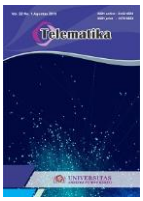

Available online at :
http://ejournal.amikompurwokerto.ac.id/index.php/telematika/

Telematika

Accreditated SINTA "2" Kemenristek/BRIN, No. 85/M/KPT/2020

Gaussian Pyramid Decomposition in Copy-Move Image Forgery Detection with SIFT and Zernike Moment Algorithms

Firstyani Imannisa Rahma[1*], Ema Utami[2], Hanif Al-Fatta[3]

[1.2, 3] Magister Teknik Informatika, Universitas Amikom Yogyakarta, Yogyakarta, Central Java, Indonesia
E-mail : firstyani.rahma@students.amikom.ac.id[1], ema.u@amikom.ac.id[2], hanif.a@amikom.ac.id[3]

ARTICLE INFO

History of the article:
Received July 16, 2021
Revised January 25, 2022
Accepted February 22, 2022

Keywords:
Copy-Move Forgery,
Scale Invariant Feature Transform,
Gaussian Pyramid Decomposition,
Zernike Moments

Correspondece:
E-mail:
firstyani.rahma@students.amikom.ac.id

ABSTRACT

One of the easiest manipulation methods is a copy-move forgery, which adds or hides objects in the images with copies of certain parts at the same pictures. The combination of SIFT and Zernike Moments is one of many methods that helping to detect textured and smooth regions. However, this combination is slowest than SIFT individually. On the other hand, Gaussian Pyramid Decomposition helps to reduce computation time. Because of this finding, we examine the impact of Gaussian Pyramid Decomposition in copy-move detection with SIFT and Zernike Moments combinations. We conducted detection test in plain copy-move, copy-move with rotation transformation, copy-move with JPEG compression, multiple copy-move, copy-move with reflection attack, and copy-move with image inpainting. We also examine the detections result with different values of gaussian pyramid limit and different area separation ratios. In detection with plain copy-move images, it generates low level of accuracy, precision and recall of 58.46%, 18.21% and 69.39%, respectively. The results are getting worse in for copy-move detection with reflection attack and copy-move with image inpainting. This weakness happened because this method has not been able to detect the position of the part of the image that is considered symmetrical and check whether the forged part uses samples from other parts of the image.

INTRODUCTION

Image manipulation for the criminal is still a dangerous impact of imaging technology growth. In Indonesia, many image manipulations cases are targeted at public figures and government officials to spread hoaxes and pornography-related content (Riadi, Fadlil, & Sari, 2017). One of the mildest image manipulation techniques is copy-move forgery. This manipulation allows a computer user to hide or add an object with the copied part on the same image (Tyagi, 2018).

There are two types of copy-move forgery detection. They are point-based and block-based (Sadeghi, Dadkhah, Jalab, Mazzola, & Uliyan, 2018). Block-based detection is done by sorting and looking for the similarity of the blocks resulting from splitting the input image before determining the damaged area, while the keypoint-based method is the process for generating a vector feature per keypoint which uses to find the similarity of objects in the damaged area.

Scale Invariant Feature Transform (SIFT) is one of many key point-based methods to detect copy-move forgery (Sadeghi et al., 2018). The advantages of SIFT are detection accuracy in rotation, noise adding, and JPEG compression (Hailing, Weiqiang, & Yu, 2008). Nuari, Utami, and Raharjo (2019) held

a quality comparison between SIFT and Speed Up Robust Feature (SURF) in copy-move detection. The comparison concludes that SIFT has a higher accuracy rate than SIFT when the maximum difference between those is 16.32%. Nevertheless, SURF detection is quicker than SIFT, when the maximum difference is 1.97 seconds.

However, SIFT method cannot identify forgery with smooth areas (Tan, Wu, Wu, & Chen, 2019). For dealing with this problem, some researchers combine SIFT with other methods. Lin et al. (2019) used SIFT with Local Intensity Order Pattern (LIOP) to help detect areas with few key points and significant geometrical transformations in copy-move detection. After extracting features, they did keypoint matching with transitive matching. The proposed method earned a 73.44% precision rate and 75.41%.

