Image Overlay Text Detection Based on JPEG Truncation Error Analysis

Dinesh Bhardwaj, *Student Member, IEEE*, and Vinod Pankajakshan, *Member, IEEE*

***Abstract*—This letter proposes a new algorithm for the detection and localization of overlay text in still images. The algorithm is based on the fact that the high-contrast edges in overlay text boundaries generate truncation error when subjected to JPEG compression. The regions containing high-contrast edges in a given test image are first identified using a discrete cosine transform (DCT) domain technique and a binary truncation error map is generated. The overlay text regions in the truncation error map are generally clustered and the cluster corresponding to a text line appears in a connected form. The initial detection is refined using connected component-based processing. Experimental results on a set of 300 images taken from news videos show that the proposed method is effective in detection of overlay text with a good accuracy.**

***Index Terms*—Image forensics, text detection, truncation error (TE).**

I. Introduction

EXTRACTION of textual information from images and videos is useful in many applications like indexing and retrieval of multimedia documents [1]. There are two types of text present in images and videos, namely *scene text* and *overlay text*. The scene text is naturally embedded in a captured image, for example, sign boards and number plates of vehicles. On the other hand, the overlay text, also known as the *caption text* is embedded in the image after it is captured. The detection of scene text is more challenging as compared to overlay text detection due to varying text orientations and contrast. Many text detection algorithms based on the features like shape, texture, edge, gradient, corner, stroke width, and color have been proposed in the literature [2]–[5]. Some of the recent works separate the detected text into overlay text and scene text. Such a separation improves the accuracy of the subsequent text recognition stage [6]. This letter investigates the detection of overlay text in still images.

Image forensics has been an active area of research over the past decade with many applications like forgery detection, verification of integrity, authenticity of digital images, etc. Image forensic techniques exploit the traces left by various signal processing operations and intrinsic fingerprints of capturing devices [7]–[9]. In a forensic perceptive, the embedding of overlay text can be considered as a tampering of the original image. Moreover, the overlay text regions in an image neither carry any trace of the artifacts introduced due to in-camera signal processing operations nor any fingerprint of the capturing device. If we consider the processing pipeline, an image is captured by a digital camera and is stored in a compressed format. The image is first decompressed, overlay text is inserted and then the resulting image is stored in compressed format. Hence, the overlay text regions of the image are single compressed whereas the remaining regions are double compressed. A forensic approach in overlay text detection is proposed in [10]. The method is based on the artifacts introduced by color filter array (CFA) interpolation used in digital cameras. The overlay text regions do not carry any trace of CFA interpolation since the text insertion is a postcamera operation. The method is shown to be effective in the detection of overlay text from both uncompressed and high-quality JPEG compressed images.

This letter proposes a new forensic approach to detect horizontally oriented overlay text in still images. For better readability, overlay text is generally embedded in such a way that there is high contrast between the text and the background. When an image with embedded overlay text undergoes JPEG compression, truncation error (TE) is introduced near high-contrast edges in the text boundaries. It has been shown in the image forensic literature[11], [12] that the number of pixels with JPEG TE is negligible in natural images. We propose to use the identification of pixels with JPEG TE as a method for the detection of overlay text. A discrete cosine transform (DCT)-domain technique is proposed to detect TE in a given image. High-contrast edges in nontext areas may also cause TE. Such areas are removed using a series of postprocessing operations based on geometrical properties of overlay text.

The rest of this letter is organized as follows. Section II briefly discusses the major steps involved in JPEG encoding and decoding. In Section III, the proposed text detection algorithm is described in detail. Section IV presents the experimental results and performance analysis, and finally Section V concludes the letter.

II. JPEG Encoding and Decoding

Consider an 8-bit grayscale image. The image is first divided into nonoverlapping blocks of size 8×8 pixels and these blocks are further processed independently. Let $X(i,j)$, $i,j \in \{0,\ldots,7\}$, be an 8×8 pixel block of the image. The block is transformed into two-dimensional DCT (2-D DCT) domain and each DCT coefficient $Y(i,j)$ is quantized with a quantization step size $Q(i,j)$ to get quantized DCT coefficient

Manuscript received April 14, 2016; revised May 29, 2016; accepted June 08, 2016. Date of publication June 15, 2016; date of current version June 24, 2016. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Yong Xiang.

