International Journal on Recent and Innovation Trends in Computing and Communication
Volume: 5 Issue: 7
ISSN: 2321-8169
168 – 174

Statistical Feature based Blind Classifier for JPEG Image Splice Detection

Surbhi Gupta
Research Scholar
Computer Science & Engineering
I. K. Gujral Punjab Technical University
Kapurthala, India
royal_surbhi@yahoo.com

Neeraj Mohan
Assistant Professor
Computer Science & Engineering
I. K. Gujral Punjab Technical University
Kapurthala, India
erneerajmohan@gmail.com

Abstract—Digital imaging, image forgery and its forensics have become an established field of research now days. Digital imaging is used to enhance and restore images to make them more meaningful while image forgery is done to produce fake facts by tampering images. Digital forensics is then required to examine the questioned images and classify them as authentic or tampered. This paper aims to design and implement a blind classifier to classify original and spliced Joint Photographic Experts Group (JPEG) images. Classifier is based on statistical features obtained by exploiting image compression artifacts which are extracted as Blocking Artifact Characteristics Matrix. The experimental results have shown that the proposed classifier outperforms the existing one. It gives improved performance in terms of accuracy and area under curve while classifying images. It supports .bmp and .tiff file formats and is fairly robust to noise.

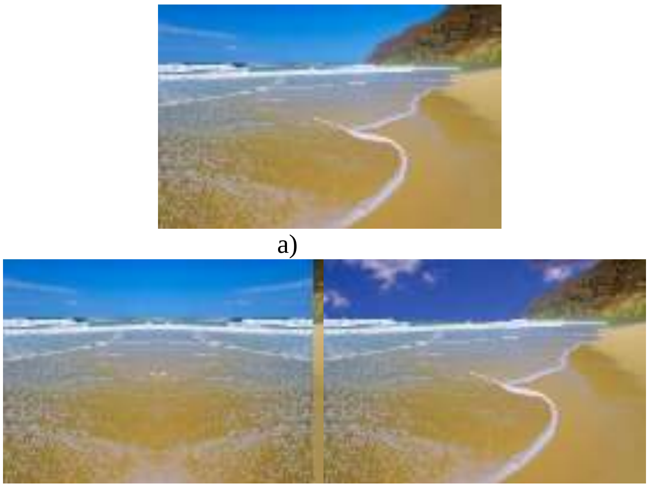
***Keywords**-component: Blocking Artifact Characteristics Matrix (BACM); Image Forensics; Image Splicing; Joint Photographic Experts Group (JPEG) compression artifacts; Support Vector Machine (SVM) classifier.*

*****

I. INTRODUCTION

The readily available software, tools and techniques have made the image processing quite easier these days. Tools developed for enhancement of image are being misused to hide the truth and establish the fallacies. There are enormous ways to manipulate or forge an image. Most common image forgery techniques are copy-move and splicing as shown in Fig. 1. In copy move forgery, some part of the image is cropped, processed and then replicated in the image to either hide or add some content to the image. In splicing, two different images are used to create a new image with new content altogether. Thus, before relying on an image we need to first check its truthfulness using image forensic tools and techniques. These techniques are based on active and passive approaches. In active approach, features like watermark or signature is added to the image which would get distorted if the image is tampered. This is mainly used for sensitive documents and images, as they are highly prone to fakery. In the absence of such active approach, a passive approach needs to be used. Passive approaches do not require any background information about the image rather they extract features and characteristics from the available image only to make a decision.

Most of the image processing tools and digital cameras now days are using Joint Photographic Experts Group (JPEG) format, so, the forensics for this format is very crucial. JPEG image forensics is done either by source or camera detection or by utilizing compression characteristics to identify image tampering. These characteristics are based on quantization and Discrete Cosine Transform (DCT) artifacts present in the image due to double compression.



a)

b) c)

Fig. 1 a) Original image; b) copy move forgery; c) splicing forgery (Dong and Wang, 2011)