Besides LIOP, many researchers use Zernike Moments to deal with SIFT problem. This method can also be better in detecting copy-move pictures with rotation, blurring noise adding, and JPEG compression (Ryu, Lee, & Lee, 2010). Md Salleh, Rohani, and Maarof (2018) use Zernike Moments as feature extraction after using Discrete Wavelet Transform (DWT) in pre-processing steps. However, the noise shown in the results is due to a lack of morphological process (as seen in Figure 1). However, the detection time is faster than Zernike Moments without DWT, where this method can detect until 234 seconds.

Figure 1 From left to right : Original image, copy-move forgery result, detection result

Zheng et al. (2016) used the adaptive segmentation method for separating textured and smooth areas. After separating the region, they used SIFT with the g2NN matching method in the textured area and Zernike moments with overlapping blocks in the flat area. This method can acquire more than 88% precision and 86% recall. Nevertheless, this method reaches 179 seconds which means the running time is slower than SIFT.

Sun, Ni, and Zhao (2018) use nonoverlapping blocks segmentation to separate textured and smooth areas in copy-move detection for decreasing computing time. In the textured region, they used SIFT feature and g2NN keypoint matching. They use Zernike Moment feature extraction and hashing-based similarity calculation in flat areas. This method earns a maximum precision rate of 99.18% and a maximum recall rate of 89.57%. However, the detection time is still slower than SIFT, where this method needs 853 seconds to process the images.

Many researchers use a combination of SIFT and Zernike Moments to detect copy-move forgery (Mohamadian & Pouyan, 2013). However, this combination is slowest than SIFT only (Sun et al., 2018; Zheng et al., 2016). On the other hand, the Gaussian pyramid is helpful to detect copy-move forgery, including the fake image with noise and JPEG Compression, without burdening the testing time (Shabanian & Mashhadi, 2018).

Shabanian and Mashhadi (2018) use Gaussian pyramid decomposition in this research field. They use it to resize the forged image before dividing it into blocks. Then, each pair of blocks check by their similarity with Structural Similarity Index Measure (SSIM). This method yields a better result with

81.62% precision and 100% recall. This method is also stable in copy-move with JPEG compression and noise-adding attacks and helps to reduce the computational time.

This study intends to add SIFT and Zernike Moments combinations with the Gaussian pyramid decomposition method to detect copy-move image falsification. This study examines the average precision, recall, and detection time in using the Gaussian pyramid as decomposition and subsequently detected by a combination of SIFT and Zernike Moments methods.

RESEARCH METHODS

The proposed method process consisting Decomposition with gaussian Pyramid, separating smooth area and textured area, SIFT extraction in textured area, Zernike moments in smooth area and post-processing.

1. Gaussian Pyramid Decomposition

The first process is decomposition with Gaussian pyramid decomposition. This method will reduce image size by ½ times and smooth it with a Gaussian kernel (Shabanian & Mashhadi, 2018). Then, the images will be put to the layer above the previous image. This process repeats until forming an image pyramid.

However, when the image size is smaller, detection quality is lower. To control the image resolution before being used to the forward process, we use the method same as previous research (Rahma, Utami, & Fatta, 2020) to get the smaller image, as shown in Figure 2.

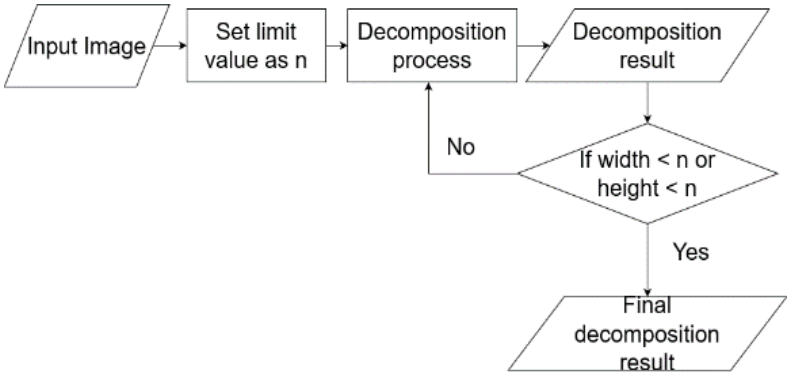

Figure 2 Controlling size after decomposition process

2. Separating Smooth and Textured Area

Before separating the image into the smooth and textured area, we will use the adaptive segmentation method (Pun, Yuan, & Bi, 2015) to get an initial size of each segment before determining the number of parts as mentioned in equation (1).

$$n = \sqrt{(H \times W)/s} \quad (1)$$

Where n is the number of segments, H and W each represent the height and width of the resized images, and s is the initial size of each part. This process begins by calculating the area of the picture by multiplying $H * W$. Next, the initial size (s) will divide the image's region area. Nevertheless, these results lead to too many segments that slow down the separation process. Therefore, we use the square root of the division result to get the n.

