I.J. Image, Graphics and Signal Processing, **2015, 10, 78-85**
Published Online September 2015 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijigsp.2015.10.08

Image Forgery Detection using Multi Scale Entropy Filter and Local Phase Quantization

Saurabh Agarwal[1], Satish Chand[2]
[1,2]Dept. of Computer Engineering, Netaji Subhash Institute of Technology, Sector-3, Dwarka, New Delhi, 110078, India
E-mail: saurabhnsit2510@gmail.com, schand20@gmail.com

***Abstract*—**Performing digital image forgery is very easy due to highly precise image editing tools. There is a concomitant need to have some mechanism to differentiate between a forged image and the original image. In this paper, we propose a passive image forgery detection method that uses entropy filter and local phase quantization (LPQ) texture operator. The entropy filter generally highlights the boundary of the forged regions. It is due to the fact that the entropy filter provides the randomness of a pixel in its local neighborhood. The LPQ operator provides internal statistics of the image based on the phase information. We apply entropy filter on different sized neighborhoods followed by LPQ operator on the CASIA v1.0, CASIA v2.0 and Columbia image forgery evaluation databases. We consider these databases in our experiments because these are standard databases and have been used in most of the methods. Our method provides promising results on both CASIA databases; however, they are comparable on Columbia database with that of the existing state of the art methods.

***Index Terms*—**Image forgery, entropy filter, local phase quantization, splicing.

I. Introduction

To make changes in the contents of an original image for malicious purpose leads to image forgery. Many image editing softwares such as GIMP, Photoshop, Corel Photo Paint, etc. help a forger to create the forged images easily. To verify the authenticity of digital images, different techniques are required which are called image forgery detection techniques. These techniques may be divided into two broad categories: active and passive (blind). In active forgery detection techniques, the image authenticity is verified using digital watermark or digital signature [1-5]. In case of passive forgery detection [6-18, 20, 22], passive or blind forgery detection techniques use the received image only for assessing its authenticity or integrity, without any signature or watermark of the original image from the sender. It is based on the assumption that although digital forgeries may leave no visual clues of having been tampered with, they may highly likely disturb the underlying statistics property or image consistency of a natural scene image which introduces new artifacts resulting in various forms of inconsistencies. These inconsistencies can be used to detect the forgery. This technique is popular as it does not need any prior information about the image. Passive image forgery can be broadly categorize into two types i.e. cloning (copy-move) and splicing image forgery. Cloning, one of the most commonly used image manipulation methods, is used to clone (copy and paste) some portions of an image to hide a person or object in the scene. The splicing combines two or more images to create a forged image. So, passive image forgery detection techniques are more helpful, but hard to implement. In this paper, we discuss a passive image forgery detection technique.

The passive forgery detection techniques are based on color illumination, camera sensor noise, color filter array, image compression (JPEG) artifacts etc. The methods [6-8] locally estimate the color of the illuminant of a image. The illuminant color inconsistency is used to detect image splicing by using a classifier. Due to imperfection in camera hardware various defects are introduced and also create different type of noises in the image like photo response non uniformity (PRNU) and sensor pattern noise. These noises are used to detect image tampering [9,10]. The paper [11,12] discuss that usually one light sensor camera is used instead of the three sensors due to the cost factor. Since we need three different colors (RGB), the color filter array (CFA) is used to create these colors from the single sensor data using interpolation. So by finding the correlation in CFA pattern and the interpolation methods, the forgery can be detected. The techniques [13-15] are based on the principal that in a JPEG image, two types of artifact appear: quantization artifacts in frequency domain and blocking artifacts in spatial domain. In these techniques, it is assumed that the image needs to be saved two times when image tempering is performed that creates double quantization artifact in DCT coefficients and the block synchronization gets disturbed. With the help of these artifacts forgery detection can be done easily.

Some forgery detection methods [16-20] also use statistical features to detect forgery. The paper [16] uses the bicoherence magnitude and phase histograms to distinguish between the spliced and authentic images. The bicoherence is the normalized value of the bispectrum of the signal. The paper [17] uses binary similarity measures between the bit planes for forgery detection. It discusses several binary similarity measures such as Sokal & Sneath, Kulczynski, binary mutual entropy etc. to detect contrast enhancement, brightness adjustment, rotation etc. that are generally required for image forgery. The paper [18] extracts different features

