y)| dxdy$$

$$\sigma_i = \sqrt{\int\int (|S_i(x,y)| - \mu_i)^2 dxdy}$$

We are interested primarily only in the low frequency sub-band of the image. The magnitude component of the image (DC coefficient) is also calculated and stored. The magnitude component, mean and standard deviation from

the feature vector pertaining to the DCT based feature extraction method.

B. Gabor Filter

Prior to any processing, the images are converted to gray-scale format and are resized to a 256 x 256 size. The general functional form of 2D Gabor filters is [7]:

$$h(x,y;f,\theta)=\frac{1}{\sqrt{\pi\sigma_1\sigma_2}}\exp(-\frac{1}{2}(\frac{R_1^2}{\sigma_1^2}+\frac{R_2^2}{\sigma_2^2})).\exp(i(\mathrm{f}_x x+\mathrm{f}_y y)) \quad (2)$$

Where,

$$\mathrm{R}_1=x\cos\theta+y\sin\theta \qquad \mathrm{R}_2=-x\sin\theta+y\cos\theta$$

$$f_x=f\cos\theta \qquad f_y=f\sin\theta$$

$$\sigma_1=\frac{c_1}{f} \qquad \sigma_2=\frac{c_2}{f}$$

Gabor filters have sinusoidal shape in the spatial domain and are limited by a Gaussian window. They may be said to be frequency sensitive and orientation sensitive band-pass filters. The central frequency is f, the orientation is θ and σ_1 and σ_2 are the standard deviations of the two dimensional Gaussian window. Here c_1 and c_2 are constants. The input image is convoluted with Gabor function as per (3).

$$G(x,y;f_k,\theta_m)=\sum_m\sum_n f(x-m,y-n)\,\mathrm{h}(m,n;f_k,\theta_m) \quad (3)$$

A Gabor Kernel is defined and is convoluted with the image. We then have a 256 x 256 output matrix whose column wise mean is calculated and these 'means' are our feature vectors. Total number of feature vectors generated is thus equal to 256. However Gabor Filter is orientation sensitive. Hence we obtain features by rotation from 0^o to 360^o in steps of 10^o to make the calculated feature vectors rotation independent.

C. Local Binary Patterns (LBP)

LBP is a powerful technique for texture classification. Any forgery invariably breaks the local texture information and introduces noise into the image. Hence obtaining the variations introduced in the local texture description can be thought as a good measure of image forgery [8][14]. Consider a gray-scale image. Let g_c denote the gray level of the center pixel. Let g_p denote the gray level of a sampling point in an evenly spaced circular neighbourhood of P pixels of radius *R* around the center pixel.

$$\mathrm{g}_\mathrm{p}=f(x_p,y_p)$$

$$x_p=x_c+R\cos(2\pi p/P)$$

$$y_p=y_c-R\sin(2\pi p/P)$$

The texture in a local neighbourhood of a gray image can be assumed to be a joint distribution of gray-scale values of its P neighbours [14].

$$T=t(g_c,g_0.....g_{P-1}) \quad (4)$$

Subtracting the value of g_c we get,

$$T=t(g_c,g_0-g_c.....g_{P-1}-g_c)$$

We can write the above equation safely without any loss of information. Assuming that the values of g_i-g_c are independent of g_c, the above equation can be approximated as,

$$T\approx t(g_c)t(g_0-g_c.....g_{P-1}-g_c)$$

The function $t(g_0-g_c.....g_{P-1}-g_c)$ can be used to model the local texture data. The term $t(g_c)$ does not contain any useful local texture information. To reduce some complications in the computation and to better its invariance, only the signs of the differences are considered. Let s (z) be the unit step function. The LBP operator is defined as [14]:

$$LBP_{P,R}(x_c,\mathrm{y_c})=\sum_{p=0}^{P-1}s(g_p-g_c)2^p \quad (5)$$

We can interpret the signs of the differences as a P bit binary pattern. This means that a total of 2^P distinct values are plausible for the LBP code. The local texture description can thus be described as a 2^P-bin discrete distribution of LBP codes. A LBP code for each center pixel is calculated by the operator mentioned above. The distribution of these LBP codes is used for calculation of the feature vector, say S.

For obtaining the feature vector, the image is first converted in to gray-scale format. Next, a circular neighbourhood cell with radius five is chosen. The value of the center pixel is compared with its neighbours. If the value of the neighbouring pixel is less than that of center pixel, 1 is written there, else 0 is written. A binary pattern is thus generated. Usually for convenience, this binary pattern is converted to decimal format. Histogram over the cell is then computed and normalized. The feature vector is obtained by concatenating all the normalized histograms.

D. Curvelet Transform

Curvelet transform is used to represent images at different scales and different orientations. Features in an image can be points, edges, lines, specific variations etc. Position, orientation, scale characterizes these features. If the conventional 2-D wavelet transforms are used for feature extraction, say DWT, anisotropic lines and edges may be missed out. Though this problem can be addressed to some extent using the complex wavelet transform, it is difficult to design these wavelets with good filter characteristics etc. To override this problem of missing directional selectivity, Multi-resolution Geometric Analysis (MGA) using the Curvelet transform is proposed. The Ridgelet Transform represents straight-line singularities. The block-based Ridgelet transform analyzes the local line or the curve singularities. This was known as the I- generation curvelet transform. Recently the II-generation curvelet transform [called the Fast Digital Curvelet Transform (FDCT)] has

become popular in image processing applications.