Next, we split up the image by adapting to the modified method from Zheng et al. (2016) First, we extract the keypoint with SIFT method and then make the segment with n value. However, due to

irregular shapes in each section, we counted the size and area of the bounding box, as exemplified in Figure 3, before we add the key point to it.

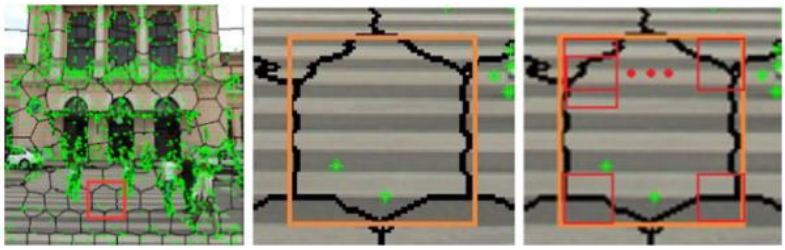

Figure 3 Zernike Moments detection from Zheng et al. (2016), where uses bounding box in each segment.

Then, make the ratio between the keypoints number and the bounding box's area. Then, we will compare it to the limit value. We put the segment to a smooth region if it is less or the same as the limit value. Otherwise, we put it into a textured section.

3. SIFT Detection

We will merge and convert the textured area into a mask as a guide area in SIFT extraction. SIFT looks for image features based "scale-space" concept to take features at more than one level of scale and type of image resolution, which not only increases the number of features available but is also tolerant of scale changes (Lowe, 1999; Lowe, 2004).

After that, we applied the g2NN feature matching method from Amerini, Ballan, Caldelli, Del Bimbo, and Serra (2011). First, we count the Euclidean distance between the key point to another. Next, we ordered the Euclidean data from the smallest. Then, we evaluate the ratio between the first distance (*d1*) and the following distance (*d2*) and compare it with the predetermined ratio value as shown in equation (2) for checking if the two pairs of points have a similarity. The last step is determining the pair of keypoint location points processed in the post-processing step.

$$\frac{d_1}{d_2} > 0{,}5 \tag{2}$$

4. Zernike Moment Detection

In the Zernike Moments detection process, we adopted and modified the Zernike moments process by Sun, Ni, and Zhao (2018). First, we merged the smooth area segment to form an image and split it into 24 × 24 pixels blocks. Next, we change each block from RGB to HSV mode and quantize H and S components to ten levels.

Then, we make *k* block groups based on the levels. We split the block groups into a main block and sub-blocks. Phase correlation is used to calibrate each sub-block. This calibration helps realign the location of the sub-blocks, where these sub-blocks are the area of forgery, as illustrated in Figure 4.

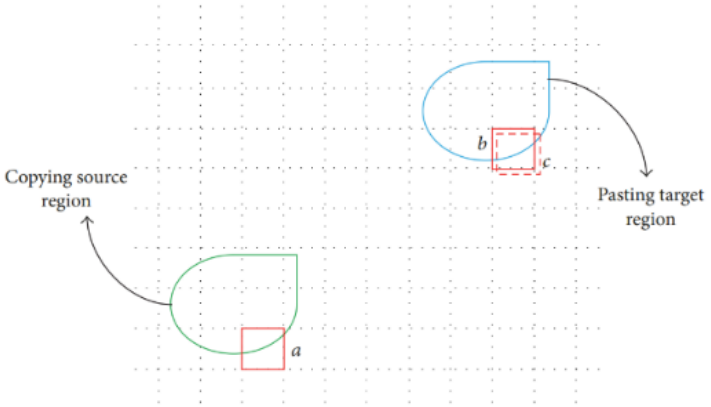

Figure 4 Alignment of main block (*a*), sub-block (*b*) and sub-block after calibration (*c*) (Sun et al., 2018)

The next step is counting Zernike Moments with V Component at each main block and sub-block. This method uses orthogonal functions called Zernike polynomials, which form a completely orthogonal set over the interior of the unit circle (Teague, 1980). Next, we determine the Euclidean distance between the Zernike Moments value of both areas. Later, we checked if the Euclidean distance is lesser than the predetermined value (0.05). The last step is finding center points from each pair of blocks. This result is also used in the post-processing step later.

