Image Splicing Localization Using Superpixel Segmentation and Noise Level Estimation

Siqian Li, Weimin Wei, Xiuru Hua, Xueling Chu
College of Computer Science and Technology
Shanghai University of Electric Power
Shanghai, China

***Abstract*—With the development of computer and artificial intelligence technology, the authenticity of digital images has been seriously challenged. How to judge the authenticity of digital images has become an important research direction. At present, splicing is one of the most common image tampered methods. Different sources of images have different noise levels, and this paper proposes a scheme to locate the image splicing areas by detecting the inconsistency of the local noise level of the image. Firstly, the image to be detected is segmented into pixel blocks with similar features by using SLIC (Simple Linear Iterative Clustering) superpixel segmentation algorithm. Secondly, the noise level estimation method based on PCA (Principal Component Analysis) is used to calculate the local noise level of each image block. Finally, using three clustering algorithms to cluster the results of the estimated noise level, and the splicing areas of the image is located according to the clustering results. The results show that the proposed scheme in this paper can effectively locate the splicing areas and retain more edge information of the detected areas.**

***Index Terms*—image splicing localization; noise estimation; superpixel segmentation algorithm; clustering; image forensics**

I. INTRODUCTION

In recent years, with the development of network sharing platforms and digital cameras, digital images have become an important source of information. Because of its unique advantages, it is used in many fields, such as the court, medicine, journalism and military. But digital images face many new challenges, it is difficult to guarantee their authenticity. Criminals use the tampered image for fraud. Therefore, how to judge the authenticity of the image and how to locate the tampered area of the image have become an essential part of the digital images research field.

Digital image forensics is divided into initiative forensic and passive forensic [1]. According to the different embedded information, initiative forensic can be divided into digital signature [2], digital watermarking [3], and digital fingerprinting [4]. However, the limitation of these methods is that the information must be embedded in advance. The passive forensic technology can judge the authenticity and the source of images without any prior knowledge [5], [6]. Nowadays, passive forensic technology has achieved a lot of achievements. The mainstream image tampered methods include splicing, copy-move, double JPEG compression and fuzzy. Among them, splicing and copy-move are the most common tampered methods. Copy-move is homologous tampering by copying areas of an image and pasting it onto another disjoint area in the same image. Compared with copy-move, splicing is heterogeneous tampering, which is to crop and paste areas from different images [7]. This paper focuses on the location of the tampered areas of the splicing images. Fig.1 is an example of tampering by cropping and splicing two images on the website *Flickr.com*, where the left is the original image and the right is the tampered image.

In the process of imaging, digital images are influenced by camera components, camera settings, or environmental factors, which will introduce noise. Noise differences are typically introduced during the imaging process or caused by intentionally adding noise to hide the tampered trace of the image. For original images, there is little difference in noise among different areas. However, for tampered images, since different images have different noise levels, the inconsistency of local noise levels becomes convincing evidence for detecting image splicing.

Fig. 1. Example of image splicing

II. RELATED WORK

Image noise has been widely used in digital image forensics. Researchers have conducted noise research on the detection and localization of splicing image, and there have achieved many results.

In literature [8], the kurtosis-based algorithm is improved by optimizing the objective function of noise estimation. However, this method assumes that the noise variances of the different pixel in the original image are similar. It is not applicable to images with clear texture and smooth areas. In literature [9], the author uses the median-based noise method to estimate the noise variance of each block and then exposes the image splicing areas. The disadvantage of this method is that the threshold must be chosen carefully, otherwise, it is

978-1-7281-4852-6/19/$31.00 ©2019 IEEE
Authorized licensed use limited to: Indian Instt of Engg Science & Tech- SHIBPUR. Downloaded on August 21,2020 at 13:50:53 UTC from IEEE Xplore. Restrictions apply.

difficult to locate the tampered area. In literature [10], the author proposes a method to locate the tampered area through noise variance estimation of overlapping blocks. It estimates the noise variance by calculating the second and fourth moments of each image block. But the method requires knowing the kurtosis of the original signal. In literature [11], Zoran proposes a noise estimation method based on kurtosis, which utilizes the relationship between kurtosis and noise variance. It does not know the kurtosis value of the original image in advance. In literature [12], an image splicing detection method is proposed by further extending the noise variance estimation of literature [11] to local image blocks. In literature [13], the noise estimation problem is expressed as a closed-form optimization problem, and then it is extended to the local noise estimation method. In literature [14], the author uses the image segmentation strategy in literature [12] and noise estimation algorithm to locate tampered areas. The literature mentioned above have achieved excellent results, and they all use effective noise estimation algorithm as support.