Initially, Lukas and Fridrich[1], 2003 and Lukas et al.[2], 2006 proposed image tamper detection by identifying source camera using sensor pattern noise but it fails to correctly classify the regions where the pattern noise was low. Ng and Chang[3], 2004 proposed physics based model to detect image splicing but the detection rate was moderate. Popescu and Farid[4-6] (2004; 2005a; 2005b) presented image resampling and color filter interpolation based methods to detect image splicing. Proposed method[5] doesn't perform well where images with high quality factors were spliced and resaved at a low quality factor. Pan et al.[7] (2004) and Perra et al.[8] (2005) utilized edge based features for detecting blocking artifacts in JPEG images and achieved good results. Fan and Queiroz[9] (2003) introduced Blocking Artifact Characteristics Matrix (BACM) based features to identify double image compression which Luo et al.[10] (2007) used to determine cropping and forgery, but this method gave a low true positive rate. Chen and Hsu[11](2008) investigated the periodic property of blocking artifact by using different features. But this method only performed well when forged image has high quality factor as compared to original image. Pan and Lyu[12](2010) proposed region duplication detection using image key-points and feature vectors as these are robust to usual image transforms. Barni et al.[13] (2010) localized tampering by statistically analysing the image both block and region wise. Bianchi and Piva[14] (2012) categorized the double JPEG compression as either aligned or non-aligned and localized the tampering. Although results presented were very comprehensive but classifier achieved low Area Under Curve (AUC) for spliced images with high Quality Factor. Thing et al.[15] (2012) tried to improve the accuracy of JPEG image tampering detection by considering the characteristics of the random distribution of high value bins in the DCT histograms. Then, Tralic et al.[16](2012) proposed a method to detect re-compression using Blocking Artifact Grid extraction but sufficient illustration of method on different types of images was lacking. Mall et al. [17](2013) proposed a combined hashing index for image which was capable of detecting structural tampering, brightness level adjustment and contrast manipulations. Chang et al.[18] (2013) proposed copy move detection by searching similarity blocks in the image and used similarity vector field to assure the true positives. Recently, Wattanachote et al.[19] (2015) utilized BACM features to identify seam modifications in JPEG images and presented efficient results.

All these researchers contributed significantly in image forensics but only few provided a comprehensive study. The aim of presented work is to design and implement a blind classifier for splice detection of JPEG images at various quality factors with higher accuracy and area under curve. Proposed classifier works for .bmp and .tiff images as well. It is robust to presence of noise in images. It detects image splicing even when pre-processing and post-processing operations have been applied and spliced area vary from small to large. The proposed design and the experimental results obtained are discussed in following sections.

II. Proposed System Design for Splice Detection Classifier

The system design consists of two main components i.e. training and testing of Support Vector Machine (LIBSVM[20]) to classify images as shown in Fig. 2. Image dataset consists of original and spliced images from CASIA[21] database. Dataset is divided as training and testing dataset. Statistical features from these images are extracted from image Blocking Artifact Characteristic Matrix (BACM) which is the mean inter-pixel intensity difference inside and across the JPEG sub-block boundaries. This difference is similar for uncompressed images but when an image is compressed, the discontinuities appear in pixel intensity difference. The statistical features of images from training dataset are fed to SVM and a model is obtained. Then this model is used to test images for their identification as original or spliced.

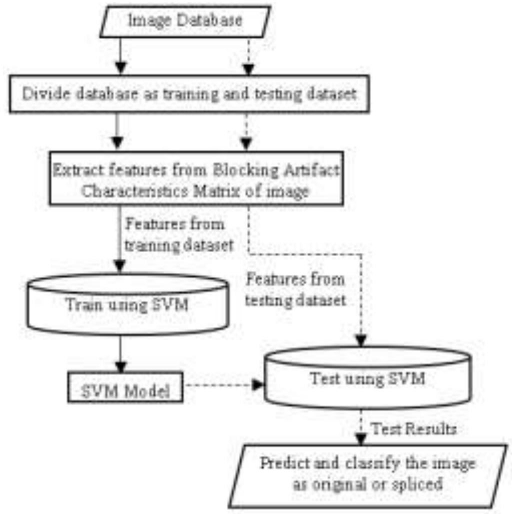


Fig. 2 System Design for proposed JPEG tool

A. *Proposed algorithm for statistical features extraction*

The algorithm used for extracting image statistical features and its complexity is as follows:

Step1: Consider an image I. transform the image I to grayscale such that I_g=rgb_to_gray (I).

Step2: Subdivide the image into sub-blocks of 8 x 8 pixels.

For each sub-block, for every pixel location x, y , where,$1 \le x, y \le 8$

Calculate difference in neighbour pixel intensities $D(x, y)$ as:

$$D(x,y) = |[P(x,y) + P(x+1,y+1)] - [P(x+1,y) + P(x,y+1)] \quad (1)$$

Where,$P(x, y)$ represent intensity of pixel at location x, y.

Calculate $D(x+4, y+4)$.

Calculate absolute difference $D'(x,y) = | D(x+4, y+4 - D(x,y)|$. (2)

Step 3. Calculate energy $K(x, y)$ at each pixel location x , y from each sub-block i as

$$K(x,y) = \sum_{i=1}^{n} D_i'(x,y) \quad (3)$$

Where,n is total number of image sub-blocks.