The authors are with the Department of Electronics & Communication Engineering, Indian Institute of Technology Roorkee, Uttarakhand 247667, India (e-mail: dinesdec@iitr.ac.in; vinodfec@iitr.ac.in).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LSP.2016.2581311

$Z(i,j)$, given by

$$Z(i,j) = \left[\frac{Y(i,j)}{Q(i,j)}\right], \qquad i,j \in \{0,\ldots,7\} \tag{1}$$

where $[\cdot]$ denotes the integer rounding operation. The quantization step size depends on the quality factor (QF) of JPEG compression. Finally, the quantized DCT coefficients are entropy encoded to get the JPEG bitstream. A JPEG encoded image can be converted back to the spatial domain by performing the inverse of the operations involved in the encoding process. The bit stream is entropy decoded and dequantized to get the DCT coefficients as

$$Y'(i,j) = Z(i,j) \cdot Q(i,j), \qquad i,j \in \{0,\ldots,7\}. \tag{2}$$

Note that the value of a dequantized DCT coefficient $Y'(i,j)$ is an integer multiple of the corresponding quantization step size $Q(i,j)$. The dequantized coefficients are then converted back to the spatial domain by applying inverse DCT on each 8×8 block. However, due to the error introduced by quantization, the resulting pixel values may not be integers in the range $[0, 255]$. The pixel values are rounded and truncated to integers in the range $[0, 255]$ to obtain the decompressed image block. The errors introduced by these operations are referred to as *rounding error* and *TE*, respectively. In JPEG encoding of a color image, the image is first converted to $YCbCr$ format and the chrominance planes (Cb and Cr) are subsampled by a factor of two. All the three planes are then processed independently with the same encoding steps used in the JPEG encoding of a grayscale image.

III. Proposed Algorithm

The proposed text detection algorithm comprises of two stages. The first stage is a coarse detection of probable overlay text regions in a given test image. The initial detection is refined in the second stage by removing nontext areas and by tightening the text block boundaries.

A. Text-Block Detection

Consider the luminance plane (Y-plane in the YC_bC_r format) of a color image obtained by decompressing a JPEG compressed image. Let $\boldsymbol{Y_1}$ denotes an 8×8 DCT block of the luminance plane and $\boldsymbol{Q}$ be the corresponding quantization matrix. The values of $\boldsymbol{Y_1}$ depend on the rounding and TEs introduced during JPEG decompression. In the absence of rounding and TEs, each DCT coefficient $Y_1(i,j)$ is an integer multiple of the corresponding quantization step size $Q(i,j)$. On the other hand, if there is rounding or truncation, $Y_1(i,j)$ is not an integer multiple of $Q(i,j)$. The effect of rounding error is such that the value of $|Y_1(i,j)|$, with a high probability, lies in the range

$$|Y_1(i,j)| \in (n \cdot Q(i,j) - 1,\ \ n \cdot Q(i,j) + 1) \tag{3}$$

for some positive integer n [11]. Hence, a value of $|Y_1(i,j)|$ outside this range, i.e.,

$$|Y_1(i,j)| \notin (n \cdot Q(i,j) - 1,\ n \cdot Q(i,j) + 1) \tag{4}$$

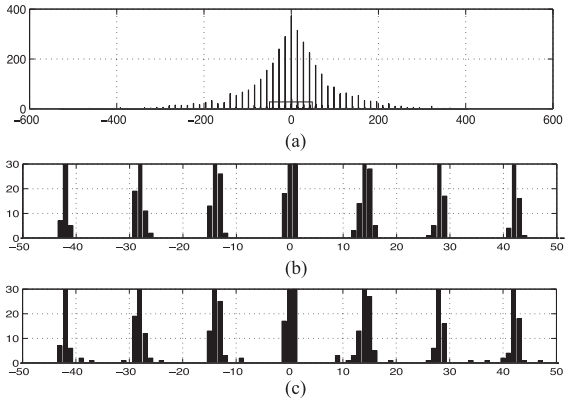

Fig. 1. Histogram of $Y_1(0,1)$. (a) Image without text. (b) Zoomed-in view of the rectangular area shown in (a). (c) Corresponding zoomed-in view for image with text.

for any positive integer n indicates TE in the corresponding 8×8 pixel block.