The rest of this paper is as follows. The third section describes the superpixel segmentation algorithm, the noise level estimation algorithm, and the three clustering methods. The fourth section is the experimental results and analysis. Finally, conclude this paper.

III. IMAGE SPLICING LOCALIZATION SCHEME

A. Algorithm flow chart

An image is first segmented by SLIC(Simple Linear Iterative Clustering) algorithm. The local noise level of each block is estimated using a PCA (Principal Component Analysis) noise level estimation. Then the clustering algorithm is performed to categorize these blocks into two clusters according to their noise levels. The cluster that has fewer blocks is regarded as the splicing area. The steps are shown in Fig.2.

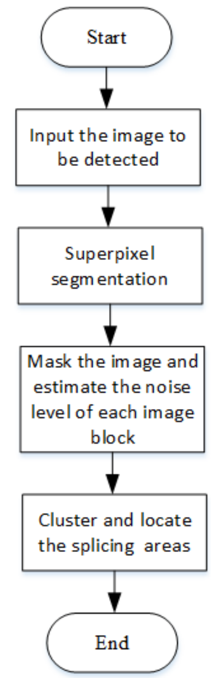

Fig. 2. Algorithm flow chart

B. Superpixel segmentation algorithm

1) Introduction: The traditional splicing image localization schemes are difficult to get accurate edge information. They divide the image into overlapping blocks or non-overlapping blocks, extract the features of image blocks and then locate the tampered areas according to the difference of the features. To better preserve the edge information of the location areas, this paper uses the superpixel segmentation algorithm to segment the image.

Superpixel refers to the small areas with adjacent positions and similar features in the image, which preserve the edge information very well. Superpixel segmentation divides the image into non-overlapping image blocks, reducing the input dimension while focusing on the study area. Superpixel segmentation algorithms are now widely used in many computer vision tasks such as semantic segmentation, visual tracking, image classification, and so on.

Superpixel segmentation algorithms are generally divided into two methods: graph method and gradient descent method. Literature [15] has comprehensively evaluated the 28 most advanced superpixel segmentation algorithms in many aspects. The experimental results show that the SLIC superpixel segmentation algorithm proposed by Achanta [16] in 2012 is one of the most effective segmentation methods at present. The algorithm can generate compact, nearly uniform superpixel with high speed. An example of the SLIC superpixel algorithm is shown in Fig.3.

The specific steps of the SLIC superpixel segmentation algorithm are as follows:

Step 1: Set the initial seed point. Set K seed points and distribute them evenly in the image.

Step 2: Adjust the position of the seeds. Within the $n \times n$ neighborhood of the seeds, move the center of the superpixel to the points where the gradient is the smallest of these points.

Step 3: Assign labels. For each superpixel center, within the range of $2n \times 2n$, if the distance from the point to the superpixel center is less than the distance from the original superpixel center, then it belongs to this superpixel center.

Step 4: Measure distance. For each pixel point searched, calculate the color distance and spatial distance between it and the seed point. Since each pixel point may be searched by multiple seed points, the seed point corresponding to the minimum value is the clustering center of the pixel point.

Step 5: Iterative optimization.

2) Parameter setting: SLIC superpixel segmentation algorithm needs to set parameter k, which is the number of clusters. Each image block is a cluster, and the center of the cluster is called superpixel. The size of k is inversely proportional to the size of the segmented image block. The larger k is, the smaller the segmented image block is, and the more the number of blocks is. Generally speaking, the size of segmented image blocks has an impact on the estimation of noise, and larger

Authorized licensed use limited to: Indian Instt of Engg Science & Tech- SHIBPUR. Downloaded on August 21,2020 at 13:50:53 UTC from IEEE Xplore. Restrictions apply.