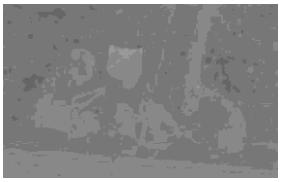
from image run-length histograms and applies the Sobel & Laplacian of Gaussian (LoG) operators for extracting edge base statistical moments features. Using these features, it gives detection accuracy of 84.36% on Columbia gray scale image dataset [19]. The paper [20] discusses the Markov features, which are extracted by capturing the correlations between DCT coefficients and wavelet coefficients. It gives 89.76% detection accuracy on CASIA v2.0 dataset [21]. The paper [22] first applies the steerable pyramid transform on the image and then applies LBP texture descriptor on each sub-band. The histograms of these LBP images are used as feature vector and it achieves 94.89% detection accuracy on CASIA v1.0 dataset [21]. In this paper, we propose an efficient method for image forgery detection in which we extract the image features using entropy filter [23] and texture descriptor. Different texture descriptors like LBP and LPQ are used for texture classification, face recognition[24], and image retrieval. We use Local Phase Quantization (LPQ) [25] as a texture descriptor which is a robust descriptor and gives good classification accuracy.

II. Our Proposed Method

There are many color models to represent the color images. The RGB model is the most popular color model to display and store the color images. However for finding manipulation in the images, it is not suitable because it simply shows intensities of red, green and blue colors. The correlation between red, green and blue components are very high. The RGB model does not differentiate between chromatic and achromatic information of the image. It has been reported in [26] that the YCbCr color model can be more suitable for forgery detection. In YCbCr model, the Y component is luminance component, Cb and Cr are the blue difference and red difference chrominance components, respectively. The Cb and Cr components highlight the forged portion in an image as shown in Figs.1(a)-(e). Fig.1(a) shows the forged RGB image in which the mouth of the central monkey has been replaced with that of the cat. The YCbCr image as shown in Fig.1(b) in which the forged part has different color display. Fig. 1(c) shows the Y image in which the forged part is not distinguishable. Figs. 1(d) and 1(e) show the Cr and Cb images in which the forged part is clearly distinguishable.

(a) (b)

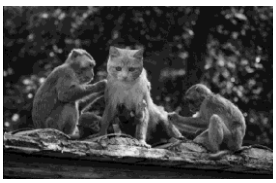
(c) (d)



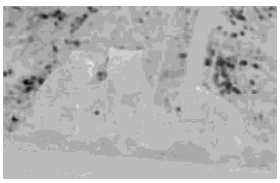
(e)

Fig.1. (a) RGB Forged Image (b) Ycbcr Image (c) Y Image (d) Cb Image (e) Cr Image

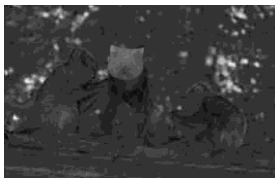
There is another color model, called L*a*b* (Lab), that is more suitable for performing image manipulations than the RGB model. It is a 3-D model based on human visual perception that has one axis for Luminance, denoted by L, one axis for green-red, denoted by a, and one axis for blue-yellow, denoted by b. This color model is machine independent and used for sharpening images and removing JPEG artifacts. The paper [27] uses the Lab color model for image segmentation.



(a)



(b)



(c)

Fig.2. Lab Color Space Images (a) L Channel Image (b) A Channel Image (c) B Channel Image

As evident from Fig. 2(c), the b* channel clearly shows the manipulated part of the image. There are various filters that can help to enhance the visual details of the forged part of the image such as Gabor filter, wavelet filter etc. We found that entropy shows the important characteristic of the image and used as a feature in many applications. Motivated by this fact, we use this important property as a filter in our experiments. The filter based on entropy is called entropy filter [23] and it provide the inheritant information of the image.

We now define entropy followed by entropy filter. It is estimated as follows. Consider a pixel c in image I and let N be the its rectangular neighborhood of size $(2m+1)\times(2n+1)$. For calculating the probability of each pixel in N, we create histogram of neighborhood N. The number of pixels in N will be $N_c = 4mn+2(m+n)+1$. The probability of pixel $q_i \in N$, denoted by p_i, is given by

$$p_i = \frac{n_i}{N_c} \tag{1}$$

where n_i is the number of pixels that have same intensity

as q_i pixel has.

Entropy E_N can be defined as

$$E_N = -\sum_{i=1}^{N_c} p_i \log(p_i) \quad (2)$$

For each point c, E_N provides the uncertainty of the image structure in the local neighborhood N.

The entropy filter replaces each pixel value in an image with the entropy of its neighbors, including itself. The entropy filter basically provides randomness of a pixel of an image in its local neighborhood and it is also used in texture characterization.

Generally, the forged portion of the image have different internal statistics. So when we apply entropy filter due to difference in information, it highlights the forged part as shown in Fig. 2. After applying entropy filter of 3x3 neighborhood on 1(d), 1(e), 2(b) and 2(c) the edges of cat mouth get more visible than other image as shown in Fig. 3(a), 3(b), 3(d) and 3(e). The images 3(c) and 3(f) respectively show the YCbCr and Lab color space entropy filtered images. To extract more information we also apply entropy filter on different size neighborhood.