A ridgelet is defined by [9]:

$$\psi_{a,b,\theta}(x,y) = \mathrm{a}^{1/2}\,\psi(\frac{x\cos\theta + y\sin\theta - b}{a}) \quad (6)$$

$a > 0$ is the scale parameter, $b \in \mathbb{R}$ is the translation parameter and $\theta \in [0, 2\pi)$ is the orientation parameter. If $f(x,y)$ is the image function, the continuous ridgelet coefficients are defined by [9]:

$$\Re_{\mathrm{f}}(a,b,\theta) = \iint \psi_{a,b,\theta}(x,y) f(x,y)\,dxdy \quad (7)$$

The ridgelets are constant along the line $x\cos\theta + y\sin\theta = k$ where k is a constant. In the implementation of the transform, the input image is decomposed in to several set of sub-bands and ridgelet analysis is performed on these sub-bands. Before feature extraction, the image is converted in to gray scale format and divided in to overlapping blocks B [i, j] of size $N \times N$. Then, the Fast Discrete Curvelet Transform [9] [29] is calculated using (8)

$$CT(a,b,\theta) = IFFT(FFT(\mathrm{B}[i,j]) \times FFT(\psi_{a,b,\theta}[i,j])) \quad (8)$$

Each block is thus decomposed in to three scales. Scales 1, 2 and 3 have 1,16,1 sub-bands respectively. The curvelet coefficients for the three levels are given by (9).

$$CT = \{(ct_{1,1}), (ct_{2,1}, ct_{2,2}, ..., \mathrm{ct}_{2,16}), (\mathrm{ct}_{3,1})\} \quad (9)$$

Mean values for the three scales are then computed. The above analysis was for a particular value of orientation θ. The process is repeated for 16 different values of θ. Mean and standard deviation values generated for $\theta = \theta_1, \theta_2, ..., \theta_{16}$ constitute the feature vector.

A comprehensive feature vector set is thus generated. Since HMM expects a finite number of states for analysis, feature vectors are divided by maximum of their values and multiplied by 50. Hence we have feature vectors ranging from 0-50. However HMM can process only non-zero values. Hence we add one to each feature vector, so that all the feature vectors are positive and lie in a definite range.

V. Decision Making

HMM and SVM are used for image classification. Both are used in the training phase. However during testing, the image is first classified by HMM. When the classification by HMM fails due to the infinite likelihood value, the image is classified by SVM. Image is either classified in to Class I (Authentic image) or Class II (Forged image).

A. Classification using HMM

HMM is a model in which the system is assumed to be a Markov process with hidden states. However what we can observe, of the hidden states, is a sequence of observations. At any discrete instant of time, the process is assumed to be in one of the finite states. Hence an observation at any instant is assumed to be a random function of the state in which process is at that instant. Hence the states can't generally be determined just by looking at the sequence of observations. A fixed probability depending only on upon the state at the preceding instant decides the transition between two states.

Let there be M states in the process and let $Q = \{\mathrm{q}_1, \mathrm{q}_2 ... \mathrm{q}_M\}$. Let at time t the system be in the state j and the probability that at the instant $t-1$ is in the state i is $a_{i,j}$. Hence the probability of the transition from state i to state j will be $a_{i,j}$ i.e. $a_{i,j} = P(\mathrm{q}_j \mid \mathrm{q}_\mathrm{i})$. Further it is assumed that $a_{i,j}$ is time independent. $a_{i,j}$ can be represented as a time independent stochastic transition matrix. Let $A = (a_{i,j})$ be the state transition probability matrix. Let u_t be the observation of the system at the time instant t. Let $u_t = (\mathrm{u}_1, \mathrm{u}_2 ... \mathrm{u}_K)$ be the set of observations. Let π_i be the initial state probability distribution that is, at time $t = 1$. Let the set of output probability distributions be $\mathrm{B} = (\mathrm{b}_\mathrm{j})$ and $b_j(k) = P(\mathrm{u}_\mathrm{t} = u_\mathrm{k} \mid \mathrm{q}_i)$. The complete set of HMM parameters is specified by $\lambda = (\mathrm{A}, \mathrm{B}, \pi)$. A problem associated with the HMMs is finding $\lambda^* = \arg\max P(\mathrm{u}_\mathrm{t} \mid \lambda)$. This problem is solved by the Baum-Welch algorithm or the Forward- Backward algorithm. The incomplete data likelihood condition is given by $P(\mathrm{u}_\mathrm{t} \mid \lambda)$ and the complete data likelihood condition is given by $P(\mathrm{u}_\mathrm{t}, \mathrm{q} \mid \lambda)$. The Q function is given by (10). Here λ' are the initial or guessed estimates of the parameters.

$$\mathrm{Q}(\lambda, \lambda') = \sum_{q \in Q} \log P(u_t, \mathrm{q} \mid \lambda) P(u_t, \mathrm{q} \mid \lambda') \quad (10)$$

The Q function is further modified by using (11).

$$\mathrm{P}(\mathrm{u}_\mathrm{t}, \mathrm{q} \mid \lambda) = \pi_{qo} \prod_{t=1}^{M} a q_{t-1} q_t b_{qt}(u_t) \quad (11)$$

Let us assume that the feature vector of a particular image block follows the Gaussian distribution given its corresponding states and is independent of other feature vectors and states. For Gaussian mixtures, the form of the Q function is given by (12).

$$\mathrm{Q}(\lambda, \lambda') = \sum_{q \in Q} \sum_{n \in N} \log P(u_t, q, \mathrm{n} \mid \lambda) P(u_t, q, \mathrm{n} \mid \lambda') \quad (12)$$