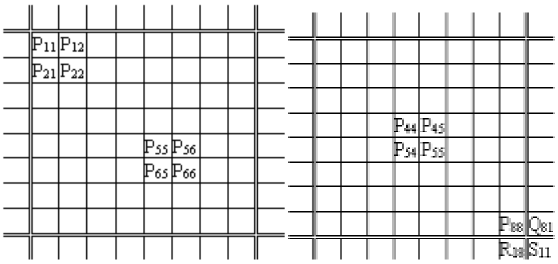
Step 4. Calculate BACM matrix $B(x, y)$ as $B(x,y) = K(x,y)/n$. (4)

Step 5. Extract features F1-F20 from BACM and input them to SVM to obtain the classifier model.

The algorithm works on 2x2 pixel neighbouring in each sub-block. Every pixel is considered neighbour to 4 pixels as shown in Fig 3. Algorithm needs to access each block once and each pixel of the image 4 times to calculate pixel intensity difference. So, the number of access for each pixel is 4 and the complexity is equivalent to O (4n) ≈ O (n). It is linearly dependent on the size of the image. The algorithm's main steps i.e. extracting BACM and defining feature set are further clarified with example in the following two sections.

B. *Extracting BACM*

Blocking Artifact Characteristics Matrix (BACM) is a matrix extracted form DCT blocks of the image. It reveals important features about the image compression history. To extract BACM, grey scale image is subdivided into sub-blocks of 8x8 pixels. For each sub-block and every pixel location the inter-pixel intensity difference is calculated. For example, P, Q, R and S are four consecutive sub-blocks in image. Then for sub-block P, the inter-pixel distance at $x = y = 1$, is calculated as $D(1,1)$ and $D(5,5)$ and the inter-pixel distance at $x = y = 4$ is calculated as $D(4,4)$ and $D(8,8)$ using Eq. (1) as shown in Fig. 3.



a) b)

Fig. 3 Calculation of inter pixel difference a) inside b) across the block boundary

$D(1,1) = |(P_{11} + P_{22}) - (P_{21} + P_{12})|$ and $D(5,5) = |(P_{55} + P_{66}) - (P_{65} + P_{56})|$
$D(4,4) = |(P_{44} + P_{55}) - (P_{54} + P_{45})|$ and $D(8,8) = |(P_{88} + S_{11}) - (Q_{81} + R_{18})|$

Further, the absolute difference $D'(x, y)$ is calculated using Eq. (2). Then energy $K(x, y)$ and then BACM $B(x, y)$ is derived using Eq. (3) & (4). Fig. 4 shows the value of BACM of an original JPEG image at each pixel location. For example, '2.5364' in BACM is the mean value for all pixels intensity differences which are located at (1, 1) in every block.

2.5364	**2.4623**	**2.5075**	**3.5274**	**2.6886**	**2.4883**	**2.6667**	**4.6139**
2.5343	**2.4259**	**2.3567**	**3.4163**	**2.3909**	**2.4273**	**2.5110**	**3.9047**
2.3738	**2.2702**	**2.4554**	**3.5679**	**2.4547**	**2.3628**	**2.3608**	**4.2888**
2.5981	**2.6049**	**2.8265**	**3.5741**	**2.7606**	**2.5171**	**2.6337**	**4.1214**
2.7311	**2.5343**	**2.8656**	**3.9444**	**2.9266**	**2.6317**	**2.7263**	**4.2798**
2.5995	**2.5816**	**2.6399**	**3.5178**	**2.5583**	**2.4067**	**2.5178**	**4.2305**
2.6310	**2.6879**	**2.9005**	**3.7634**	**2.7908**	**2.5583**	**2.7661**	**4.3909**
3.3765	**2.9156**	**3.0391**	**3.7558**	**3.2449**	**3.0192**	**3.4136**	**4.4266**

Fig. 4 Sample BACM for JPEG image

BACM of an image gives important characteristics about it. Experiments conducted on JPEG images at different quality factors revealed that if an image with QF100 is spliced and recompressed at same level the deviation in BACM values increases as compared to original image. This deviation in BACM values increases further if spliced images is recompressed at lower levels as shown in Fig. 5.