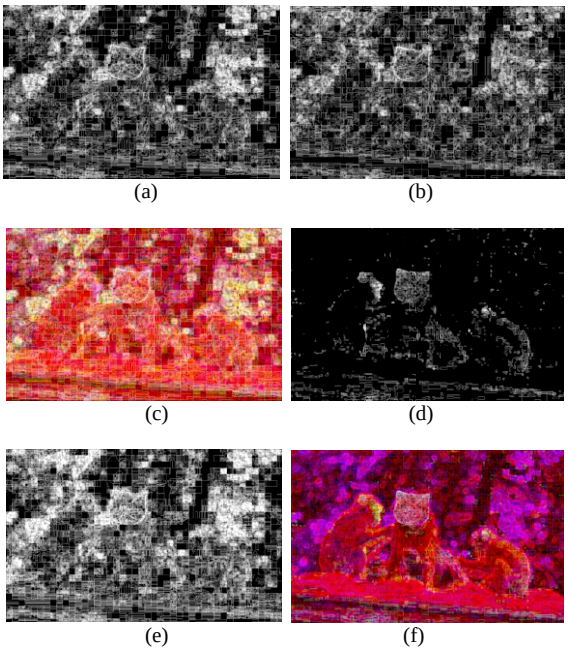


Fig.3. Images After Applying Entropy Filter (a) Cb Entropy Image (b) Cr Entropy Image (c) Ycbcr Entropy Image (d) A Entropy Image (e) B Entropy Image (F) Lab Entropy Image

There are some constraints of the entropy filter: (a) It requires human intervention for finding the forgery and (b) for small size forged part, it is difficult to detect forgery by human being.

In order to remove human intervention for classifying the forged and non-forged images, we extract the features of these entropy-filtered images. For extracting features, we will use phase information. The phase conveys more information regarding signal structure in comparison to signal magnitude and it is also true in case of the image. The phase calculated locally is called local phase and of the whole is called global phase of the image. It is find that local phase have more information than global phase. Many methods of texture classification based on local phase gives good classification accuracy, one of them is LPQ. Motivated by the high accuracy of the LPQ texture descriptor in different classification applications, we extract the features using the LPQ texture descriptor. The LPQ is wildly used for texture classification, face recognition, image retrieval and it gives good results even if the texture and faces are blurred images. We apply the LPQ on the entropy-filtered image to extract the image internal statistics. We briefly describe LPQ operator here.

A. Local Phase Quantization operator

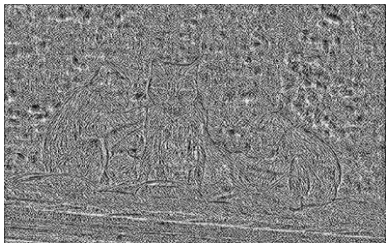
In LPQ, phase is estimated locally by using a short term Fourier transform (STFT) of different neighborhood window at each pixel position of the image.

Let be image I(p) and local m-by-m neighborhoods K_p at each pixel position p. Then STFT is defined as

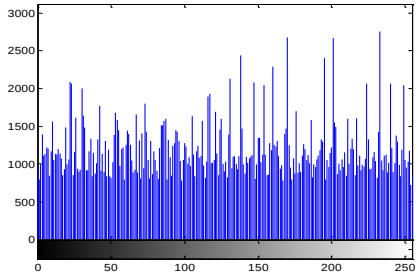
$$S(v, p) = \sum_{y \in k_p} I(p - y) e^{-j2\pi v^T y} \quad (3)$$

where v is the 2-D frequency, K_p is the window can be considered uniform window, Gaussian window or Gaussian derivative quadrature filter pair. In order to reduce the length of the feature vector, angles are quantized to four angles (0, π/2, π, 3π/2). The STFT coefficients are quantized in to a two bit code: first bit for real part and second bit for imaginary part. This gives 8-bit code from four quantized coefficients and these gray values varies from 0 to 255. This process update each pixel value and gives an image is called LPQ image. The neighborhood window K_p can be taken of any odd size, we find experimentally that 3x3 size uniform window gives best results in our method. LPQ used singular value decomposition (SVD) for de-correlation.

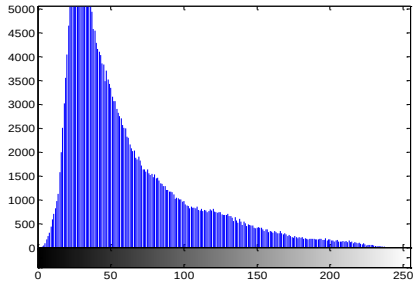
Fig. 4(a) shows gray scale forged image after applying LPQ operator, Fig. 4(b) shows this LPQ image histogram and Fig. 4(c) is the grayscale forged image histogram. It can be seen that histograms Figs. 4(b) and 4(c) are very different. LPQ image histograms have many peaks and valleys. This rebels the internal statistical signature of the image and it helps in classifying forged and non-forged image.