5. Post-Processing

First, we joined SIFT keypoint locations and Zernike Moments center points into single-point location data. Then, we counted Euclidean distance between two coordinates of the position (indicated with (*i*,*j*) and (*k*,*l*)) and checked it with a predetermined value (*D*) as shown in equation (3).

$$\sqrt{(i-k)^2+(j-l)^2} < D \tag{3}$$

Next, we make a density-based cluster to make groups from selected positions (Soni, Das, & Thounaojam, 2018). The group is made based on the density of the distance between the main point and another. The last steps are to reduce the unnecessary location and the morphology process for filling the gap between the points.

6. Tools and Dataset

In this research, we use cloud-based Google Colaboratory. It has specifications equivalent to Intel Xeon 2.3 GHz, 13 GB RAM, and 108 GB space of Hard Disk. We also use Python 3.6.7 and image processing libraries like OpenCV library 3.4.2.17, Scikit-image library 0.16.2, and Mahotas library 1.4.11.

This research also used datasets from varying sources. First, using 96 images (48 plain copy-move forgeries and 48 ground truths) from Christlein et al. (2012). We also used copy-move with rotation (240 copy-move images and 288 ground truth) and copy-move with JPEG compression (240 copy-move images and 288 ground truth) (Christlein et al., 2012). All images in this dataset have a resolution between 533 × 800 pixels and 3888 × 2592 pixels.

We also use another dataset using 40 copy-move images and 40 ground truths, modified from the same dataset above (Christlein et al., 2012) taken from MICC F600 (Amerini et al., 2011) for detecting multiple copy-move. This image has a range with an equal resolution as above.

We modified 40 original images from MICC F2000 (Amerini et al., 2011) with GNU Image Manipulation Program (GIMP) for copy-move with reflection attack. All images in this dataset have the 2048 × 1536 pixels resolution. In copy-move with image inpainting, we changed 46 fresh pictures (Christlein et al., 2012).

7. Test Scenario

In the test scenario, we use plain copy-move and plain copy-move with rotation and JPEG Compression. With the criteria of transformations:

1. Rotation angle between 2° to 10° with 2° steps
2. JPEG compression with a quality factor between 20 to 100 in multiples of 10

Besides plain copy-move and copy-move with transformations, we also use other scenarios that related to process at this method:

1. Comparison between Gaussian pyramid decomposition size limit with three limit values in pixel 512, 768, 960.
2. Comparison between area separation ratio limit with three limit values in pixel 0,001, 0,003, 0,005.

Last, we add three scenarios contains others type of copy-move as mentioned by Walia and Kumar (2019) : multiple copy-move, copy-move with reflection attack and copy-move with image inpainting.

8. Analysis

In the analysis process, we will measure precision and recall rate at the pixel level using a confusion matrix, as shown in Figure 5. Then, we count accuracy, precision and recall rate as defined in equations (4), (5), (6).

		Predicted (detection map)	
		Positive (copy-move pixels)	Negative (Authentic pixels)
Actual (ground truth)	Positive (copy-move pixels)	TP	FN
	Negative (Authentic pixels)	FP	TN

Figure 5 Confusion Matrix table (Al-Qershi & Khoo, 2018)

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

$$Precision = \frac{TP}{TP+FP} \quad (5)$$

$$Recall = \frac{TP}{TP+FN} \quad (6)$$

Accuracy rate will measure the proportion between all true result and all result in confusion matrix (Al-Qershi & Khoo, 2018). Precision rate is a rate to measure if a detection result indicates the forgery on the image, when recall rate shows while recall shows the possibility that part of the faked image is detected (Christlein et al., 2012). All tests will be repeated three times to get an accurate result. Then we will count the average result of each test.