The effect of TE on the distribution of DCT coefficients is demonstrated with the help of the histograms shown in Fig. 1. A JPEG compressed image with QF $= 90$ is decompressed and two different images are generated from the decompressed image. The first image is obtained by recompressing the image with QF $= 40$, whereas the second image is generated by inserting overlay text and then recompressing the image with QF $= 40$. Both the images are decompressed and the histograms of $Y_1(0,1)$, i.e., $(0,1)$th DCT coefficients taken from each 8×8 block in the Y-plane of the decompressed image, are shown in Fig. 1. The histogram bins are centered at integer multiples of 14, the value of $Q(0,1)$ for JPEG compression with QF $= 40$. For the image without text, most of the DCT coefficients follow (3), except in few blocks which contain high-contrast edges. On the other hand, for the image with embedded text, there are many blocks having TE as indicated by the DCT coefficients that follow (4). Hence, the change in the distribution of the DCT coefficients of the decompressed image is an effective feature for detecting TE introduced by overlay text.

The TE in an 8×8 block can be detected from the values of the corresponding DCT coefficients. If any of the DCT coefficients follows (4), then that 8×8 block can be classified as having TE. However, such an approach may cause false detection of nontext areas since the TE may occur in smooth areas of the image when the pixel values are close to zero or near saturation (close to 255 for an 8-bit image). A DCT-domain technique is proposed to differentiate between the TE in smooth areas and that near high-contrast edges. The number of DCT coefficients that are resulted from truncation is more in the blocks containing high-contrast edges as compared to that in smooth areas. In each 8×8 DCT block of the luminance plane of the decompressed image, the number of DCT coefficients that are resulted from TE is computed. If this number is greater than a threshold, then that 8×8 block is labeled as having TE. A binary TE map is then obtained as

$$M(i,j) = \begin{cases} 1, & \text{if TE in } (i,j)^{\text{th}} \text{ block} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

Fig. 2. Test image and the corresponding TE maps. (a) Test image. (b) TE map (without shifting). (c) TE map (with shifting).

where $i = 0, 1,..., r/8 - 1$ and $j = 0, 1,. . ., c/8 - 1$, and r and c are the number of rows and number of columns of the image, respectively. An empirically determined threshold value of 16 is used in the experiments.

The TE depends on the QF of JPEG compression: lower the QF, higher is the TE. If the test image is JPEG compressed with a high QF, it is recompressed with a low QF before the computation of TE map. Fig. 2(a) shows a test image (QF $= 80$) and Fig. 2(b) is the TE map obtained after recompressing the image with QF $= 40$. It is clear from the figures that the TE map provides a good initial detection of overlay text regions. The initial detection can be further improved by applying a simple shifting operation before recompression. In the proposed algorithm, the odd numbered rows of the test image are shifted right by one pixel and the odd numbered columns are shifted down by one pixel. The shifted image is then JPEG compressed and the TE map is computed. The shifting operation increases the number of high-contrast edges in the text regions. For instance, the shifting operation converts a straight line to a zigzag pattern. As a result, TE is introduced in more number of 8×8 blocks in the text regions. Fig. 2(c) depicts the TE map obtained after shifting and recompressing (QF $= 40$) the test image shown in Fig. 2(a). On comparing with the TE map shown in Fig. 2(b), it is evident that the shifting operation improves the initial detection of overlay text regions.