(a) LPQ image



(b) LPQ image histogram



(c) Gray scale image histogram

Fig.4. (a) LPQ Image (b) LPQ Image Histogram (c) Gray Scale Image Histogram

In most of the applications, the LPQ is applied on the non-overlapping blocks of the image and their combined histograms are used as a feature vector. It is due to the fact that the texture and face images have some regular patterns. However, in case of image forgery, it is not necessary that image follow any regular patterns. Therefore, we apply the LPQ on whole image at once and this image histogram is used as a feature vector after normalization.

We may summarize our method as follows:

a Convert RGB image into YCbCr color model and extract Cb and Cr images.
b Apply multi scale entropy filter on Cb and Cr images.
c Apply LPQ operator on these entropy-filtered images.
d Calculate histogram of LPQ image by taking bin size as 256 to obtain feature vector.
e Apply Support Vector Machine (SVM) linear classifier to classify forged and non-forged images.

The block diagram of the above mentioned proposed method is shown in Fig. 5. As shown below, RGB image database of forged and non-forged images are given as input and find the classified forged and non-forged images as output using SVM linear classifier. The multi-scale entropy filter of different sizes are considered like 3x3, 3x5, 5x5, 5x3, 5x7 etc in step b. This highlights the sudden changes in the image. We can also take different bin size histograms like 128, 256, 512 etc. in step d. It is find experimentally and also from other applications of LPQ that 256 bin size gives optimum performance. This is the reason to choose 256 bin size. We apply SVM classifier with linear kernel function because it gives optimum performance for large data.

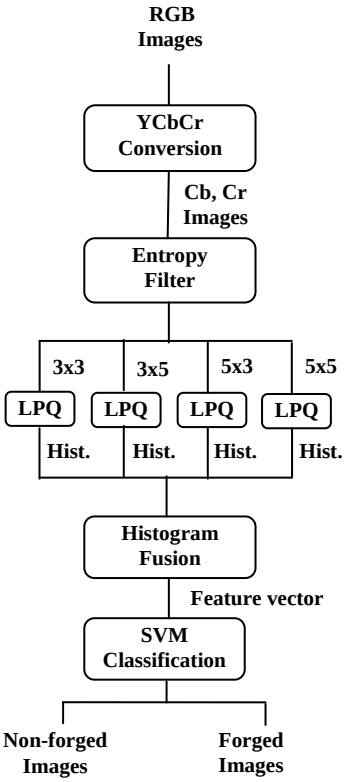


Fig.5. Block Diagram of our Proposed Method

We evaluate the performance of our method on CASIA v1.0, CASIA v2.0 and Columbia color image databases [28]. In these databases forged images are created by performing copy-move or splicing operations. These operations disturb the underlying statistics of an image that help in detecting image forgery.

In next section, we discuss the performance of our method.

III. Experimental Details And Results

In this section, we will discuss about classifiers, image databases and experimental results.

A. Classifier

There are different classifier such as k-nearest neighbor, Fisher's linear discriminant analysis and, SVM, etc. We use SVM as a classifies because it outperforms for binary classes. There are many kernel functions used in SVM like Radial Basis Function (RBF), quadratic function, multilayer perceptron function and linear function. Experimentally we find that the SVM with linear kernel function provides better results especially for large number of features, which has also been supported in [29]. We also use quadratic kernel for Columbia image database because it have only 363 images. We find good detection accuracy using quadratic kernel on Columbia database.

We have carried out 10-fold cross validation 100 times and taken their average to get the stable results. We use the sensitivity, specificity and accuracy to statistically

measure the performance of our proposed work. The sensitivity and specificity define true positive rate and true negative rate, respectively, of each image class. The accuracy measures the percentage of the correctly classified images.

We perform experiments on CASIA v1.0, CASIA v2.0 and Columbia databases by using SVM Linear classifier. Now, we briefly discuss about both CASIA databases and Columbia database.

B. Image database

We carry out our experiment on CASIA v1.0 [17] databases that are usually used to evaluate the performance of the image forgery detection methods. The CASIA v1.0 database contains total 1,721 JPEG color images of size 384x256 in which 800 are authentic (non-tampered) and 921 forged. The CASIA v2.0 database is the extended version of CASIA v1.0 database which contains 12,614 images of different formats JPEG, BMP, and TIFF and sizes 240×160 to 900×600 pixels. This dataset have 7,491 authentic and 5,123 forged images. Out of 5,123 forged images, 3,313 images have been created by copy-move type forgery and 1,810 images by splicing. In both CASIA databases, some forged images are created without doing any post processing operations like blurring etc. and in some images different post processing operations are performed. The forged part size also vary from 30% to 60%.

