Mehran University Research Journal of Engineering and Technology

https://doi.org/10.22581/muet1982.2202.14

2022, 41(2) 146-154

A novel approach of ceramic tile crack detection using morphological operations

Mahzaib Younas [a], Qamar Nawaz [a], Isma Hamid [b,*], Syed Mushhad Mustuzhar Gilani [c], M. Javed Iqbal [d].

[a] *Department of Computer Science, University of Agriculture, Faisalabad Pakistan*

[b] *Department of Computer Science, National Textile University, Faisalabad Pakistan*

[c] *University Institute of Information Technology, PMAS, Arid Agriculture University, Rawalpindi Pakistan*

[d] *Department of Software Engineering, University of Engineering and Technology Taxila, Pakistan*

[*] Corresponding author: Isma Hamid, Email: ismahamid@ntu.edu.pk

Received: 09 October 2020, Accepted: 05 May 2021, Published: 01 April 2022

KEYWORDS

Edge Detection

Quality Control

Morphological Operators

Defect Detection

Pattern Recognition

ABSTRACT

In ceramic tiles manufacturing industry, tiles are manufactured at large scale which makes it more challenging to ensure the quality of each tile according to the set standards. Mostly, Statistical Process Control (SPC) is used by tile manufacturers at each step to monitor various processing parameters. SPC procedures are implemented manually that requires sufficient number of experienced human resource to identify defected tiles from a batch of tiles. The manual inspection also gives low accuracy of defect detection due to human errors and hard environment. Considering these drawbacks, in this paper an automated defect detection method is proposed which is based on image processing and morphological operation to ensure the quality and standard of tiles. The proposed method resizes and converts the input RGB image into grayscale image and removes any possible noisy artifacts. An edge detection algorithm is applied on grayscale image to enhance the edges representing the cracks. Afterwards, morphological erosion and dilation operations are applied, one at a time, to get two intermediate images. Finally, edges are detected by subtracting eroded intermediate image from dilated intermediate image. For detection, the proposed algorithm does not require any separate reference image. The algorithm is tested on an image set of sixty different defected tile images and attained 92% average detection accuracy.

1. Introduction

Previously, in ceramic tiles manufacturing industry, Statical Process Control (SPC) [1] was a widely used method to monitor the build quality of tiles. The SPC is a manual procedure, and its successful implementation depends on the knowledge and training of the person deployed for quality control. The manual procedure for quality check is somewhat slow and results could be different regarding same object because of experience, level of concentration, and tiredness of the person [2].

For last couple of decades, the field of digital image processing and computer vision has turned into exceedingly onerous unit of information Technology in automating industrial processes. Digital images are processed by means of digital computer to extract important information, also called features, from images using simple to complex mathematical procedures [3]. Defect identification of tiles through digital images is widely used application of digital image processing to enforce computerized quality control process in manufacturing industry.

For defect identification in tiles images, researchers have proposed many different algorithms depending on the types, sizes, and textures of tiles [4]. Traditionally, defect detection algorithms perform resizing, noise removal, and edge detection procedures to identify manufacturing defects in tiles through digital images [5]. Based on the functionality, most of the defect detection algorithms can be viewed as a two-stage process i.e., low-level stage and high-level stage. At low-level stage, the algorithm extracts feature like colours, texture, edges, and patterns. At high-level stage, the algorithm utilizes low level information to extract semantic meanings to identify defects like corner break, cracks, and faded textures.

Another defect detection algorithm proposed by Elbehiery et al. [2], works in two stages. At first stage, the algorithm converts RGB image into grayscale to adjust the intensity of the image. Afterwards, intensity adjusted image is converted into binary image. At second stage, different morphological operations are performed to identify various types of defects like spot, pinhole, and long crack. This algorithm is not completely automated as user must select morphological operation according to the detection problem.

In [3], the authors proposed a gradient based algorithm to detect defect in Ceramic tiles. This algorithm performs scale-down operation to reduce image size and then converts it into grayscale. Edge detection is performed by using an edge detection operator [6] fixing the threshold value 0.13. In the next step, they applied a morphological operator on test image as well as on reference image. To detect the defects in the image, the algorithm counts white pixels of both of test image and compares their values with reference image. Main drawback of this algorithm is its huge processing time as it performs pixel by pixel comparison of test and reference image.