B. Nontext Area Removal and Text Boundary Refinement

The initial detection provided by the TE map is now processed to remove nontext areas and to refine text block boundaries. These postprocessing steps are based on the geometrical properties of horizontal text lines. Due to the detection in 8×8 block level, text regions in the TE map are mostly connected. However, there may be some unconnected areas, for instance in the regions of large font size text. Morphological dilation operation using a horizontal structuring element $[1\ 1\ 1]$ is applied to the TE map to improve the connectivity in the text regions. The resulting TE map is then resized using nearest-neighbor interpolation to get a binary text map $\boldsymbol{M_r}$, having the same size as the test image. Fig. 3(a) shows the overlay text regions of a test image and Fig. 3(b) is the corresponding binary text map. Many adjacent text lines are connected in the binary text map. They are separated using spatial-domain information. A horizontal gradient map having the same size as $\boldsymbol{M_r}$ is computed as

$$G(i,j) = \begin{cases} L(i,j) - L(i,j-1), & \text{if } M_r(i,j) = 1 \\ 0, & \text{if } M_r(i,j) = 0 \end{cases} \quad (6)$$

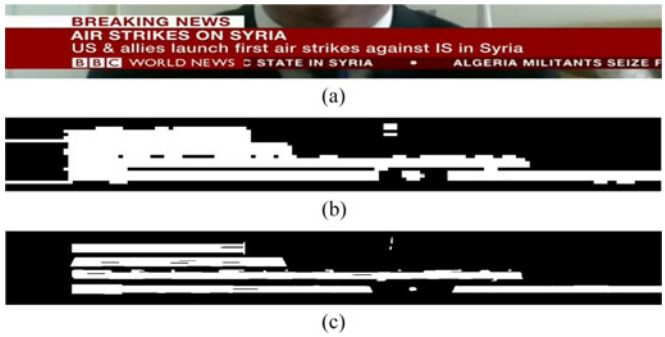

Fig. 3. Effect of postprocessing operations on TE map. (a) Test Image. (b) $M_r - map$. (c) $M_g - map$.

where $\boldsymbol{L}$ is the luminance plane of the test image, i is the row index, and j is the column index. It is expected that the overlay text boundaries result in large gradient values. The map is converted into a binary gradient map by applying a thresholding operation:

$$G(i,j) = \begin{cases} 1, & \text{if } |G(i,j)| > T_1 \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

where T_1 is the threshold. A modified binary text map $\boldsymbol{M_g}$ is then obtained by performing pixel-wise AND operation of the binary text map ($\boldsymbol{M_r}$) and the binary gradient map ($\boldsymbol{G}$). Since the binary gradient map contains only text boundaries, a horizontal dilation operation is applied to it before performing the AND operation. The rationale behind this dilation operation is to maintain the connectivity in the overlay text regions for subsequent processing. In each row of the binary gradient map, if the separation between two 1s is less than a threshold T_2, then they are considered to be a part of the same word and all the zeros between those 1s in the binary gradient map are filled with 1 s. Fig. 3(c) shows the binary map $\boldsymbol{M_g}$ ($T_1 = 35$ and $T_2 = 30$) for the test image shown in Fig. 3(a). It can be observed from the map that the individual text lines are now separated and the text block boundaries are refined. The final step in the postprocessing is the removal of nontext areas depending on the size and the aspect ratio of connected components. A connected component is considered as a text line if the number of 1s is above a threshold T_3 and the aspect ratio is above a threshold T_4. Any connected component that does not satisfy both these conditions is removed from the map $\boldsymbol{M_g}$. Finally, the remaining connected components are enclosed in rectangular bounding boxes and these bounding boxes represent the locations of detected overlay text.