The Columbia color image database consists 363 TIFF format uncompressed images with sizes range from 757x568 to 1152x768, out of which 183 are authentic and 180 are spliced images. The mostly images are of indoor scenes, only 15% imagers are taken outdoors on a cloudy day. All images are created using 4 cameras: Canon G3, Nikon D70, Canon 350D Rebel XT, and Kodak DCS 330. Using six different pairs of these four cameras, 180 spliced images are created (thirty spliced images from each pair of camera). No post processing operation has been applied to forged spliced images like edge blurring etc.

C. Experimental results

We first perform our experiments on complete CASIA V1.0 database to classify tampered and non-tampered images. Performance is evaluated in terms of sensitivity, specificity and accuracy.

We apply entropy filter by taking the neighborhoods of 3x3, 3x5, 5x3, 5x5, 3x7, 7x3, 5x7, 7x5, 7x7 and 9x9 sizes on both chrominance channels i.e. Cb and Cr in step (b) of our method discussed in section 3. We have found that taking further large neighborhood window deteriorates the performance. It is due to the fact that taking large window size looses the local information.

The luminance channel Y results are not shown in results because it gives very poor detection accuracy i.e. less than 70%.

As evident from Table 1, the accuracy using individual filter window size i.e., 3x3 to 9x9 varies in the range from 82% to 89%. We have explored different combinations of entropy filters of different neighborhood. However, we got good results using the combination of 3x3, 5x5, 3x5, 5x3 neighborhood window sizes and it provided 95.41% detection accuracy. We have also explored the performance by taking different neighborhood window sizes for LPQ operator. We found that taking larger size provided the poor performance. It is possibly due to the fact that the large window size diminishes the internal statistics locally.

Table 1. Performance on Complete CASIA V1.0 Database

Filter Size	Features	Sensitivity (%)	Specificity (%)	Accuracy (%)
3x3	512	94.02	83.81	89.27
3x5	512	93.83	84.75	89.61
5x3	512	94.13	84.71	89.75
5x5	512	92.29	85.13	88.96
7x7	512	91.49	79.99	86.14
3x7	512	91.44	82.15	87.12
7x3	512	92.91	84.81	89.14
5x7	512	91.38	82.01	87.02
7x5	512	91.91	84.69	88.55
9x9	512	88.87	75.7	82.75
3x3,5x5, 3x5,5x3	2048	97.65	93.16	95.41

Out of 921 forged images in CASIA v1.0 database, 469 are spliced images and 452 copy-move forge images. We have also applied our method on both types of forged images separately to check the robustness of our method. We have obtained best performance 96.17% on spliced images as shown in Table 2. In case of copy-move forge images we have obtained best performance 95.23% as shown in Table 3.

Table 2. Performance on CASIA V1.0 with Spliced Forged Images

Filter Size	Features	Sensitivity (%)	Specificity (%)	Accuracy (%)
3x3	512	95.69	95.14	94.18
3x5	512	90.08	95.93	93.77
5x3	512	91.96	96.57	94.87
5x5	512	87.38	95.98	92.82
7x7	512	86.19	94.41	91.37
3x7	512	86.23	94.02	91.77
7x3	512	86.48	93.89	91.87
5x7	512	86.13	93.37	90.32
7x5	512	86.29	93.69	90.56
9x9	512	84.13	92.25	88.31
3x3,5x5, 3x5,5x3	2048	93.23	97.89	96.17

Table 3. Performance on CASIA V1.0 with Copy-Move Forged Images

Filter Size	Features	Sensitivity (%)	Specificity (%)	Accuracy (%)
3x3	512	95.91	82.22	87.16
3x5	512	96.71	83.54	88.29
5x3	512	96.34	85.44	89.38
5x5	512	95.26	85.17	88.81
7x7	512	93.12	83.21	86.79
3x7	512	93.56	82.11	87.14
7x3	512	93.25	82.34	87.31
5x7	512	92.34	83.01	86.89
7x5	512	92.55	83.28	86.76
9x9	512	92.66	79.68	84.37
3x3,5x5, 3x5,5x3	2048	95.19	96.02	95.23

We also apply the entropy filter of sizes 3x3, 3x5, 5x3 & 5x5 and LPQ on a and b channels of Lab color space images on CASIA v1.0 database. We have carried out over experiments by considering copy-move and spliced forged images separately. Their respective detection accuracies are 88.12% and 90.14%. We have also evaluated the performance of our method on the entire CASIA 1 database that provides 87.52% detection accuracy. The detection accuracy using Lab color space (which is 87.52%) is lower than that of using YCbCr color space (which is 95.41%). This is due to the fact that the boundaries of other regions than the forged one of the Lab images are also highlighted after applying the entropy filter on it, as can be seen in fig. 3(f).