RESULTS AND DISCUSSION

1. Results in Plain Copy-Move

In plain copy-move, we used gaussian size limit 768 pixels, area separation ratio 0.003. The result, however, still generating false detection. Some unmanipulated parts are detected as copy-move, even though the detection reached the right forgery area. The resulting image is shown in Figure 6.

Due to many false results, there is an imbalance between precision and recall rate. The recall rate value is much greater than the value of precision. The average level of precision and recall in this experiment reached 18.21% and 69.39%, respectively. The average accuracy is also relatively low in 58.46%.

Figure 6 From left to right : Original image, plain copy-move image, ground truth, result map of the detection.

2. Results in Rotation Transformation

In copy-move with rotation attack detection, we use gaussian size limit 768 pixels and area separation ratio 0.003. Based on data in Table I, average precision and accuracy level are generally stable. However, there are still many undetected areas or the excess of the detection result. Nevertheless, the recall rate and processing time are rising with the increase in the value of the rotation angle. This result shows in Figure 7. The results of these data indicate that the higher the rotation value, the wider the detection ranges. The graph of accuracy, precision, recall and time is shown in Figure 8.

Figure 7 From left to right : Original image, copy-move with 10° rotation image, ground truth, result map of the detection.

Table 1 Results in Copy-Move with rotation transformation detection test

Angle (°)	*Avg. Accuracy (%)*	*Avg. Precision (%)*	*Avg. Recall (%)*	*Avg. Time (s)*
2	58.62	17.87	68.90	216.10
4	58.66	17.86	69.06	220.11
6	57.96	17.86	70.19	233.34
8	58.58	17.98	70.83	242.33
10	58.83	17.89	69.32	194.94

Figure 8 The graph form of results of rotation transformation detection test (a) average accuracy (b) average precision (c) average recall (d) average time

3. Results in JPEG Compression

We also use preset value for gaussian size limit and area separation ratio in copy-move with JPEG compression attack detection. The average precision and recall rate in this detection has fluctuated when they decreased in quality factor 40 to 70 but increased back to 80. The results data has served in Table 2.

This condition happened due to a large number of detections results along with a large number of pixels. The fluctuation happens because of the increase in the quality factor value. It also triggers false detection, which results in low precision values. The graph of accuracy, precision, recall and time is shown in Figure 9.

Table 2 Results in Copy-Move with JPEG Compression detection test

Quality Factor	*Avg. Accuracy (%)*	*Avg. Precision (%)*	*Avg. Recall (%)*	*Avg. Time (s)*
20	59.72	18.37	67.78	197.89
30	60.30	18.67	69.22	174.76
40	59.98	18.55	67.39	185.40
50	60.13	18.44	67.96	156.66
60	59.13	18.08	67.80	177.81
70	58.96	17.98	66.98	169.96
80	58.73	18.32	71.78	165.26
90	59.01	18.44	70.53	176.66
100	58.38	18.05	70.38	187.32

Figure 9 The graph form of results of JPEG compression detection test (a) average accuracy (b) average precision (c) average recall (d) average time

4. Results in Other Scenarios

a. Comparison between decomposition result size limit

In this experiment, we use three gaussian limits in pixel (512, 768, 960). Block size in Zernike Moments adjusts with image size. The result served in Table 3 and Figure 10.

Based on these data, when the limit value is rising, the precision rate increased, but the recall rate value is reduced. Image size reducing by Gaussian pyramid is a cause of this result. For the test time itself, the image processing time will be longer as the value limit for the image size and the size of each block increases.

Table 3 Results in decomposition size limit comparison test

Limit Value (pixel)	*Avg. Accuracy (%)*	*Avg. Precision (%)*	*Avg. Recall (%)*	*Avg. Time (s)*
512	54.48	17.36	72.00	69.25
768	58.46	18.21	69.39	186.93
960	58.21	18.32	69.38	219.90