TABLE I
COMPARATIVE PERFORMANCE OF THE PROPOSED TEXT DETECTION METHOD

	Set 1			Set 2		
Method	RR	PR	Time	RR	PR	Time
Proposed (QF=20)	86.7	87.2	0.8	82.6	85.9	2.1
Proposed (QF=40)	90.9	92.5	0.8	90.7	89.1	2.2
[15]	91.6	86.6	25.7	90.4	84.8	73.5
[3]	74.7	77.4	0.7	76.1	80.5	1.7
[16]	73.9	76.5	56.5	80.9	77.4	111.5

IV. EXPERIMENTAL RESULTS

A number of experiments were conducted to evaluate the performance of the proposed algorithm. The experiments were conducted in MATLAB 8.6 running on a workstation with Intel Xeon (E5) CPU at 2.4 GHz and 128 GB RAM under Microsoft Windows 8. An image database made of thumbnail images obtained from YouTube videos of popular news channels is used in the experiments. Note that the thumbnail of a video is an intracoded frame (I-frame), which is compressed similar to JPEG compression of a still image. Hence, the proposed algorithm is expected to be effective in detection of overlay text from such images. The database consists of images of two different resolutions, 1280×720 (set 1) and 1920×1080 (set 2) pixels. Each dataset consists of 150 images. Word-wise ground truth is manually generated for each image in the database and only the overlay text regions are included in the ground truth. The ICDAR 2013 evaluation framework [13], [14] with the *recall rate* (RR) and *precision rate* (PR) as the performance measures is used in the experimental analysis. The RR is the percentage of correctly detected text blocks out of all the text blocks in the ground truth. On the other hand, the PR is the percentage of correctly detected text blocks out of all the detected text blocks, is a measure of false detection.

For computing the TE map, each test image is subjected to row and column shift (Section III-A) and then recompressed. We considered different QFs for the recompression and the results for QF $= 40$ and QF $= 20$ are reported. The TE maps are processed using the steps described in Section III-B to get the detected text blocks for each test image. The threshold values $T_1 = 35, T_2 = 30, T_3 = 1000$ (for images in set 1), $T_3 = 1500$ (for images in set 2), and $T4 = 1.5$ are used for the postprocessing of TE map.

For comparative performance analysis, we have chosen the corner-based [3], wavelet-Laplacian-based [15], and link-energy-based [16] text detection methods. For a fair comparison, the same postprocessing steps as in the proposed method are applied to the initial text maps obtained using [15] and [3]. Since the images in our database are having larger dimensions than the test images considered in [15] and [3], we have used local window size $N = 31$ (for images in set 1) and $N = 51$ (for images in set 2) for obtaining maximum gradient difference values in [15] and dilated corner map in [3].

The performance of the four considered text detection algorithms on the datasets is summarized in Table I. The time given in the table is the average execution time (in seconds) per image. For all the four text detection methods, detected scene text, if any, is not considered while calculating the PR. It can be observed from the table that both the proposed and the Laplacian-based methods outperform the other two text detection methods. The performance of the proposed technique depends on the QF of JPEG recompression. The TE increases with the reduction in the QF and, hence, the initial text map is better with low-QF JPEG recompression. However, it has been observed that reducing the QF below a certain value reduces the performance of the proposed text detection technique. This may be due to the fact that severe JPEG compression removes the high-frequency DCT coefficients. The best results in our experiments are obtained with recompression QF 40. It can also be observed from the table that despite the low computational complexity of the proposed technique, both the proposed and the wavelet-Laplacian-based techniques have comparable performance in terms of RR. Furthermore, the PR of the proposed technique is better than the Laplacian-based technique.

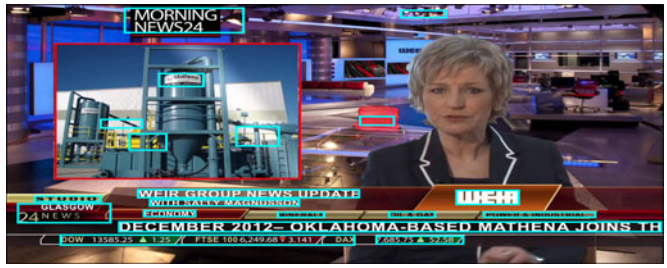

Fig. 4. Detection by proposed algorithm.