Table 4. Performance on CASIA V1.0 Using Lab Color Space

CASIA v1.0	Features	Sensitivity (%)	Specificity (%)	Accuracy (%)
Complete	2048	89.59	85.12	87.52
Copy-move	2048	85.36	89.64	88.11
Spliced	2048	86.31	92.39	90.14

We further explored the performance of our method by combining the features obtained using YCbCr and Lab color models on CASIA v1.0 database and we got further improvement in the detection accuracy as shown in Table 5.

Table 5. Performance on CASIA V1.0 Using Lab and Ycbcr Color Space

CASIA v1.0	Features	Sensitivity (%)	Specificity (%)	Accuracy (%)
Complete	4096	97.71	94.23	95.87
Copy-move	4096	94.17	96.69	96.03
Spliced	4096	95.67	96.32	96.21

We have carried out experiments on CASIA v2.0 database. This database has 7,491 authentic and 5,123 forged images. The results are shown in Table 6, using the features in YCbCr color model, Lab color model, and their combined features. The detection accuracy in this database is calculated considering entropy filter of sizes 3x3 and 5x5 and it gives very good detection accuracy using both color models separately.

Table 6. Performance on CASIA V2.0 Using Ycbcr and Lab Color Spaces

Color Model	CASIA v2.0	Features	Sensiti-vity	Specifi-city	Accuracy (%)
YCbCr	Complete	1024	99.22	97.73	98.33
	Copy-move	1024	97.66	96.13	96.75
	Spliced	1024	98.82	97.21	97.86
Lab	Complete	1024	98.63	96.55	97.41
	Copy-move	1024	96.58	96.61	96.57
	Spliced	1024	97.07	96.13	96.51

We have evaluated the performance of our method on CASIA databases, which have relatively large size. We now apply our method on Columbia database, which is of small size, consisting of just 363 images.

We have applied the SVM classifier with both the kernels i.e. Linear and Quadratic on Columbia database. We have obtained the detection accuracy as 81.52% using SVM classifier with linear kernel and 88.41% using SVM classifier with quadratic kernel. It has been reported in literature that the SVM classifier with Linear kernel provides good results on large datasets [29]. In case the database is of small size like, the SVM classifier with Quadratic kernel provides good results [29]. This finding has also been observed in our experimental results as shown in Table 7.

Table 7. Performance on Columbia Database Using Linear and Quadratic Kernels

Color Model/ Kernel function	Sensitivity (%)	Specificity (%)	Accuracy (%)
Lab/ Linear	87.21	80.05	83.59
Lab/ Quadratic	86.18	82.89	84.52
YCbCr/ Linear	90.11	73.55	81.76
YCbCr/ Quadratic	90.94	85.35	88.13
Lab+YCbCr/ Linear	88.39	74.41	81.35
Lab+YCbCr/ Quadratic	93.07	83.86	88.41

Generally the features of the members in a class have high correlation and the members in different classes have low correlation and such features may be termed as optimal features. Sometimes there are some features in a class that equally represent the other class and these

features may be termed as weak features. While considering such features in classification the detection accuracy deteriorates. If we are able to discard such features, the detection accuracy improves. There are some methods which helps to obtain optimal features such as sequential floating forward selection (SFFS), sequential forward selection (SFS), mutual information (MI), statistical dependence (SD) [30]. These types of methods follow two main approaches, one is by selecting optimal features (it is called feature selection) and other is by discarding bad features (it is called feature filtering). The SFFS and SFS belong to feature selection methods and MI and SD belong to feature filtering methods. The feature selection methods are multifold computational intensive as compared to feature filtering algorithms. Therefore, we will apply feature filtering algorithm SD. This algorithm provides a score value for each feature to reflect its usefulness, according to which a chosen number of features having the highest values are selected. The SD between the feature values $f \in F$ (feature vector) and the class labels $c \in C$ (number of classes i.e. 2) is given by

$$SD = \sum_{f \in F} \sum_{c \in C} p(f,c) \frac{p(f,c)}{p(f)\,p(c)} \tag{4}$$

where $p(f)$ and $p(c)$ are the probability densities of f and c, and $p(f,c)$ is the joint density.

The larger value of SD represent high dependency between the feature values and the class labels.

After getting the optimal features using SD method we applied SVM classifier with quadratic kernel that provided 91.14% detection accuracy on Columbia database as shown in Table 8.

Table 8. Performance on Columbia Database using Feature Scoring Algorithm SD

Color Model	Sensitivity (%)	Specificity (%)	Accuracy (%)
Lab+YCbCr	93.51	88.63	91.14

We evaluate the performance of our method on CASIA databases by applying SD feature filtering method; however, we did not get any improvement in the detection accuracy. In fact the detection accuracy obtained was almost same as that obtained without using SD feature filtering method.