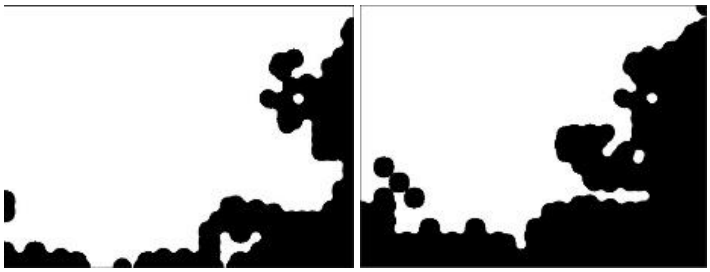

Figure 10 Left : Result with 768 pixels limit value; Left : Result with 960 pixels limit value

b. Comparison between area separation ratio limit

In this experiment, we used three separation ratio limit values: 0,001, 0,003, 0,005 and used Gaussian pyramid decomposition limit 768 pixels. The result served in Table 4 and Figure 11.

Based on these data, the right of separation ratio value is helping the quality of copy-move with a particular area. If the value is too small, some smooth regions are not detected because SIFT cannot produce a key point in this area. Otherwise, if the ratio is too large, there are many false detections, including smooth field.

Table 4 Results in area separation ratio limit comparison test

Separation Ratio	*Avg. Accuracy (%)*	*Avg. Precision (%)*	*Avg. Recall (%)*	*Avg. Time (s)*
0,001	58.71	18.64	71.71	250.21
0,003	58.46	18.21	69.39	186.93
0,005	53.52	16.75	70.89	178.86

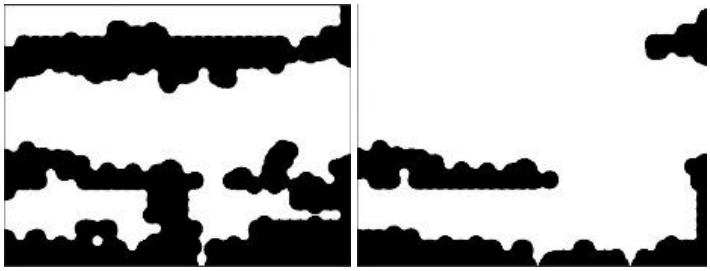

Figure 11 Result with 768 pixels limit value; Left : Result with 960 pixels limit value

5. Results in Other Copy-Move Types

a. Multiple Copy-Move

Figure 12 From left to right : Original image, multiple copy-move image, ground truth, result map of the detection.

In multiple copy-move, we used the same settings in the regular copy-move experiment. This experiment reached the highest result in all detection where the precision rate is 18.66%, recall rate is 71.73% and accuracy rate is 56.97%. This result shows proposed method can detect copy-move images with multiple duplications, but there are still many unwanted detections. The detection time is slightly slower, which earn 213 seconds. The resulting image is shown in Figure 12 above.

b. Copy-Move with Reflection Attack

Figure 13 From left to right : Original image, copy-move with reflection attack result, ground truth, result map of the detection.

In copy-move with reflection attack, we found the method is inadequate to detect this type, as the precision rate is 6.74% and recall rate is 22.40%, although the accuracy rate is 64.5%. The cause of the low level of precision and recall is matching process on SIFT and Zernike Moment step has not been able to check whether the keypoint or block detected by the copy-move has a symmetrical position with the original part (Warif, Wahab, Idris, Salleh, & Othman, 2017). The resulting image is shown in Figure 13 above.

c. Copy-Move with Image Inpainting

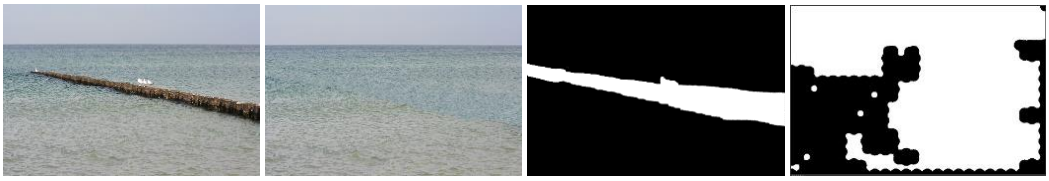

Figure 14 From left to right : Original image, copy-move with image inpainting result, ground truth, result map of the detection

Same with the previous type, we got the lowest rate in all detection where the precision rate value is 4,7%, and the recall rate is 41.72% with accuracy rate 59.58%. These results indicate that this method does not have a detection that checking whether original parts or undetected images that sampled to cover other objects in the picture (Liang, Yang, Ding, & Li, 2015). The resulting image is shown in Figure 14 above.