The detection performance of the proposed algorithm for a test image with complex background is shown in Fig. 4. The detected text blocks are enclosed in cyan-colored rectangles. In addition to the correct detection of overlay text blocks, some of the nontext areas are also detected. These false detections are due to the presence of high-contrast edges in the background. The main reason for missed detection by the proposed algorithm is that low-contrast overlay text boundaries may not result in JPEG TE. However, for better readability, the important text information is generally inserted in such a way that there is high contrast between the text and the background. Such text can be detected by the proposed algorithm. A limitation of the proposed method is that it detects only horizontally oriented text. This limitation is due to the postprocessing stage which uses geometrical properties of horizontal text.

V. CONCLUSION

This letter proposed a new technique for the detection of overlay text regions in still images. The main contribution of the study is the introduction of a new low-complexity, forensic-based method for overlay text detection. It is shown that the high-contrast edges in the overlay text boundaries introduce TE in JPEG recompressed image. A binary TE map is obtained from the test image and then by using geometrical properties of text lines, refined text block boundaries are calculated. The experimental results show better performance of the proposed technique as compared to the existing techniques.

REFERENCES

[1] K. Jung, K. I. Kim, and A. K. Jain, "Text information extraction in images and video: A survey," *Pattern Recog.*, vol. 37, no. 5, pp. 977–997, 2004.

[2] Q. Ye and D. Doermann, "Text detection and recognition in imagery: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 7, pp. 1480–1500, Jul. 2015.

[3] X. Zhao, K. H. Lin, Y. Fu, Y. Hu, Y. Liu, and T. S. Huang, "Text from corners: A novel approach to detect text and caption in videos," *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 790–799, Mar. 2011.

[4] L. Neumann and J. Matas, "Real-time scene text localization and recognition," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, Jun. 2012, pp. 3538–3545.

[5] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, Jun. 2010, pp. 2963–2970.

[6] P. Shivakumara, N. V. Kumar, D. S. Guru, and C. L. Tan, "Separation of graphics (superimposed) and scene text in video frames," in *Proc. 11th IAPR Int. Workshop Document Anal. Syst.*, Apr. 2014, pp. 344–348.

[7] J. Fridrich, "Digital image forensics," *IEEE Signal Process. Mag.*, vol. 26, no. 2, pp. 26–37, Mar. 2009.

[8] A. Piva, "An overview on image forensics," *ISRN Signal Process.*, vol. 2013, 2013, Art. no. 496 701.

[9] P. Ferrara, T. Bianchi, A. De Rosa, and A. Piva, "Image forgery localization via fine-grained analysis of CFA artifacts," *IEEE Trans. Inform. Forensics Security*, vol. 7, no. 5, pp. 1566–1577, Oct. 2012.

[10] A. Hooda, M. Kathuria, and V. Pankajakshan, "Application of forgery localization in overlay text detection," in *Proc. Indian Conf. Comput. Vision Graph. Image Process.*, 2014, pp. 36:1–36:7.

[11] W. Luo, J. Huang, and G. Qiu, "JPEG error analysis and its applications to digital image forensics," *IEEE Trans. Inform. Forensics Security*, vol. 5, no. 3, pp. 480–491, Sep. 2010.

[12] B. Li, T. T. Ng, X. Li, S. Tan, and J. Huang, "Revealing the trace of high-quality JPEG compression through quantization noise analysis," *IEEE Trans. Inform. Forensics Security*, vol. 10, no. 3, pp. 558–573, Mar. 2015.

[13] D. Karatzas *et al.*, "ICDAR 2013 robust reading competition," in *Proc. 12th Int. Conf. Document Anal. Recog.*, Aug. 2013, pp. 1484–1493.

[14] C. Wolf and J.-M. Jolion, "Object count/area graphs for the evaluation of object detection and segmentation algorithms," *Int. J. Document Anal. Recog.*, vol. 8, no. 4, pp. 280–296, Apr. 2006.

[15] P. Shivakumara, T. Q. Phan, and C. Tan, "New wavelet and color features for text detection in video," in *Proc. 20th Int. Conf. Pattern Recog.*, Aug. 2010, pp. 3996–3999.

[16] J. Zhang and R. Kasturi, "A novel text detection system based on character and link energies," *IEEE Trans. Image Process.*, vol. 23, no. 9, pp. 4187–4198, Sep. 2014.