We have also given the comparative performance of our method with that of the state of the art methods [22, 31, 20] as shown in Table 9. The method [31] extracts the features using the multi-resolution Weber law descriptor and it gives 93.33% detection accuracy on CASIA v1.0 database. The paper [22] gives 94.89% detection accuracy on CASIA v1.0, whereas Our method gives 95.41%, which is better than both the methods [22, 30].

The paper [20] uses the Markov features in both DCT domain and wavelet domain of the images and it gives detection accuracy 89.76% on CASIA v2.0. The paper [22] gives 97.33% detection accuracy on the same database CASIA v2.0, whereas our method gives 98.33%, which is better than both the methods [20, 22].

In [22] the detection accuracy has been reported as 96.39% on Columbia database. Our method provides 88.41% without feature selection and 91.14% using SD feature selection algorithm on the same database. Basically there are two reasons that our method detection accuracy is less in comparison to method [22]. First reason is that in Columbia database, no post processing operation is performed that makes detection difficult. Second, from experiments we found that for sixty spliced images that uses cameras Canon 350D Rebel XT gives only 82% detection accuracy. Other four pairs of camera images (120 images) give detection accuracy up to 97%.

Table 9. Comparison of Our Method with Other Methods

Methods	Columbia	CASIA v1.0	CASIA v2.0
Our method	91.14%	95.41%	98.33%
Method [22]	96.39%	94.89%	97.33%
Method [31]	-	93.33%	-
Method [20]	87.52%	-	89.76%

IV. Conclusion

We have discussed a image forgery detection method that is based on entropy filter and local phase quantization (LPQ) texture operator. The entropy filter highlights the random changes in the images that helps in locating forged part. The LPQ operator provides the information about the internal statistics of this entropy filtered image that helps in classify forged and non-forged images. Our method work equally well for both type of forged images i.e. copy-move and spliced images. Our method provides 95.41%, 98.33% and 91.14% detection accuracies on CASIA v1.0, CASIA v2.0 and Columbia databases respectively.

References

[1] D. Kundur and D. Hatzinakos, "Digital watermarking for telltale tamper proofing and authentication," proceedings of the IEEE 87(7), 1167–1180 (1999).

[2] R. Venkatesan, S.-M. Koon, M. H. Jakubowski, and P. Moulin, "Robust Image Hashing," IEEE International Conference on Image Processing, ICIP 2000, Vol. 3, Sept. 2000, pp. 664–666.

[3] C. S. Lu and H. Y. M. Liao, "Structural digital signature for image authentication: An incidental distortion resistant scheme," proceedings of Multimedia Security Workshop, 8th ACM International Conference Multimedia, pp. 115-118, 2000.

[4] C. Rey and J. Dugelay, "A survey of watermarking algorithms for image authentication," EURASIP Journal on Applied Signal Processing 2002(6), 613–621 (2002).

[5] P. L. Lin, C. K. Hsieh, and P. W. Huang, "A Hierarchical Digital Watermarking Method for Image Tamper Detection and Recovery," Pattern Recognition, Vol. 38, Issue 12, 2005, pp. 2519-2529.