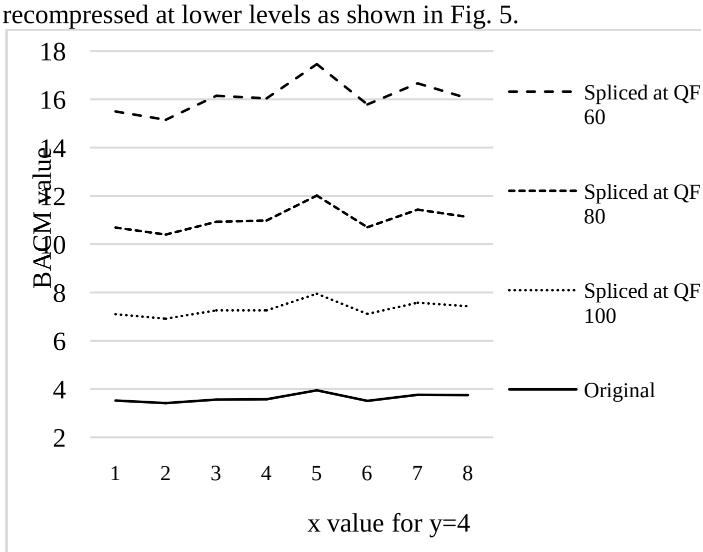


Fig. 5 Comparison of 4th column values of BACM of Au_nat_00093.jpg with its spliced versions at QF100, QF80, and QF60

C. *Defining Feature Set*

After calculating BACM, statistical features need to be defined and extracted. For feature extraction, BACM is divided in various regions. In existing techniques[9, 10, 19], only 7x7matrix from BACM is considered for extracting features but for proposed classifier whole 8x8matrix is considered. Regions in BACM are defined as R1, R2, R3, R4, H1, H2, V1, V2, C1, C2, and C3 and C4 as shown in Fig. 6. Further, BACM is divided as R4, R5, R6, and R7 to extract additional four features.

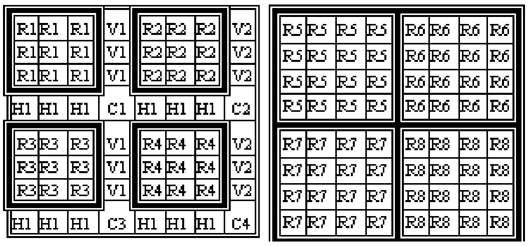


Fig. 6 Division of BACM in regions for extracting statistical features

The first set of features is based on symmetry of horizontal region H1and vertical region V1. For H1 and V1 feature $F1$and $F2$ are extracted as:

$$F1 = \sum_{y=1}^{3} |B(4, y) - B(4, 8 - y)| \quad (5)$$

$$F2 = \sum_{x=1}^{3} |B(x, 4) - B(8 - x, 4)| \quad (6)$$

Where $B(x, y)$ represents BACM matrix value at location x, y. The next set of features is based on symmetry of four regions R1, R2, R3 and R4. Feature $F3$ is based on symmetry of R1 and R2, $F4$ is based on the symmetry of blocks R3 and R4, $F5$ is based on the symmetry of blocks R1 and R3, $F6$ is based on the symmetry of blocks R2 and R4, $F7$ is based on the symmetry of blocks R1 and R4 and $F8$ is based on the symmetry of blocks R2 and R3.

$$F3 = \sum_{x=1}^{3} \sum_{y=1}^{3} B(x, y) - B(x, 8 - y) \quad (7)$$

$$F4 = \sum_{x=5}^{7} \sum_{y=1}^{3} B(x, y) - B(x, 8 - y) \quad (8)$$

$$F5 = \sum_{x=1}^{3} \sum_{y=1}^{3} B(x, y) - B(8 - x, y) \quad (9)$$

ISSN: 2321-8169
168 – 174

$F6 = \sum_{x=1}^{3}\sum_{y=5}^{7} B(x,y) - B(8-x,y)$ (10)

$F7 = \sum_{x=1}^{3}\sum_{y=1}^{3} B(x,y) - B(8-x,8-y)$ (11)

$F8 = \sum_{x=1}^{3}\sum_{y=5}^{7} B(x,8-y) - B(8-x,y)$ (12)

Further six features, $F9 - F14$ are extracted based on percentage of occupancy of centre point C1 against different regions R1, R2, R3, R4, H1 and V1. These are calculated as:

$F9 = C1/\sum_{x=1}^{3}\sum_{y=1}^{3} B(x,y)$ (13)

$F10 = C1/\sum_{x=1}^{3}\sum_{y=5}^{7} B(x,y)$ (14)

$F11 = C1/\sum_{x=5}^{7}\sum_{y=1}^{3} B(x,y)$ (15)

$F12 = C1/\sum_{x=5}^{7}\sum_{y=5}^{7} B(x,y)$ (16)

$F13 = C1/\sum_{y=1}^{7} B(4,y) - C1$ (17)