In (12) $n = \{\mathrm{n}q_1 1, \mathrm{n}q_2 2, ..., \mathrm{n}q_\mathrm{M}\mathrm{M}\}$ is the vector that denotes the mixture component in each of the states at each instant. The update equations to be used in the process are [6]:

$$c_{il} = \frac{\sum_{t=1}^{M} P(q_t = i, m_{qt} = l \mid u_t, \lambda')}{\sum_{t=1}^{M}\sum_{l=1}^{N} P(q_t = i, m_{qt} = l \mid u_t, \lambda')} \quad (13)$$

$$\mu_{il} = \frac{\sum_{t=1}^{M} u_t P(q_t = i, m_{qt} = l \mid u_t, \lambda')}{\sum_{t=1}^{M} P(q_t = i, m_{qt} = l \mid u_t, \lambda')} \quad (14)$$

$$\sum_{il} = \frac{\sum_{t=1}^{M} (u_t - \mu_{il})(u_t - \mu_{il})^M P(q_t = i, m_{qt} = l \mid u_t, \lambda')}{\sum_{t=1}^{M} P(q_t = i, m_{qt} = l \mid u_t, \lambda')} \quad (15)$$

Conventional HMMs are used to model 1-D data. However they can be used to model even 2-D images, if the 2-D data is converted to 1-D without any loss of information. Classification using HMM is based on the principle of empirical risk minimization (ERM) wherein a decision rule is chosen which is based on a finite number of observations (training set). HMM has the ability to imbibe the variability and similarity between the features of two images.

a. Training

Each of the forged and authentic images is modelled by estimating the HMM parameters. The HMM is initialized as $\lambda = (\mathrm{A}, \mathrm{B}, \pi)$. Training is performed using 1250 authentic and 1250 forged images. Each of these images is divided in to 4 states (S1, S2, S3, S4). The observation vectors are clustered in to an m-dimensional vector using the k-mean algorithm. These feature vectors values are used to obtain an initial estimate of the observation probability matrix B. The values of the matrix A and π are given in a left to right manner in the HMM structure. Further the HMM model parameters are re-estimated using the Baum-Welch algorithm [6], in order to maximize $P(\mathrm{u_t} \mid \lambda)$. The iterative process stops when the difference for the $(k)^{th}$ and $(k+1)^{th}$ values is less than a previously set threshold (H) as represented in (16).

$$\left| P(u_t \mid \lambda^{(k+1)}) - P(u_t \mid \lambda^{(k)}) \right| < H \quad (16)$$

b. Training

A search is performed for the best combination of the states for maximum a posteriori possibility. The trained HMMs are used for testing and computing the likelihood condition. Feature vectors are generated for the test image. They form the observation sequence, u_t. The probability of the observation sequence given each image model, $P(u_t \mid \lambda_i)$, is calculated using the Viterbi algorithm [6]. The observed vector is labeled with the class model which maximizes the above probability. The test image is recognized as similar to one of the images (say k) in the database (generated during the training phase) if, $P(u_t \mid \lambda_k) = \max_n P(u_t \mid \lambda_i)$. Features of the test image are compared with features of trained images, using the log likelihood estimator. When the process detects a class, the iteration is stopped. However when the likelihood value is infinite with respect to all training instances, the image is tested with SVM. Since SVM is a two class classifier, the detection is accurate and fast.

B. Classification using SVM

Classification in the concerned case is a two-class problem. Generally in detecting whether an image is forged or not, two classes of images are required in the learning phase. One is the class of tampered images and another is the set of authentic images. The objective of the learning phase is to differentiate between the two classes. Many a time it is not easy to find the threshold between the two classes. Hence a classifier–SVM is used. SVM may also be used for multi-class classification tasks. SVM's are superior to Neural Networks in terms of computational efficiency and performance. Support Vector Machines can be defined as systems which use hypothesis space of a linear function in a high dimensional feature space, trained with a learning algorithm from optimization theory that implements a learning bias derived from statistical learning theory. SVM is a powerful tool for classification of data. It is based on the principle of structural risk minimization (SRM). It attempts to maximize the geometric margin which is the Euclidean distance from the hyper plane to the closest instances on either side. It has to be noted that training SVM becomes quite challenging when the number of data points is substantially large. Suppose we are given a set of training data $\{(x_i, y_i) \in \Re^n \times \Re\}$ and the corresponding unknown probability distribution is $\mathrm{P(x, y)}$. We know that $\mathrm{y_i} \in \{-1, 1\}$. $\mathrm{y_i} = -1$ corresponds to say class W_1 and $\mathrm{y_i} = +1$ corresponds to class W_2. The loss function is denoted as $V(y, f(\mathrm{x}))$ where $f(\mathrm{x})$ denotes the predicted value and y denotes the actual value. The task consists of finding a function f such that the expected error $\int V(y, f(\mathrm{x}))\, \mathrm{P(x, y)}\, \mathrm{dxdy}$ is minimized. We have to choose a model from the hypothesis space which is closest to the actual underlying model. A problem which we face is that there are many linear classifiers (hyper planes) which classify that data. However we have to choose the best solution. And the apparent best solution is choosing the optimal hyper plane with the maximum margin. According to the SRM principle there is only one optimal hyper plane separating the two classes with maximum margin δ. The hyper plane can be written as a set of all points satisfying

$w.\mathrm{x}-\mathrm{b}=0$ ('.' denotes the dot-product). Here w is the weight vector, x is the input pattern and b is a threshold. For maximal separation the two hyper planes are defined as:

$\mathrm{w}.\mathrm{x}_\mathrm{i}-\mathrm{b}\geq 1$ for x_i of the class W_2

$\mathrm{w}.\mathrm{x}_\mathrm{i}-\mathrm{b}\leq -1$ for x_i of the class W_1

Support vectors are defined as the points satisfying the equations $\mathrm{w}.\mathrm{x}-\mathrm{b}=1$ or $\mathrm{w}.\mathrm{x}-\mathrm{b}=-1$.All the above conditions can be expressed as $y_i(w.x_i-b)\geq 1$ for all i. The distance between the two hyper planes is $\frac{2}{\|w\|}$. Hence the task is to minimize $\|w\|$ subject to $y_i(w.x_i-b)\geq 1$ for all i. For simplicity of computation this is converted to a quadratic optimization problem to minimize $\frac{\|w\|^2}{2}$ subject to the condition mentioned above [4]. During the testing phase, if $(w.x-b)\geq 0$ the second class is chosen else the first class is chosen. In the practical problems of the real world it is hard to get an exact straight line dividing two sets of data points. There may be some points lying on the wrong side of the hyper plane. Hence the concept of slack variable is introduced. The condition becomes $y_i(w.x_i-b)\geq 1-\xi_i$. This condition allows a point to be ξ_i distance on the incorrect side of the hyper plane. The error penalty function is defined as [33]:

$$f(\xi)=\sum_i \xi_i \tag{17}$$

However one must be cautious as to not to allow huge values of slack variable, otherwise any line may separate the data-sets. This can be done by introducing the Lagrange multiplier. Thus, (18) represents the optimization problem.

$$\min_{w,\xi,b}\max_{\alpha,\beta}\{\frac{w.w}{2}+C\sum_i \xi_i-\sum_i \alpha_i(y_i(w.x_i-b)+\xi_i-1)-\sum_i \beta_i\xi_i\} \tag{18}$$

C represents a regularization constant corresponding to the trade-off between higher margin and smaller error penalty. A larger C implies higher penalty for the errors and reduction of C implies more and more data-points lie on the wrong side of the hyper plane [33]. C has to be chosen according to the needs of the user. We can arrive at the dual form of the above optimization problem by substituting $w=\sum_i \alpha_i x_i y_i$.

In the above example it is assumed that the data is linear and hence the corresponding SVM will be called a linear SVM classifier. However in most practical cases (the concerned case for example) the data is non-linear and not so easily separable. Hence the input data is non-linearly mapped in to a high dimensional space using kernels and the new input data thus formed is now linearly separable. The resulting algorithm is similar to that described above except for the fact that every dot-product is replaced by the non-linear kernel function. Thus the kernel represents a valid inner product in the feature space. The kernel chosen plays an important part in the classification and performance [2]. (19) defines a kernel function.

$$K(x,x')=<\phi(x).\phi(x')> \tag{19}$$

An important point to note while using the SVM classifier is that there is a trade-off between classifier complexity and the classification error and a trade-off between a large margin and a small error penalty. While using SVM, the input data is first mapped non-linearly in to a high dimensional space using a non-linear kernel. In the proposed technique, a common non-linear kernel functions viz. Radial Basis Function (RBF) kernel is used. (20) represents the RBF.

$$K(x,x')=\exp\{-\frac{\|x-x'\|^2}{2\sigma^2}\} \tag{20}$$

Secondly, the system is trained by finding the optimal hyper-plane. Once the system is trained, it will be able to classify any test image in to either of the classes.

VI. Experiments and Results

Before presenting the results, a few parameters are defined. They measure the performance of algorithms, hence called performance parameters [21]. Performance of any system can be measured in terms of sensitivity, specificity and accuracy. A few terms are first defined.

TP (True Positive): Forged image identified as forged
FP (False Positive): Authentic image identifies as forged
TN (True Negative): Authentic image identified as authentic
FN (False Negative): Forged image identified as authentic

$$sensitivity=\frac{TP}{TP+FN} \tag{21}$$

$$specificity=\frac{TN}{TN+FP} \tag{22}$$

$$accuracy=\frac{TP+TN}{TN+FP+TP+FN} \tag{23}$$

$$PPV=\frac{TP}{TP+FP}\ \text{(PPV: Positive Predictive Value)} \tag{24}$$

$$NPV=\frac{TN}{TN+FN}\ \text{(NPV: Negative Predictive Value)} \tag{25}$$

$$FPR=1-specificity\ \text{(FPR: False Positive Rate)} \tag{26}$$

$$FNR=1-sensitivity\ \text{(FNR: False Negative Rate)} \tag{27}$$

Sensitivity relates to the ability of the algorithm to detect a forged image correctly as forged. It is also called as recall. Specificity relates to the ability of the algorithm to identify an authentic image correctly as authentic. Hence a high value of sensitivity and specificity imply better performance of the system. Precision is the probability of truly detecting a forgery. It is also known as Positive Predictive Value (PPV).

A. Performance with different classifiers

In this work, 1250 authentic images and 1250 forged images taken from the CASIA database were used for training the HMM and the SVM model. The training data set used for modeling both HMM and SVM was the same. All the images were available in JPEG format with different Q factors. The size of the images varied from 240×160 to 900×600 pixels. These 2500 images were randomly selected. Since the images were randomly selected, they suffered from different types of attacks like JPEG Compression, Splicing, AWGN, Gaussian Blurring, Copy-Move and combinations of these attacks. All computations were performed on a machine with 16 GB RAM and an Intel Core i7 processor. The software used was MATLAB R2012a. An additional library for SVM [23] was also used. Another 2500 images were used for testing. Out of those, 1250 were authentic images and 1250 were tampered images. The images which were used for testing were different from those used for training. Some images were taken from the CASIA database, while others were sourced from the internet.