[6] S. Gholap and P. K. Bora, "Illuminant color based image forensics," TENCON proceedings of IEEE Region 10 Conference, 2008, pp. 1–5.
[7] X. Wu and Z. Fang, "Image splicing detection using illuminant color inconsistency," proceedings of IEEE International third Conference of Multimedia Information Networking and Security (MINES), Nov. 2011, pp. 600–603.
[8] T.J. de Carvalho, C. Riess, E. Angelopoulou, H. Pedrini and A. de Rezende Rocha, "Exposing Digital Image Forgeries by Illumination Color Classification," IEEE Trans. Information Forensics and Security, vol. 8, no. 7, pp.1182 -1194, 2013.
[9] J. Lukas, J. Fridrich, and M. Goljan, "Detecting digital image forgeries using sensor pattern noise," proceedings of the SPIE Electronic imaging, security, stegnography, and watermarking of multimedia contents VIII, vol. 6072, 2006.
[10] G. Chierchia, G. Poggi, C. Sansone and L. Verdoliva, "A Bayesian-MRF approach for PRNU-based image forgery detection," IEEE Trans. Information Forensics and Security, on 9(4), 554-567, 2014.
[11] Popescu, Alin C., and Hany Farid. "Exposing digital forgeries in color filter array interpolated images." IEEE Transactions on Signal Processing, 53, no. 10, 3948-3959, 2010.
[12] Mahdian, Babak, and Stanislav Saic. "Blind methods for detecting image fakery," IEEE Aerospace and Electronic Systems Magazine 25.4, 18-24, 2010.
[13] H. Farid, "Exposing digital forgeries from JPEG ghosts," IEEE Transaction on Information Forensics and Security, vol. 4, no. 1, pp. 154–160, Mar. 2009.
[14] X. Feng and G. Doërr, "JPEG recompression detection," Proceedings of the SPIE-Media Forensics and Security II, vol. 7541 of , 75410J, Jan. 2010.
[15] Y.-L. Chen and C.-T. Hsu, "Detecting recompression of JPEG images via periodicity analysis of compression artifacts for tampering detection," IEEE Transaction on Information Forensics and Security, vol. 6, no. 2, pp. 396-406, Jun. 2011.
[16] Ng, Tian-Tsong, and Shih-Fu Chang. "A model for image splicing," Image Processing, 2004, ICIP '04, International Conference on. Vol. 2. IEEE, 2004.
[17] S. Bayram, I. Avcibas, I. Sankur and I. Memon "Image manipulation detection with binary similarity measures", proceedings of European Signal Processing Conference, 2005.
[18] J. Dong, W. Wang, T. Tan and Y. Q. Shi, "Run-length and edge statistics based approach for image splicing detection." Digital Watermarking. Springer Berlin Heidelberg, 76-87, 2009.
[19] Ng, Tian-Tsong, Jessie Hsu, and Shih-Fu Chang, "Columbia image splicing detection evaluation dataset," 2009.
[20] Z. He, W. Lu, W. Sun and J. Huang "Digital image splicing detection based on Markov features in DCT and DWT domain," Pattern Recognition 45.12, 4292-4299, 2012.
[21] CASIA Tampered Image Detection Evaluation Database, 2010. [Online]. Available: http://forensics.idealtest.org.
[22] G. Muhammad, M. Al-Hammadi, M. Hussain, G. Bebis, "Image forgery detection using steerable pyramid transform and local binary pattern," Machine Vision and Applications, pp. 1-11, 2013.
[23] J. B. Jordan and L. C. Ludeman. "Image segmentation using maximum entropy techniques," Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'84.. Vol. 9. IEEE, 1984.
[24] G. Satyanarayana Murty, J. Sasi Kiran, and V. Vijaya Kumar, "Facial Expression Recognition Based on Features Derived From the Distinct LBP and GLCM," International Journal of Image, Graphics and |Signal Processing (IJIGSP) 6.2 (2014): 68.
[25] T. Ahonen, E. Rahtu, V. Ojansivu, J. Heikkila¨, "Recognition of Blurred Faces Using Local Phase Quantization," proceedings of nineteenth International conference of Pattern Recognition (ICPR), 2008.
[26] W. Wang, J. Dong, T. Tan, "Effective image splicing detection based on image chroma," proceeding of IEEE International conference on Image Processing (ICIP), Nov. 2009.
[27] B. Liu, C. M. Pun and X. C. Yuan, "Digital Image Forgery Detection Using JPEG Features and Local Noise Discrepancies." the Scientific World Journal, 2014.
[28] Y.F. Hsu, S.F. Chang, "Detecting image splicing using geometry invariants and camera characteristics consistency," proceedings of IEEE International Conference on Multimedia and Expo (ICME), pp. 549–552, 2006.
[29] C. Hsu, C. C. Chang, and C. J. Lin, "A practical guide to support vector classification," Technical report, 2005.
[30] Pohjalainen, Jouni, Okko Räsänen, and Serdar Kadioglu.,"Feature selection methods and their combinations in high-dimensional classification of speaker likability, intelligibility and personality traits," Computer Speech & Language 29.1, 145-171, 2015.
[31] M. Hussain, G. Muhammad, S. Q. Saleh, A. M. Mirza, G. Bebis, "Image forgery detection using multi-resolution Weber local descriptors," Eurocon2013, pp. 1570- 1577, Zagreb, Croatia, July 2013.

Author's Profile

Saurabh Agarwal has received his B.Tech from Barkatullah University, Bhopal in Computer Science and Engineering and his M.Tech from Uttar Pradesh Technical University, Lucknow in Software Engineering.

He is pursuing Ph.D in Department of Computer Engineering, Netaji Subhas Institute of Technology, New Delhi, India.

Satish Chand did his M.Sc. in Mathematics from Indian Institute of Technology, Kanpur, India and M.Tech. in Computer Science from Indian Institute of Technology, Kharagpur, India and Ph.D. from Jawaharlal Nehru University, New Delhi, India. Presently he is working as a Professor in Computer Engineering Division, Netaji Subhas Institute of Technology, Delhi, India. Areas of his research interest are Multimedia Broadcasting, Networking, Video-on-Demand, Cryptography, and Image processing.