$F14 = C1/\sum_{x=1}^{7} B(x,4) - C1$ (18)

Next four new features, $F15 - F18$ are extracted based on mean of four sub-regions i.e. R5, R6, R7 and R8 as:

$F15 = \sum_{i=1}^{4}\sum_{j=1}^{4} B(i,j)$ (19)

$F16 = \sum_{i=5}^{8}\sum_{j=1}^{4} B(i,j)$ (20)

$F17 = \sum_{i=1}^{4}\sum_{j=5}^{8} B(i,j)$ (21)

$F18 = \sum_{i=5}^{8}\sum_{j=5}^{8} B(i,j)$ (22)

Last two features $F19$ and $F20$ are based on symmetry of horizontal region H2 and vertical region V2:

$F19 = \sum_{y=1}^{3} |B(8,y) - B(8,8-y)|$ (23)

$F20 = \sum_{x=1}^{3} |B(x,8) - B(8-x,8)|$ (24)

The values for all these features have been studied. Luo et al., 2007 used first fourteen features i.e. $F1 - F14$ based on Eq. 5 to Eq. 18 to classify the images. In addition to these fourteen features another set of six features based on Eq. 19 to Eq. 24 have been added to increase the capability of the classifier. Another set of these features which are based on the Occupancy of centre points C2, C3 and C4 have been studied but are not included in classifier design as less deviation is observed in their feature values. Fig. 7 illustrates an example of feature values for original and spliced images for existing and proposed classifiers for image shown in Fig. 1. First fourteen (1-14) features are common for both the classifiers and next six (15-20) are added in the proposed classifier.

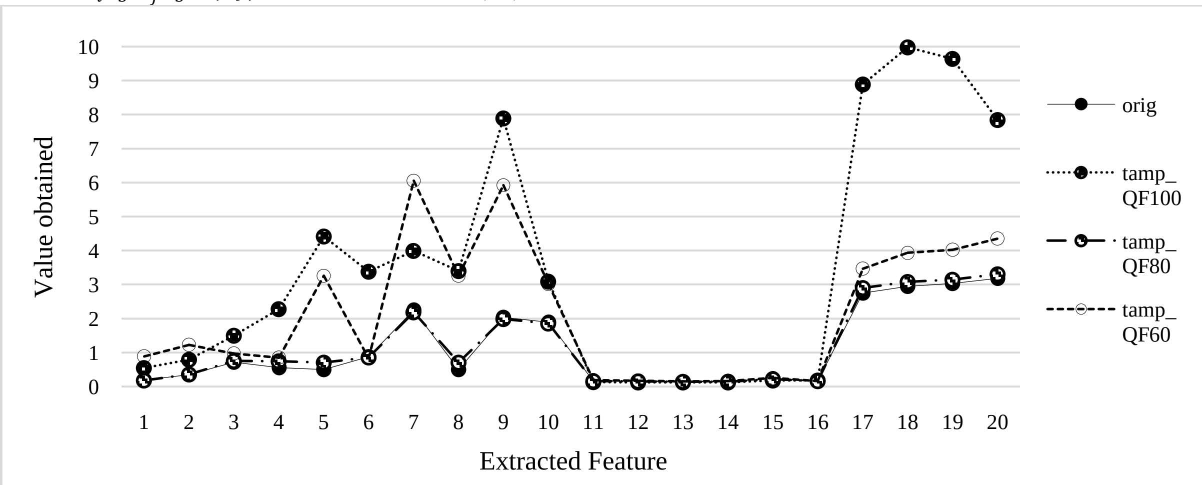


Fig. 7 Representation of Feature values for original (Fig. 1a) and spliced images (Fig. 1c at QF100, QF80, QF60)

III. EXPERIMENTAL RESULTS

The experimental setup consists of images from CASIA V2.0. 875 original and 665 spliced images are taken from the database. Original and spliced images are saved at different Quality factors 60, 80 and 100 to study the classifier performance. Considered spliced images have

a. Randomly crop-and-paste image region(s)
b. Cropped image region(s) processed with resizing, rotation or other distortion
c. post-processed region(s) (processed with operations such as blurring) to finish crop-and-paste operation of the fake image
d. Difference sizes (small and large) of spliced regions
e. Been considered at Quality factor 60, 80 and 100
f. Been considered to be realistic images by human eyes.

Training set consists of 900 images (both original at QF1 and spliced at QF2) and testing set consists of 640 images (both original at QF1 and spliced at QF2). SVM classifier with Radial Basis Function kernel is used. The penalty parameter C is chosen by Grid Search method.