TABLE I
RESULTS ACHIEVED

Classifier	Au	Tp	TP	TN	FP	FN
HMM	1250	1250	1013	995	255	237
HMM & SVM	1250	1250	1049	1063	187	51

TABLE II
PERFORMANCE PARAMETERS

Classifier	Sensitivity	Specificity	Harmonic Mean	Accuracy
HMM	0.8104	0.796	0.8046	0.8032
HMM & SVM	0.8392	0.8504	0.8981	0.8448

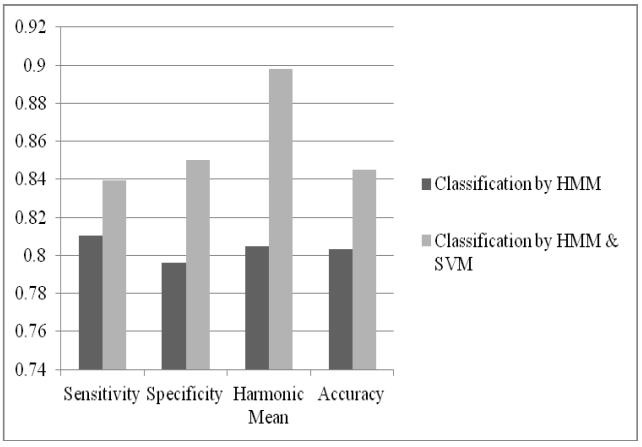

Fig. 4. Performance of the system with different classifiers

Each image was first classified using only HMM. After that, the image was classified using SVM as well. Classification using both HMM and SVM yielded better results. Table I shows the achieved results. In Table I, 'Au' represents the set of authentic images and 'Tp' represents the set of tampered images. Performance parameters of the system in case of both the classifiers are shown in Table II. It is evident from Fig. 2, that the performance of the system improves with SVM. Average time required for classifying an image with both the classifiers is less than that required when the image is classified only with HMM. Average time for classifying an image with HMM classifier was found to be 96.2 seconds, while average time for classifying an image with both the classifiers is about 14.6 seconds. Average time required for training the system was found to be 95.6 minutes.

B. Performance under different types of attack

Performance of the technique under various attacks was also evaluated. Attacks considered were Lossy JPEG Compression, AWGN, Gaussian Blurring, Image Splicing and Copy-Move Forgery. Firstly, 500 images which had undergone lossy JPEG compression were taken. They were divided into 10 sets (each of 50 images) with different quality factors ranging from Q=50 to Q=95. The training dataset was the same as used previously. For each value of quality factor, the number of authentic images and the number of tampered images used for testing is 50. The authentic images used each time are different. As is evident from Fig. 5(a) and 5(b), the values of precision and recall remain almost the same for all quality factors, thus establishing the robustness of the technique. Overall, the precision is found to be 0.874 and the recall is found to be 0.888. Results are represented in Table III.

TABLE III
LOSSY JPEG COMPRESSION

JPEG Quality	TP	TN	FP	FN	Precision	Recall
50	44	42	8	6	0.846	0.88
55	43	44	6	7	0.877	0.86
60	47	46	4	3	0.921	0.94
65	44	44	6	6	0.880	0.88
70	46	44	6	4	0.884	0.92
75	41	40	10	9	0.803	0.82
80	45	42	8	5	0.849	0.90
85	43	43	7	7	0.860	0.86
90	44	45	5	6	0.897	0.88
95	47	46	4	3	0.921	0.94
Total	444	436	64	56	0.874	0.888

Further, 500 images attacked with AWGN (Additive White Gaussian Noise) were taken. They were divided in to 5 sets (each of 100 images) with different SNR values ranging from 45 to 65 dB. For each SNR value, the number of authentic and tampered images used for testing is 100. Again, the values of precision and recall for different SNR values are found to be very near to each other as seen from Fig. 5(c) and 5(d). Overall, the technique achieves a precision of 0.856 and a recall rate of 0.862. Table IV represents the achieved results.

Further, a set of 500 images is taken which are filtered with the Gaussian Blur. They are again divided in to 10 sets of 50 images each, with different values of standard deviation, σ. For each value of standard deviation, the number of authentic images and the number of tampered images used for testing is 50. It is found that as the standard deviation increases, the precision and recall rate increase. Overall, the technique achieves a precision of 0.843 and a recall rate of 0.82. Table V shows the achieved results.