6. Discussion

We will explain how this method can only reach low precision and recall rates. First, we found that the Gaussian pyramid influences detection results. The image resolution decreases, the false detection increases, as shown in Figure 15.

Figure 15 Effect in Gaussian Pyramid in detection result in different size (in pixels); Left to right : 512 × 512, 256 × 256, 128 × 128 and 64 × 64 (Shabanian & Mashhadi, 2018)

Another factor that affects the result is the filtering of the Zernike Moments issue on smooth surface detection still relies on the similarity of Euclidean distances from one block to another. We also found the fault in objects that have smooth surfaces, such as the sky and ground areas.

Besides them, the clustering process in the post-processing process has not been able to produce accurate detection results where there are still many clusters formed from parts that are not detected. We also get the Google Colaboratory systems are also playing the detection role. We can conclude that the causes of this are lack of symmetrical position check on SIFT and Zernike Moment detection and sample checking in image inpainting, In response to low precision and recall rate at copy-move with reflection attack and image inpainting.

Our experiment has a similarity in computational complexity with one of our reference studies (Sun et al., 2018), where the computational complexity of the g2NN process at both experiments is O(n2), where n is the number of SIFT keypoints. However, the computational complexity in smooth area

detection at our test is O(k(m/k)3), where k is the number of color groups, while m is the total size of the smooth area divided by the block size.

CONCLUSIONS AND RECOMMENDATIONS

We tested the impact of the Gaussian pyramid at the combination of SIFT and Zernike Moments in copy-move forgery detection. All results have too few values, both in precision and recall rate. The cause of this weakness is the false detections that have gone too wide, although they reach the right faked area. Besides that, there are many undetected areas on some types of copy-move forgery.

After investigating this weakness, we found that the causes of low accuracy are the effect of the Gaussian Pyramid decomposition process, area filtering in Zernike Moments that is less deep, and clustering in post-processing that cannot produce accurate detection results. Besides them, the lack of additional features like a symmetrical position in the matching process and sample checking at uncopied parts or original parts. We hope the following research can overcome these problems.

REFERENCES

Al-Qershi, O. M., & Khoo, B. E. (2018). Evaluation of Copy-move Forgery Detection: Datasets and Evaluation Metrics. *Multimedia Tools and Applications*, *77*(24), 31807–31833. doi: 10.1007/s11042-018-6201-4

Amerini, I., Ballan, L., Caldelli, R., Del Bimbo, A., & Serra, G. (2011). A SIFT-based Forensic Method for Copy-move Attack Detection and Transformation Recovery. *IEEE Transactions on Information Forensics and Security*, *6*(3 PART 2), 1099–1110. doi: 10.1109/TIFS.2011.2129512

Christlein, V., Riess, C., Jordan, J., Riess, C., & Angelopoulou, E. (2012). An Evaluation of Popular Copy-move Forgery Detection Approaches. *IEEE Transactions on Information Forensics and Security*, *7*(6), 1841–1854. doi: 10.1109/TIFS.2012.2218597

Hailing, H., Weiqiang, G., & Yu, Z. (2008). Detection of Copy-move Forgery in Digital Images Using SIFT Algorithm. *Proceedings - 2008 Pacific-Asia Workshop on Computational Intelligence and Industrial Application, PACIIA 2008*, *2*, 272–276. doi: 10.1109/PACIIA.2008.240

Liang, Z., Yang, G., Ding, X., & Li, L. (2015). An Efficient Forgery Detection Algorithm for Object Removal by Exemplar-based Image Inpainting. *Journal of Visual Communication and Image Representation*, *30*, 75–85. Elsevier Inc. doi: 10.1007/s11042-017-4829-0

Lowe, D. G. (1999). Object Recognition from Local Scale-invariant Features. *Proceedings of the Seventh IEEE International Conference on Computer Vision* (Vol. 482, pp. 1150–1157 vol.2). IEEE. doi: 10.1109/ICCV.1999.790410

Lowe, D. G. (2004). Distinctive Image Features from Scale-invariant Keypoints. *International Journal of Computer Vision*, *60*(2), 91–110. doi: 10.1023/B:VISI.0000029664.99615.94