Different features studied are:

- Accuracy and Area Under Curve for proposed classifier
- Accuracy of classification of images with small and large spliced area
- Impact of noise on classifier accuracy

A. Accuracy and Area Under Curve for proposed classifier

The True Positive Rate (TPR), True Negative Rate (TNR), Accuracy (ACC) and Area Under Curve(AUC) for the existing and proposed classifier are compared in Table 1. The TPR for existing classifier drops significantly when the QF of spliced image is high but proposed classifier maintained a promising TPR. TNR is almost comparable for both the classifiers. Fig. 8 compares the overall accuracy for both the classifiers. Accuracy for proposed classifier remains high for all the scenarios. It is clear that the proposed classifier outperforms the existing classifier in terms of TPR and accuracy (ACC).

ISSN: 2321-8169
168 – 174

Table 1 Performance comparison of Existing and Proposed Classifier

Original image Quality factor	Spliced image Quality factor	Existing approach (Luo et al., 2007)				Proposed Approach			
		TPR	TNR	ACC	AUC	TPR	TNR	ACC	AUC
QF100	QF60	95.4	98.6	97.0	0.9912	94.3	98.8	96.5	0.9971
	QF80	71.0	95.4	83.2	0.8518	79.6	94.6	85.7	0.9195
	QF100	71.0	96.0	80.8	0.8477	80.6	94.6	86.0	0.9067
QF80	QF60	86.7	98.5	92.4	0.9764	90.7	98.8	95.1	0.9796
	QF80	74.1	95.0	85.2	0.8634	78.2	95.8	87.4	0.9168
	QF100	70.3	97.1	83.8	0.8616	80.5	96.2	88.5	0.9324
QF60	QF60	66.1	96.3	82.2	0.8253	71.1	95.9	84.4	0.8905
	QF80	81.7	96.9	89.8	0.9414	83.4	97.3	90.8	0.9712
	QF100	86.5	97.5	92.6	0.9567	86.0	97.9	92.0	0.9639

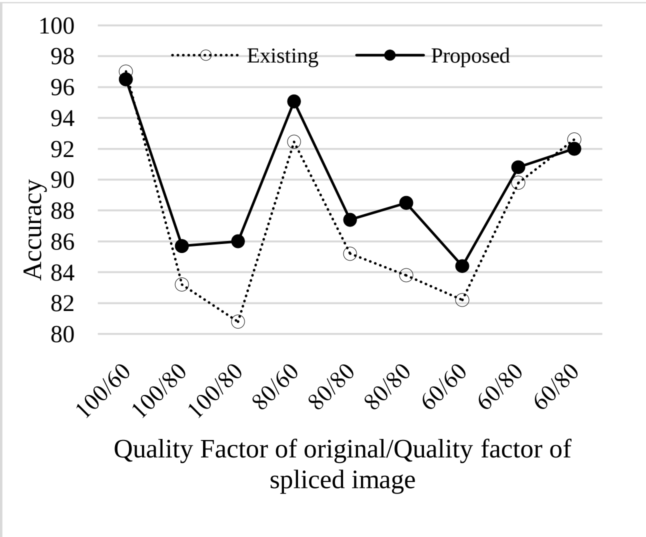
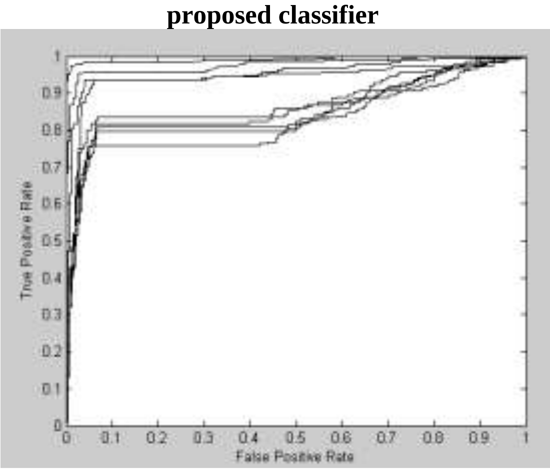
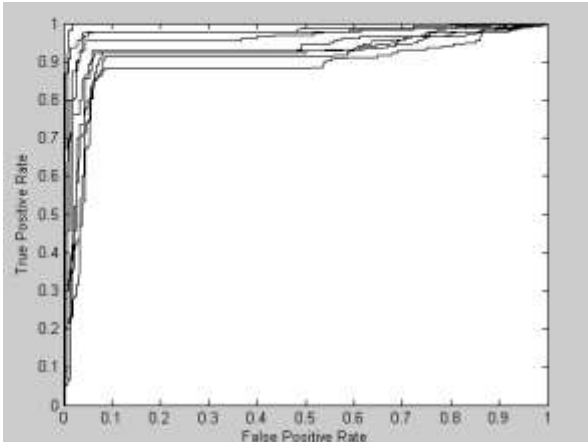


Fig. 8 Comparison of Accuracy for existing and proposed classifier



a)



b)

Fig. 9 Receiver Operating Characteristic Curve for a) existing and b) proposed classifier

A perfect classifier has AUC value equal to 1. Proposed classifier has AUC≥0.9 at all QFs. Moreover, the technique achieved improved results without addition to algorithm complexity.