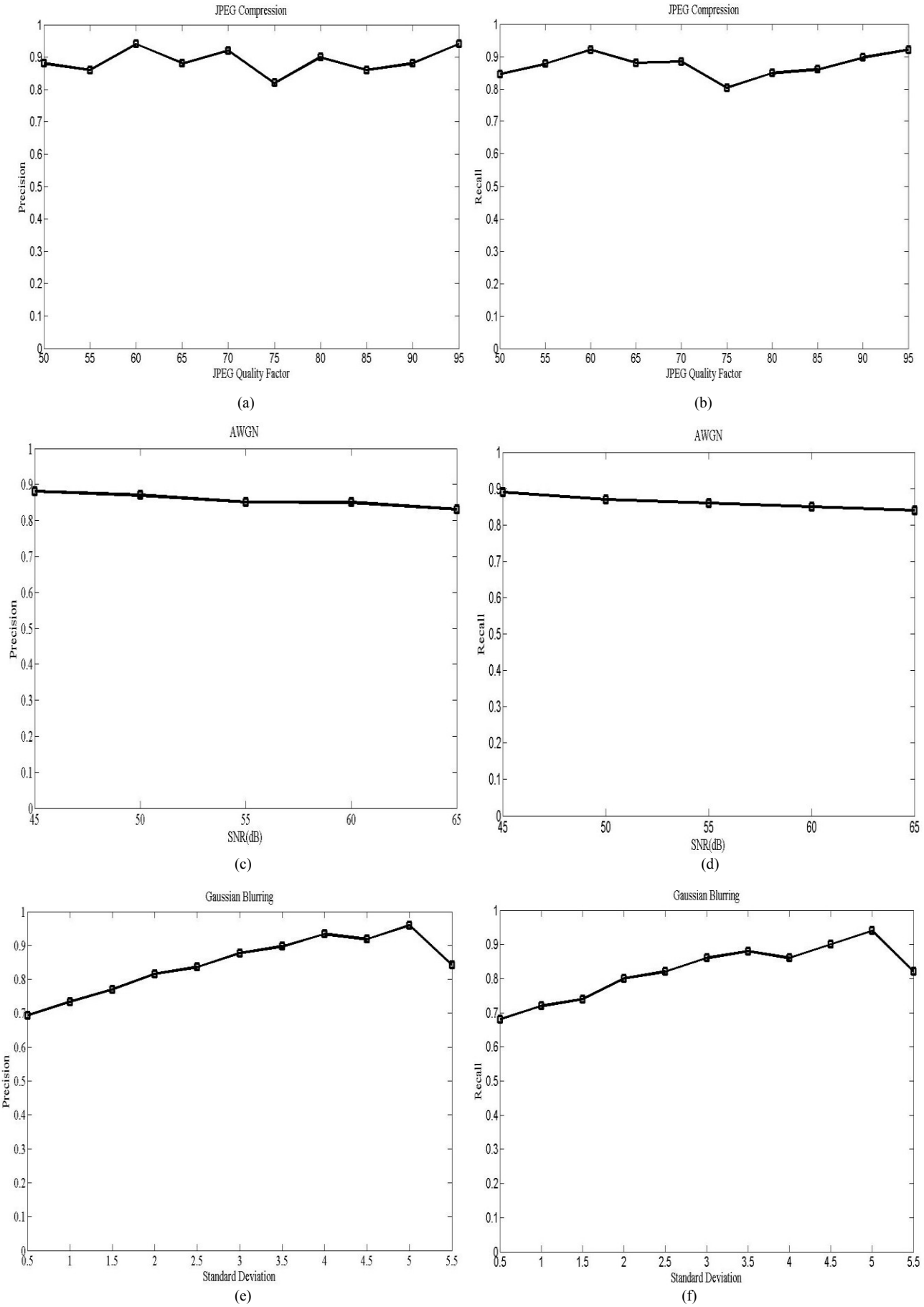

Fig.5. Variation of Precision and Recall (a) Variation of Precision with JPEG Quality Factor (b) Variation of Recall with JPEG Quality Factor (c) Variation of Precision with SNR (dB) (d) Variation of Recall with SNR (dB) (e) Variation of Precision with Standard Deviation, σ (f) Variation of Recall with Standard Deviation, σ.

show the variation of precision and recall with the value of σ.

TABLE IV
ADDITION OF NOISE (AWGN)

SNR (dB)	TP	TN	FP	FN	Precision	Recall
45	89	88	12	11	0.881	0.89
50	87	87	13	13	0.870	0.87
55	86	85	15	14	0.851	0.86
60	85	85	15	15	0.850	0.85
65	84	83	17	16	0.831	0.84
Total	431	428	72	69	0.856	0.862

TABLE V
GAUSSIAN BLURRING ATTACK

σ (S.D)	TP	TN	FP	FN	Precision	Recall
0.5	34	35	15	16	0.693	0.68
1.0	36	37	13	14	0.734	0.72
2.0	37	39	11	13	0.770	0.74
2.5	40	41	9	10	0.816	0.80
3.0	41	42	8	9	0.836	0.82
3.5	43	44	6	7	0.877	0.86
4.0	44	45	5	6	0.897	0.88
4.5	43	47	3	7	0.934	0.86
5.0	45	46	4	5	0.918	0.90
5.5	47	48	2	3	0.959	0.94
Total	410	424	76	90	0.843	0.82

500 images tampered with copy-move attack and image splicing were tested. A database of 500 authentic images was also used. Along with CASIA database, the database of MICC-F220 [25] was also used. The CASIA database primarily contains images which have been tampered with splicing or copy-move attack. Since this database had been used for training the system, the precision achieved was naturally high in this case. Precision achieved was 0.932 and the recall rate was 0.906.

A. Comparative Study

TABLE VI
COMPARATIVE STUDY

Methods	Sensitivity (%)	Specificity (%)	Harmonic Mean (%)	Accuracy (%)
Lukàs-2006	66.93	90.93	77.81	91.82
Mahdian-2008	37.84	82.09	51.80	80.21
Farid-2009	37.70	90.0	53.14	87.80
Li-2009	91.59	45.24	60.56	47.21
Bianchi-2011	59.29	95.17	73.07	93.65
Proposed Technique	90.00	88.00	88.98	89.00

The performance parameters of this technique were compared with those of previously known techniques. As seen from Table VI and Fig. 6, only Li [20], was able to manage a sensitivity of above 90%. However this is achieved at the cost of a very low specificity (below 50 %). Lukàs-2006 [19] provided a good value of specificity and accuracy with an acceptable value of sensitivity.Bianchi-2011[16] provides a high specificity and accuracy. However the sensitivity achieved is less. Same is the case with Mahdian-2008[18]. Farid-2009 achieves a high specificity and accuracy [17]. However the sensitivity is very low (less than 40%). The proposed technique provides acceptable values for all parameters. All values are above 83%. It achieves an accuracy of 84.48%. Hence the proposed technique has a greater ability to detect a forged image as forged and an authentic image as authentic.