Fig. 3. Example of SLIC algorithm

image blocks tend to be more reliable. In this paper, image blocks of different sizes are generated according to different k values to evaluate their noise estimation performance.

In order to evaluate the noise estimation performance of image blocks with different sizes, this paper uses TID2013 [17] database to test. The dataset contains 25 authentic images. We first add zero-mean white noise with a standard deviation of 10 to the images, and then divide the images into different blocks according to different k values. For different blocks of different k values, the standard deviation of noise estimation values of blocks is calculated for evaluating. In this paper, k values are 10,20,30,... ,100 and the results are shown in Fig.4

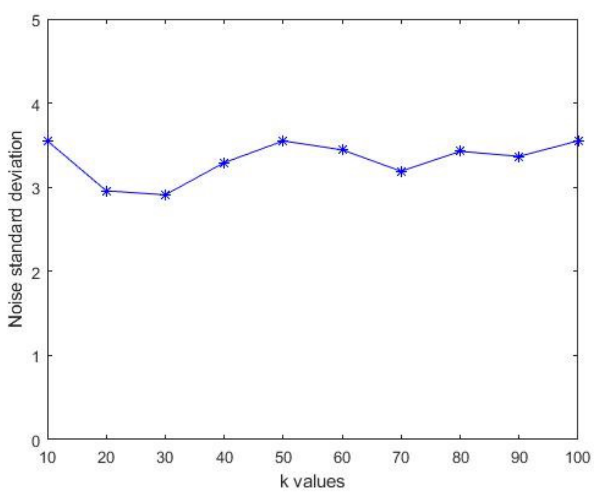

Fig. 4. Evaluation results of k value

The line chart is the error of noise estimation. It can be seen that the larger block has better noise estimation performance in general, but the larger block cannot obtain more accurate edge information. Taking these two aspects into consideration, the final value of k is 70.

C. Noise level estimation algorithm based on PCA

1) Algorithm principle: Noise will be introduced into the digital image during imaging or post-processing operations, and it is usually evenly distributed on the whole image. The noise level of different areas in the splicing image is often different, and the inconsistency of noise level can be used to detect whether the image was tampered.

The noise estimation algorithm proposed in the literature [18] is adopted to estimate the local noise level of the image blocks. This algorithm is one of the current excellent noise estimation algorithms. It is almost unaffected by the texture, and the estimation accuracy is high.

At present, the most widely used noise model is the additive white Gaussian noise model. The assumption is as follow:

$$y = x + n \tag{1}$$

Where y is the noise image, n is the zero-mean white Gaussian noise with a variance of σ^2 , and x is the original image.

First, suppose x is the original noiseless image of size $S_1 \times S_2$, where S_1 is the number of columns and S_2 is the number of rows. y is the image generated by the additive white Gaussian noise independent of the signal. Each x, n, y image contains $N = (S_1 - M_1 + 1)(S_2 - M_2 + 1)$ blocks of size $M_1 \times M_2$, the upper left corner of which is taken from the set $\{1, ..., S_1 - M_1 + 1\} \times \{1, ..., S_2 - M_2 + 1\}$. These blocks can be rearranged into vectors with $M = M_1 M_2$ elements and are treated as $x_i, n_i, y_i, i = 1, ..., N$ implementations of random vectors X, N and Y respectively.Since n is a signal-independent additive white Gaussian noise, $N \sim \mathcal{N}_M(0, \sigma^2 I)$ and $cov(X, N) = 0$.