Receiver Operating Characteristic Curve (ROC) for the proposed and existing classifier is shown in Fig. 9. It can be observed that the AUC values obtained for proposed approach are much higher than those obtained for existing classifier for all the scenarios.

B. Accuracy of Classification of Images with Small versus Large Spliced Area

The classifier performance is also evaluated in terms of small versus large splicing area shown in Table 2. It is observed that classifier performs better in classifying images with small (<=30%) spliced area as compared to images with large (30%-60%) spliced area as shown in graph in Fig. 10.

Table 2 Performance comparison of proposed Classifier for Small and large spliced area

S. No.	Quality factor	Small spliced area	Large spliced area
1	QF60	98.9	98.8

ISSN: 2321-8169
168 – 174

3	QF80	95.6	93.6
4	QF100	96.7	92.6

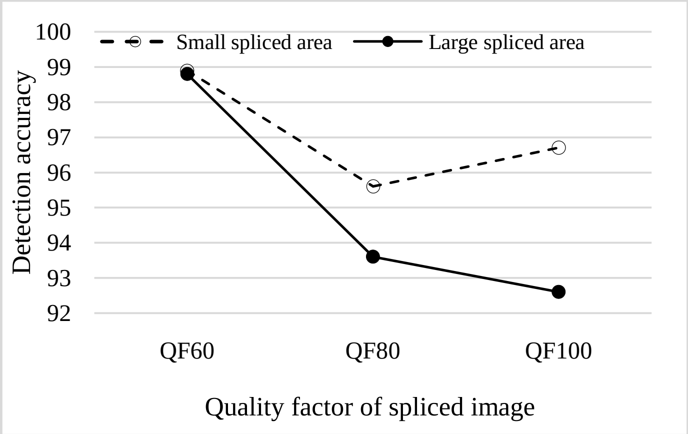


Fig. 10 Classifier accuracy for images with small and large spliced area

C. Impact of Noise on Classifier Accuracy

As images are very much prone to noise, it is obvious that the classifier results will vary in presence of noise. In this paper four different types of noise are considered i.e. fast fading, gaussian blur, white noise and JPEG. 320 images with different types of noise from LIVE2[22]database have been taken. These authentic images are checked for their true classification using the proposed classifier. The classifier classifies the images having Gaussian blur and white noise accurately. The accuracy obtained is 100%.For Fast fading and JPEG noise, the accuracy decreases to 85.7% and 84.8% respectively. For more comprehensive study, a number of original and spliced images with noise may be tested. But it needs another experimental setup and creation of new dataset by adding each type of noise to various types of spliced images which is out of the scope of this paper.

IV. Discussion and Conclusion

In this paper, machine learning based blind JPEG classifier for detecting spliced images has been proposed and implemented. The statistical differentiating features based on image i.e. Blocking Artifact Characteristics Matrix (BACM) have been extracted. Original images and spliced images at various quality factors i.e. QF60, QF80, QF100 have been considered to train and test LIBSVM based classifier. The main advantage of proposed classifier is that it performs well irrespective of the quality factor at which image is saved. It can be used to detect spliced images undergone through any kind of pre-processing operation as cropping, resampling, rotation etc. as well as any post-processing operation such as blurring. Moreover, the spliced area may be large or small. Additionally, it supports .bmp and .tiff images. The receiver operating characteristic curve and area under the curve demonstrated that proposed classifier performs better as compared to existing one. The only limitation is that classifier accuracy drops when both the original and spliced images are saved at poor QF60.The proposed classifier may be extended to make an integrated forensic tool which can detect splicing, copy move, seam carving, steganography and other types of tampering in images.

References:

[1] J. Lukas and J. Fridrich: ‘Estimation of primary quantization matrix in double compressed jpeg images’, *Proc. of DFRWS, Cleveland, USA*, August 2003, pp. 5-8.

[2] J. Lukas, J. Fridrich, and M. Goljan: ‘Detecting digital image forgeries using sensor pattern noise’, *Proc. of SPIE*, 2006, pp. 0Y1–0Y11.