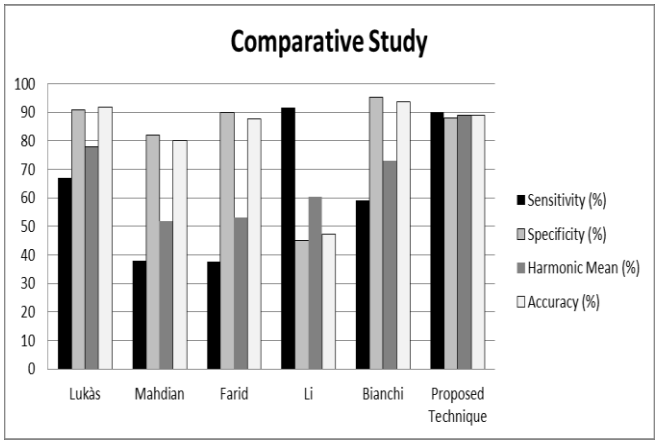

Fig.6. Comparison of other techniques with the proposed one.

VII. CONCLUSION

Various techniques of image tamper were studied. It was found that attempts of forgery detection in images were directed towards a particular type of forgery. However a comprehensive technique which could detect all types of forgeries was not available. This work tries to provide such a comprehensive technique. The technique is based on machine learning algorithms viz. HMM and SVM. While testing, the image was first classified by HMM and then by SVM. It was found that classification with HMM and SVM outperforms image classification with just HMM. Further, the technique was investigated under conditions of different types of attacks. The system was found to be robust to lossy JPEG compression, AWGN and Gaussian Blurring. The technique was compared to the existing techniques. It was found that the existing techniques gave high value of one performance parameters at the cost of other parameters. The proposed method was found to give reasonable values for all performance parameters. In future, the authors hope to make the system more intelligent by incorporating more techniques of feature extraction and data classification.

ACKNOWLEDGMENT

The authors would like to thank the coordinators of CoE, VNIT Nagpur for their valuable financial assistance. They also extend special thanks to Director, VNIT Nagpur for providing them with the needed administrative support.

REFERENCES

[1] Vikas, S., and J. P. Gupta. "Collusion Attack Resistant Watermarking Scheme for colored images using DCT." IAENG International Journal of Computer science, 34:2, pp171-177 (2007).

[2] Deris, Ashanira Mat, Azlan Mohd Zain, and Roselina Sallehuddin. "Overview of support vector machine in modeling machining

performances “in Proceeding of International Conference on Advances in Engineering (ICAE2011), Procedia Engineering 24 (2011), pp. 308-312, 2011.

[3] Zhao, Jie, and Weifeng Zhao. "Passive Forensics for Region Duplication Image Forgery Based on Harris Feature Points and Local Binary Patterns."*Mathematical Problems in Engineering* 2013 (2013).

[4] Ryu, Seung-Jin, Hae-Yeoun Lee, Il-Weon Cho, and Heung-Kyu Lee. "Document forgery detection with SVM classifier and image quality measures." In Advances in Multimedia Information Processing-PCM 2008, Springer Berlin Heidelberg, pp. 486-495, 2008.

[5] Gopi, E. S., N. Lakshmanan, T. Gokul, S. KumaraGanesh, and Prerak R. Shah. "Digital image forgery detection using artificial neural network and auto regressive coefficients." In proceeding of IEEE Canadian Conference on Electrical and Computer Engineering, (CCECE 2006), pp. 194-197, 2006.

[6] Ma, Xiang, Dan Schonfeld, and Ashfaq Khokhar. "A general two-dimensional hidden markov model and its application in image classification.", “In proceedings of IEEE International Conference on Image Processing,(ICIP 2007), vol. 6, pp. VI-41- VI-44, 2007.

[7] Hsu, Hao-Chiang, and Min-Shi Wang. "Detection of copy-move forgery image using Gabor descriptor." In proceedings of IEEE International Conference on Anti-Counterfeiting, Security and Identification (ASID-2012), pp. 1-4, 2012.

[8] Pietikäinen, Matti, Abdenour Hadid, Guoying Zhao, and Timo Ahonen. "Local binary patterns for still images."In Computer Vision Using Local Binary Patterns, , Computational Imaging and Vision 40, Springer-Verlag London Limited, pp. 13-47, 2011.

[9] Qiao, Mengyu, Andrew Sung, Qingzhong Liu, and Bernardete Ribeiro. "A novel approach for detection of copy-move forgery." In proceedings of Fifth International Conference on Advanced Engineering Computing and Applications in Sciences (ADVCOMP 2011), pp 44-47, 2011.

[10] Qazi, Tanzeela, Khizar Hayat, Samee U. Khan, Sajjad A. Madani, Imran A. Khan, Joanna Kołodziej, Hongxiang Li, Weiyao Lin, Kin Choong Yow, and Cheng-Zhong Xu. "Survey on blind image forgery detection." *IET Image Processing* 7, no. 7 (2013): 660-670.

[11] Cozzolino, Davide, Giovanni Poggi, Carlo Sansone, and Luisa Verdoliva. "A comparative analysis of forgery detection algorithms." In Structural, Syntactic, and Statistical Pattern Recognition, Springer Berlin Heidelberg, pp. 693-700, 2012.

[12] Bayram, Sevinc, Husrev Taha Sencar, and Nasir Memon. "A survey of copy-move forgery detection techniques." In Proc. of IEEE Western New York Image Processing Workshop, pp. 538-542, 2008.

[13] Sutthiwan, Patchara, Yun-Qing Shi, Jing Dong, Tieniu Tan, and Tian-Tsong Ng, "New developments in color image tampering detection." In proceedings of IEEE International Symposium on Circuits and Systems (ISCAS-2010), pp. 3064-3067, 2010.