Mohamadian, Z., & Pouyan, A. A. (2013). Detection of Duplication Forgery in Digital Images in Uniform and Non-uniform Regions. *Proceedings - UKSim 15th International Conference on Computer Modelling and Simulation, UKSim 2013*, *1*, 455–460. doi: 10.1109/UKSim.2013.94

Pun, C. M., Yuan, X. C., & Bi, X. L. (2015). Image Forgery Detection Using Adaptive Oversegmentation and Feature Point Matching. *IEEE Transactions on Information Forensics and Security*, *10*(8), 1705–1716. doi: 10.1109/TIFS.2015.2423261

Rahma, F. I., Utami, E., & Fatta, H. Al. (2020). The Using of Gaussian Pyramid Decomposition, Compact Watershed Segmentation Masking and DBSCAN in Copy-Move Forgery Detection with SIFT. *2020 3rd International Conference on Information and Communications Technology (ICOIACT)* (pp. 325–330). IEEE. doi: 10.1109/ICOIACT50329.2020.9332081

Riadi, I., Fadlil, A., & Sari, T. (2017). Image Forensic for Detecting Splicing Image with Distance Function. *International Journal of Computer Applications*, *169*(5), 6–10. doi: 10.5120/ijca2017914729

Ryu, S.-J., Lee, M.-J., & Lee, H.-K. (2010). Detection of Copy-Rotate-Move Forgery Using Zernike Moments. *International Workshop on Information Hiding* (pp. 51–65). doi: 10.1007/978-3-642-16435-4_5

Sadeghi, S., Dadkhah, S., Jalab, H. A., Mazzola, G., & Uliyan, D. (2018). State of The Art in Passive Digital Image Forgery Detection: Copy-move Image Forgery. *Pattern Analysis and Applications*, *21*(2), 291–306. Springer London. doi: 10.1007/s10044-017-0678-8

Shabanian, H., & Mashhadi, F. (2018). A New Approach for Detecting Copy-move Forgery in Digital Images. *2017 IEEE Western New York Image and Signal Processing Workshop, WNYISPW 2017*, (November), 1–6. doi: 10.1109/ICCS.2008.4737205

Soni, B., Das, P. K., & Thounaojam, D. M. (2018). MultiCMFD: Fast and Efficient System for Multiple Copy-move Forgeries Detection in Image. *ACM International Conference Proceeding Series* (pp. 53–58). doi: 10.1145/3191442.3191465

Sun, Y., Ni, R., & Zhao, Y. (2018). Nonoverlapping Blocks Based Copy-Move Forgery Detection. *Security and Communication Networks*, *2018*. doi: 10.1155/2018/1301290

Tan, W., Wu, Y., Wu, P., & Chen, B. (2019). A Survey on Digital Image Copy-Move Forgery Localization Using Passive Techniques. *Journal of New Media*, *1*(1), 11–25. doi: 10.32604/jnm.2019.06219

Teague, M. R. (1980). Image Analysis Via The General Theory of Moments. *Journal of the Optical Society of America*, *70*(8), 920. Retrieved from https://www.osapublishing.org/abstract.cfm?URI=josa-70-8-920

Tyagi, V. (2018). *Understanding Digital Image Processing*. Boca Raton: CRC Press. doi: 10.1201/9781315123905

Walia, S., & Kumar, K. (2019). Digital Image Forgery Detection: a Systematic Scrutiny. *Australian Journal of Forensic Sciences*, *51*(5), 488–526. doi: 10.1080/00450618.2018.1424241

Warif, N. B. A., Wahab, A. W. A., Idris, M. Y. I., Salleh, R., & Othman, F. (2017). SIFT-Symmetry: A Robust Detection Method for Copy-move Forgery with Reflection Attack. *Journal of Visual Communication and Image Representation*, *46*, 219–232. doi: 10.1016/j.jvcir.2017.04.004

Zheng, J., Liu, Y., Ren, J., Zhu, T., Yan, Y., & Yang, H. (2016). Fusion of block and keypoints based approaches for effective copy-move image forgery detection. *Multidimensional Systems and Signal Processing*, *27*(4), 989–1005. doi: 10.1007/s11045-016-0416-1