Let S_X, S_Y be the covariance matrix of X and Y, $\widetilde{\lambda}_{X,1} \geq \widetilde{\lambda}_{X,2} \geq ... \geq \widetilde{\lambda}_{X,M}$ is the eigenvalue of S_X, whose corresponding normalized eigenvector is $\widetilde{V}_{X,1}, ..., \widetilde{V}_{X,M}$. $\widetilde{\lambda}_{Y,1} \geq \widetilde{\lambda}_{Y,2} \geq ... \geq \widetilde{\lambda}_{Y,M}$ is the eigenvalue of S_Y, and its corresponding normalized eigenvector is $\widetilde{V}_{Y,1}, ..., \widetilde{V}_{Y,M}$. $\widetilde{V}_{Y,1}^T Y, ..., \widetilde{V}_{Y,M}^T Y$ represents the sample principal component of Y, which satisfies $s^2(\widetilde{V}_{Y,k}^T Y) = \widetilde{\lambda}_{Y,k}$, where $k = 1, 2, ..., M$.

The author defines a class of noiseless images and satisfies the following assumptions: assuming m is a predefined positive integer, and the information in the noiseless image x is redundant, because all x_i are in the subspace $V_{M-m} \subset \mathbb{R}^M$, and its dimension $M - m$ is less than the number of coordinates M.

If the assumption is satisfied, the upper limit of $E(\left|\widetilde{\lambda}_{Y,i} - \sigma^2\right|)$ is asymptotic to $\sigma^2/\sqrt{N}$:

$$E(\left|\widetilde{\lambda}_{Y,i} - \sigma^2\right|) = O(\sigma^2)/\sqrt{N}, N \to \infty \tag{2}$$

where $i = M - m + 1, ..., M$.

When the assumptions satisfy the following formula:

$$\lim_{N \to \infty} E(\left|\widetilde{\lambda}_{Y,M} - \sigma^2\right|) = 0 \tag{3}$$

Authorized licensed use limited to: Indian Instt of Engg Science & Tech- SHIBPUR. Downloaded on August 21,2020 at 13:50:53 UTC from IEEE Xplore. Restrictions apply.

$\widetilde{\lambda}_{Y,M}$ converges to σ^2, so the noise variance can be estimated as $\widetilde{\lambda}_{Y,M}$. Since mean convergence means probability convergence, so the $\widetilde{\lambda}_{Y,M}$ is a consistent estimate of the noise variance.

The specific steps of the noise level estimation algorithm are as follows:

Step 1: Decompose y into overlapping small blocks of size $4 \times 4, 5 \times 5$, or 6×6.

Step 2: Calculate $\sigma_{ub}^2 = C_0 Q(p_0)$. σ_{ub}^2 is the upper limit of the true noise variance, and the true noise variance will not be higher than it.

Step 3: The subset of the image block is selected according to $Y_p = \left\{y_i | s^2(y_i) \leq Q(p), i = 1, ..., N\right\}$. By recursively discarding the block with the largest variance until the previous assumption is satisfied.

Step 4: Estimate the current noise level. Iterate step 3 and 4 until convergence.

Compared with the existing noise level estimation algorithms, the noise level estimation based on PCA has a good performance in accuracy and speed, so this paper selects this noise level estimation scheme for subsequence experiments.

2) Data preprocessing: For the segmented image, we use the mask to make the irregular image block into the same size as the image before the segmentation and then estimate the noise. In the process of the experiment, we found that some abnormal data often interfere with the subsequent clustering results, so we preprocess the estimated noise data. We use the logarithm function conversion $y = log(x, 2)$, where x is the noise data before processing, and y is the processed noise data. Fig.5 shows an example of data preprocessing. The left is original data, and the right is preprocessed data. It can be seen from the figure that there are large outliers in unprocessed data, and the processed data is smoother.

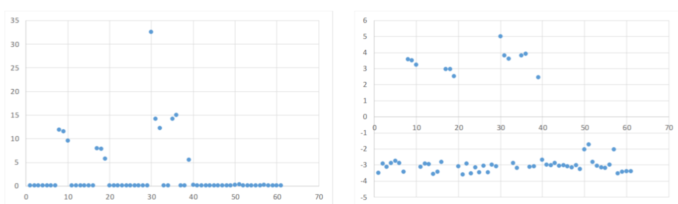

Fig. 5. Example of data preprocessing

D. Clustering algorithm

Clustering is the process of gathering data with the same characteristics. We use the clustering algorithm to locate the splicing areas in the image. In general, the cluster that has fewer blocks is regarded as the splicing areas. We use three clustering algorithms to cluster the results of noise estimation, and the best algorithm will be evaluated finally.