[3] T.T. Ng and S.F. Chang: ‘Blind detection of digital photomontage using higher order statistics’, *IEEE International Symposium on Circuits and Systems (ISCAS 2004)*, Vancouver, Canada, Vol. 5, pp. 688–691.

[4] A.C. Popescu and H. Farid: ‘Statistical tools for digital forensics’, *International Workshop on Information Hiding, Springer Berlin Heidelberg*, 2004, pp. 128-147.

[5] A.C. Popescu and H. Farid: ‘Exposing digital forgeries by detecting traces of re-sampling’, *IEEE Trans. on Signal Processing*, 2005a, Vol. 53, no. 2, pp. 758–767.

[6] A.C. Popescu and H. Farid: ‘Exposing digital forgeries in color filter array interpolated images’, *IEEE Trans. on Signal Processing*, 2005b, Vol. 53, no. 10, pp. 3948–3959.

[7] F. Pan, X. Lin, S. Rahardja, E. P. Ong and W. S. Lin : ‘Measuring blocking artifacts using edge direction information’, *In Multimedia and Expo, ICME 2004*, IEEE International Conference on, Vol. 2, pp. 1491-1494.

[8] C. Perra, F. Massidda and D. D. Giusto: ‘Image blockiness evaluation based on sobel operator’, *IEEE International Conference on Image Processing*, 2005, Vol.1 (part-1), pp. 389–392.

[9] Z. Fan and R. L. De Queiroz: ‘Identification of bitmap compression history: JPEG detection and quantizer estimation’, *IEEE Transactions on Image Processing*, 2003, Vol. 12, no. 2, pp. 230–235.

[10] W.Q. Luo, Z.H. Qu, J.W. Huang and G.P. Qiu: ‘A novel method for detecting cropped and recompressed image block’, *in IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2007, pp. II-217–II-220.

[11] Y. L. Chen and C. T. Hsu: ‘Image tampering detection by blocking periodicity analysis in JPEG compressed images’, *in Proc. of IEEE 10th Workshop on Multimedia Signal Processing*, Oct. 2008, pp. 803–808.

[12] X. Pan and S. Lyu: ‘Region duplication detection using image feature matching’,*IEEE Transactions on Information Forensics and Security*,Vol. 5, no. 4, 2010, pp. 857–867.

[13] M. Barni, A. Costanzo and L. Sabatini: ‘Identification of cut & paste tampering by means of double-JPEG detection and image segmentation’, *in Proc. of ISCAS*, 2010, pp. 1687–1690.

[14] T. Bianchi and A. Piva: ‘Image forgery localization via block-grained analysis of JPEG artifacts’, *in IEEE Trans. Information Forensics and Security*, Vol. 7, no. 3, 2012, pp. 1003–1017.

[15] V. L. L. Thing, Y. Chen and C. Cheh: ‘An Improved Double Compression Detection Method for JPEG Image Forensics’, *IEEE Int. Symp. Multimedi*a, Dec. 2012, pp. 290–297.

ISSN: 2321-8169
168 – 174

[16] D. Tralic, J. Petrovic and S. Grgic: ‘JPEG image tampering detection using blocking artifacts’, *19th International Conference on Systems, Signals and Image Processing (IWSSIP)*, IEEE, 2012, pp. 5-8.

[17] Mall, S. Shukla, S.K. Mitra and A.K. Roy: ‘Comprehensive image index and detection of tampering in a digital image’, *in Int. Conf. Informatics, Electronics & Vision (ICIEV)*, 2013, pp. 1–7.

[18] I.C. Chang, J.W. Yu, and C.C. Chang: ‘A forgery detection algorithm for exemplar-based inpainting images using multi-region relation’, *Image and Vision Computing*,2013, Vol. 31, no. 1, pp. 57–71.

[19] K. Wattanachote, T. K. Shih, W. L. Chang and H. H. Chang: ‘Tamper Detection of JPEG Image Due to Seam Modifications’, *IEEE Transactions on Information Forensics and Security*, 2015, Vol. 10, no. 12, pp. 2477-2491.

[20] C.W. Hsu, C.C. Chang, and C.J. Lin: ‘A practical guide to support vector classification’, 2003, pp.1-16, http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

[21] J. Dong and W. Wang: ‘CASIA tampered image detection evaluation database’, 2011, http://forensics.idealtest.org.

[22] H.R. Sheikh, Z. Wang, L. Cormack and A.C. Bovik, ‘LIVE Image Quality Assessment Database Release 2’, 2005, http://live.ece.utexas.edu/research/quality.