[14] Li, Leida, Shushang Li, Hancheng Zhu, Shu-Chuan Chu, John F. Roddick, and Jeng-Shyang Pan. "An Efficient Scheme for Detecting Copy-move Forged Images by Local Binary Patterns.", Journal of Information Hiding and Multimedia Signal Processing, Volume 4, Number 1,pp. 15-56,January 2013.

[15] Popescu, Alin C., and Hany Farid. "Exposing digital forgeries by detecting traces of resampling", IEEE Transactions on Signal Processing, vol. 53, no. 2,pp.758-767,2005.

[16] Bianchi, T., De Rosa, A., Piva, A., “Improved DCT coefficient analysis for forgery localization in JPEG images” In proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP-2011),pp.2444 –2447, May 2011.

[17] Farid, Hany. "Exposing digital forgeries from JPEG ghosts." IEEE Transactions on Information Forensics and Security, vol. 4, no. 1, pp. 154-160, 2009.

[18] Mahdian, Babak, and Stanislav Saic. "Blind authentication using periodic properties of interpolation." IEEE Transactions on Information Forensics and Security, vol., no. 3, pp.529-538, 2008.

[19] Lukáš, Jan, Jessica Fridrich, and Miroslav Goljan. "Detecting digital image forgeries using sensor pattern noise." In Proceedings of the SPIE, Electronic Imaging 2006, International Society for Optics and Photonics, Volume 6072, pp. 60720Y-60720Y, 2006.

[20] Li, Weihai, Yuan Yuan, and Nenghai Yu. "Passive detection of doctored JPEG image via block artifact grid extraction." Signal Processing, volume 89, Issue 9, pp. 1821–1829, September 2009.

[21] Cozzolino, Davide, Francesco Gargiulo, Carlo Sansone, and Luisa Verdoliva. "Multiple Classifier Systems for Image Forgery Detection." In Proceedings of Image Analysis and Processing (ICIAP 2013), Springer Berlin Heidelberg, pp. 259-268, 2013.

[22] CASIA Tampering Detection Dataset V1.0, 2009. http://forensics.idealtest.org/.

[23] C. C. Chang and C. J. Lin, LIBSVM: A Library for Support Vector Machines. http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[24] Tatsuya Fujii, Jae Hoon Lee, and Shingo Okamoto, Gesture Recognition System for Human-Robot Interaction and Its Application to Robotic Service Task, Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists 2014, IMECS 2014, 12-14 March, 2014, Hong Kong, pp63-68.

[25] Amerini, Irene, Lamberto Ballan, Roberto Caldelli, Alberto Del Bimbo, and Giuseppe Serra, "A sift-based forensic method for copy–move attack detection and transformation recovery", IEEE Transactions on Information Forensics and Security, vol. 6, no. 3, pp.1099-1110, 2011.

[26] Khan, S., and A. Kulkarni. "Detection of copy-move forgery using multiresolution characteristic of discrete wavelet transform" In Proceedings of the 2011 ACM International Conference & Workshop on Emerging Trends in Technology (ICWET 2011), TCET, Mumbai, India, pp. 127-131, 2011.

[27] Birajdar, Gajanan K., and Vijay H. Mankar. "Digital image forgery detection using passive techniques: A survey." *Digital Investigation* 10, no. 3 (2013): 226-245.

[28] Li, Jia, Amir Najmi, and Robert M. Gray. "Image classification by a two-dimensional hidden Markov model", IEEE Transactions on Signal Processing, vol 48, no. 2, pp.517-533, 2010.

[29] Demanet L. CurveLab 2.1.2 2008, Available from: http://www.curvelet.org.

[30] Mahdian, Babak, and Stanislav Saic. "A bibliography on blind methods for identifying image forgery." *Signal Processing: Image Communication* 25, no. 6 (2010): 389-399.

[31] Al-Qershi, Osamah M., and Bee Ee Khoo. "Passive detection of copy-move forgery in digital images: State-of-the-art." *Forensic science international* 231, no. 1 (2013): 284-295.

[32] Fridrich, A. Jessica, B. David Soukal, and A. Jan Lukáš. "Detection of copy-move forgery in digital images." In Proceedings of Digital Forensic Research Workshop, 2003.

[33] J Edson, R. Justino, F. Bortolozzi and R. Sabourin, "A comparison of SVM and HMM classifiers in the off-line signature verification", Pattern Recognition Letters, Volume 26 Issue 9, pp. 1377-1385, 1 July, 2005.

The author received his B.E in Electronics & Communication Engineering from R.G.P.V Bhopal University. He obtained his M.E. in Digital Techniques & Instrumentation in 2010 from R.G.P.V Bhopal University. He is a member of IAENG. He is currently pursuing his doctoral studies at VNIT Nagpur under the supervision of Dr.A.G.Keskar. He has published up to 34 papers in international conferences and journals. He has a teaching experience of 3.5 years. His current research interests are Computer Vision, Circuit Design, and Digital IC Design etc. Mr. Mohammad F. Hashmi is a member of IEEE, ISTE, and IAENG.

The author completed his B.E. from VNIT, Nagpur in 1979 and received gold medal for the same. He completed his M.E. from IISc, Bangalore in1983, receiving the gold medal again. The author is a member of IAENG. He has 26 years of teaching experience and 7 years of industrial experience. He is currently a PROFESSOR at Department of Electronics Engineering, VNIT Nagpur. His current research interests include Computer Vision, Soft Computing, and Fuzzy Logic etc. Dr.Keskar is a senior member of IEEE, FIETE, LMISTE, FIE.

The author is an under-graduate student at VNIT Nagpur. He will complete his B.Tech (Electronics & Communication Engineering) in May, 2015. His areas of interest include Computer Vision, Pattern Recognition, and Signal Processing etc. Mr. Aaditya Hambarde has published one paper in an international conference and one paper in an international journal.