a) K-means clustering

The steps of the K-means algorithm are as follows: divide m points into n clusters, where each point belongs to the nearest clustering center. These points can be an observation or an instance of the sample. However, K-means has a disadvantage that it randomly selects k points as a clustering center in the dataset.

b) K-means++ clustering

The K-means ++ algorithm improves the K-means algorithm, it selects the clustering center according to the following steps: suppose that $n(0 < n < k)$ initial clustering centers have been selected, when the $n+1$ clustering center is selected, the point farther away from the current clustering centers will have a higher probability to be selected. However, K-means++ also use random methods when selecting the first clustering center($n = 1$).

c) DBSCAN clustering

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm is a clustering analysis algorithm proposed by Martin Ester. Given a set of points in one space, the algorithm can divide nearby points into a group and mark external points in the low-density areas. DBSCAN can find clusters of any shape in the spatial database with noise and connect adjacent areas with large density, which can effectively process abnormal data.

IV. EXPERIMENT RESULTS AND ANALYSIS

A. Database

In order to verify the effectiveness of the proposed scheme, Columbia University image database DVMM [19] was used for experiments. There are a total of 363 images in the data set. Among them, 183 images are real images, and 180 are spliced images. These images are taken from four digital cameras, including Canon, Nikon, Canon and Kodak. The image is in TIFF format and the size range is from 757 $\times$ 568 to 1152 $\times$ 768. In addition, they mainly contain interior scenes, such as desks, computers or hallways.

B. Experimental localization effect

We randomly select some images from the DVMM to locate the spliced area and the experimental effect is shown in Fig.6.

In the figure, column (a) is the image to be detected, column (b) is the localization effect using K-means, column (c) is the localization effect using K-means ++, and column (d) is the localization effect using DBSCAN. It can be seen from the figure that the three clustering methods can detect and locate the tampered areas well. The K-means scheme is in good order, but it omits some image blocks that may not have obvious noise differents. The DBSCAN scheme can almost completely cluster tampered blocks, but some irrelevant blocks will be introduced. Comprehensively speaking, the K-means ++ scheme has the best clustering effect. It can locate the tampered areas very well.

This paper uses TPR (true positive rate), FPR (false positive rate), and ACC (accuracy rate) to evaluate the three clustering schemes. The evaluation results are shown in TABLE 1.

$$TPR = \frac{TP}{TP + FN} \tag{4}$$

$$FPR = \frac{FP}{FP + TN} \tag{5}$$

Authorized licensed use limited to: Indian Instt of Engg Science & Tech- SHIBPUR. Downloaded on August 21,2020 at 13:50:53 UTC from IEEE Xplore. Restrictions apply.

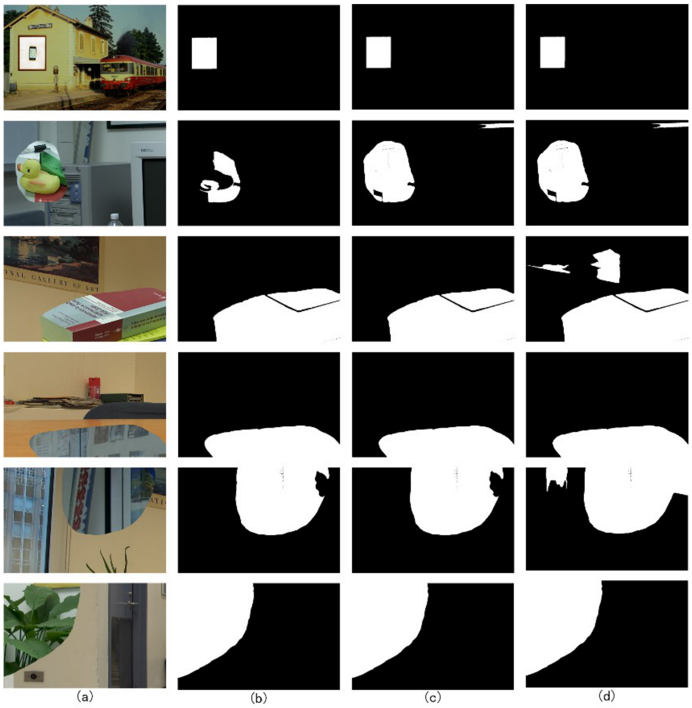

Fig. 6. Experimental localization effect

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

C. Comparison with existing schemes

We compare our scheme with the literature [12] and literature [14]. The experimental images used are from the literature [12], and the results of comparison are shown in Fig.7.

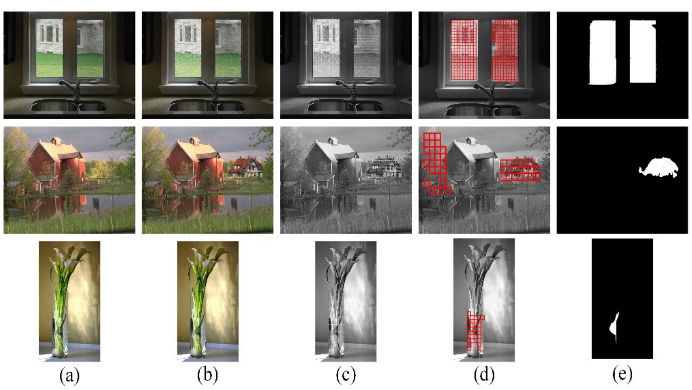

Fig. 7. Comparison with existing schemes

In the figure, column (a) is the original image, column (b) is the tampered image, column (c) is the localization effect by using the scheme in literature [12]. The column (d) is the localization effect by using the scheme in literature [14], and column (e) is the localization effect by using our scheme. The two windows of the first image are tampered with the added rain and snow. In the second image, the house on the right is the splicing area. In the third image, the vase is tampered

TABLE I
PERFORMANCE OF THREE SCHEMES

Images	Index	K-means	K-means++	DBSCAN
image 1	TPR	66.67%	66.67%	66.67%
	FPR	0	0	0
	ACC	97.87%	**97.87%**	97.87%
image 2	TPR	44.44%	88.89%	88.89%
	FPR	0	2.0%	2.0%
	ACC	91.53%	**96.61%**	96.61%
image 3	TPR	95.24%	95.24%	95.24%
	FPR	0	0	6.98%
	ACC	98.44%	**98.44%**	93.75%
image 4	TPR	100%	100%	100%
	FPR	0	0	0
	ACC	100%	**100%**	100%
image 5	TPR	90%	90%	100%
	FPR	0	0	2.63%
	ACC	96.55%	96.55%	**98.28%**
image 6	TPR	100%	100%	100%
	FPR	0	0	0
	ACC	100%	**100%**	100%

by adding grinding effect. It can be seen our scheme is more accurate and can reduce the wrong location. Meanwhile, this scheme can better preserve the edge of information of the splicing areas.

V. CONCLUSION

This paper proposes an effective scheme to locate the tampered areas of the splicing images. We use the difference of noise level to locate the splicing areas. For the case of abnormal data in the noise level estimation, we adopt the nonlinear normalization method to process the data. In the experiment, this paper evaluates three common clustering algorithms: K-means, K-mean++ and DBSCAN, among which K-means ++ and DBSCAN algorithm have better clustering effect. However, DBSCAN algorithm will have excessive clustering and assign some irrelevant image blocks to tampered areas. In summary, the K-means++ algorithm performs better.

The proposed scheme can better locate the splicing areas and preserve the edge information of the splicing areas more accurately than some previous schemes. In the future, we hope to strengthen the robustness of our scheme to deal with more complex splicing images.

Authorized licensed use limited to: Indian Instt of Engg Science & Tech- SHIBPUR. Downloaded on August 21,2020 at 13:50:53 UTC from IEEE Xplore. Restrictions apply.

ACKNOWLEDGEMENTS

This work was supported by Shanghai Natural Science Foundation (15ZR1418500), Shanghai Municipal Science and Technology Commission local colleges capacity building project (15110500700). We would like to express our gratitude to the teachers and students in the lab for their help in this paper.

REFERENCES

[1] A. Singh, N. Jindal, and K. Singh. A review on digital image forensics. In *International Conference on Signal Processing (ICSP 2016)*, pages 1–6, Nov 2016.

[2] Junling Zhang. A study on application of digital signature technology. In *2010 International Conference on Networking and Digital Society*, volume 1, pages 498–501, May 2010.

[3] G. Zhou and D. Lv. An overview of digital watermarking in image forensics. In *2011 Fourth International Joint Conference on Computational Sciences and Optimization*, pages 332–335, April 2011.

[4] MC Chang, DC Lou, and HK Tso. Combined watermarking and fingerprinting technologies for digital image copyright protection. *The Imaging Science Journal*, 55(1):3–12, 2007.

[5] M. C. Stamm, M. Wu, and K. J. R. Liu. Information forensics: An overview of the first decade. *IEEE Access*, 1:167–200, 2013.

[6] T. Qazi, K. Hayat, S. U. Khan, S. A. Madani, I. A. Khan, J. Kolodziej, H. Li, W. Lin, K. C. Yow, and C. Xu. Survey on blind image forgery detection. *IET Image Processing*, 7(7):660–670, October 2013.

[7] C. N. Bharti and P. Tandel. A survey of image forgery detection techniques. In *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pages 877–881, March 2016.

[8] X. Pan, X. Zhang, and S. Lyu. Exposing image splicing with inconsistent local noise variances. In *2012 IEEE International Conference on Computational Photography (ICCP)*, pages 1–10, April 2012.

[9] Babak Mahdian and Stanislav Saic. Using noise inconsistencies for blind image forensics. *Image and Vision Computing*, 27(10):1497 – 1503, 2009. Special Section: Computer Vision Methods for Ambient Intelligence.

[10] Alin C. Popescu and Hany Farid. Statistical tools for digital forensics. In Jessica Fridrich, editor, *Information Hiding*, pages 128–147, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[11] D. Zoran and Y. Weiss. Scale invariance and noise in natural images. In *2009 IEEE 12th International Conference on Computer Vision*, pages 2209–2216, Sep. 2009.

[12] Xunyu Pan, Xing Zhang, and Siwei Lyu. Exposing image forgery with blind noise estimation. In *Proceedings of the Thirteenth ACM Multimedia Workshop on Multimedia and Security*, MM&Sec '11, pages 15–20, New York, NY, USA, 2011. ACM.

[13] Siwei Lyu, Xunyu Pan, and Xing Zhang. Exposing region splicing forgeries with blind local noise estimation. *International Journal of Computer Vision*, 110(2):202–221, Nov 2014.

[14] Hui Zeng, Yifeng Zhan, Xiangui Kang, and Xiaodan Lin. Image splicing localization using pca-based noise level estimation. *Multimedia Tools and Applications*, 76(4):4783–4799, Feb 2017.

[15] David Stutz, Alexander Hermans, and Bastian Leibe. Superpixels: An evaluation of the state-of-the-art. *Computer Vision and Image Understanding*, 166:1–27, 2018.

[16] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Ssstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, Nov 2012.

[17] Nikolay Ponomarenko, Lina Jin, Oleg Ieremeiev, Vladimir Lukin, Karen Egiazarian, Jaakko Astola, Benoit Vozel, Kacem Chehdi, Marco Carli, Federica Battisti, and C.-C. Jay Kuo. Image database tid2013: Peculiarities, results and perspectives. *Signal Processing: Image Communication*, 30:57 – 77, 2015.

[18] S. Pyatykh, J. Hesser, and L. Zheng. Image noise level estimation by principal component analysis. *IEEE Transactions on Image Processing*, 22(2):687–699, Feb 2013.

[19] Y. Hsu and S. Chang. Detecting image splicing using geometry invariants and camera characteristics consistency. In *2006 IEEE International Conference on Multimedia and Expo*, pages 549–552, July 2006.

Authorized licensed use limited to: Indian Instt of Engg Science & Tech- SHIBPUR. Downloaded on August 21,2020 at 13:50:53 UTC from IEEE Xplore. Restrictions apply.