A fairly simple defect detection algorithm that is based on Laplacian of Gaussian operator is proposed by Rajashri et al. [4]. This algorithm converts input RGB image into a grayscale image and then applies Gaussian Filter to remove noise and smoothen the image then edges are detected using Laplacian of Gaussian Operator [7]. To identify cracks, this algorithm performs pixel by pixel comparison of edge detected image with a reference image of a perfect tile. This algorithm is simple to understand and easy to implement but, it can only be used to identify defects in a flat or single-color tile.

In [8], the authors presented a real time defect detection system to identify cracks based on Erosion operator [9]. The accuracy of the algorithm depends on threshold value as this value is not constant for different types of defects as well as images of plain and textured tiles.

Another technique for defect detection proposed in [10] is based on Prewitt algorithm [11]. This algorithm uses adaptive histogram equalization technique for contrast enhancement. Afterwards, noise is removed using median filter and edges are detected using Prewitt algorithm. Finally, defects are detected using morphological operation i.e. union, intersection, and inclusion. This algorithm only works for images of having size range between 600 x 600 and 800 x 800.

To address the main drawbacks discussed above, in this study, we propose an algorithm for defect detection that is easy to implement, requires less processing time and power, and is able to identify defects in plain and textured tile images without requiring any additional reference image.

The rest of the paper is organized as follows: In Section 2, details of the proposed algorithm are given. Section 3 contains results of the proposed algorithm and its comparison with some of the existing algorithm in the same domain. Section 4 summarizes the paper.

2. Proposed Methodology for Defect Detection System

The proposed algorithm is a three-step process. At first step, input image is converted into grayscale and pre-processing is performed to enhance the quality of the grayscale image. At second step, edge detection methodology is implemented on enhanced image. At third step, enhanced grayscale image produced at first step and edge detected image at second step are converted into binary and defects are detected after subtracting magnitude of both images. Complete procedure of the proposed algorithm is shown in Fig. 1. Details of each steps are given in the following sub-sections.

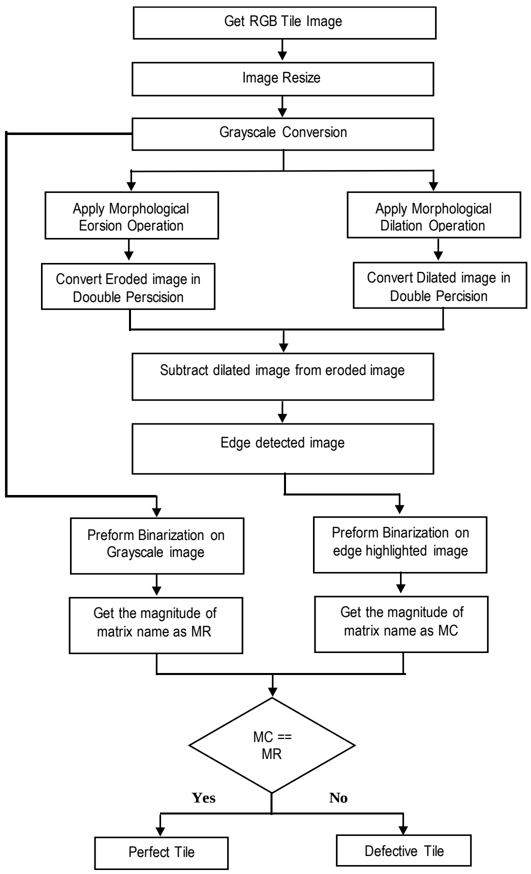

Fig. 1. Proposed Algorithm

2.1 Pre-Processing

Image pre-processing is an important step to remove any unwanted artifact from the input image. At this step, we resized input image it its size is greater than 500 x 500 to reduce the overall processing time of the algorithm. After resizing, we converted resized image into grayscale because colour features do not play any role in identification of defects like cracks, holes, and broken edges.

2.2 Edge Detection

The boundary between two dissimilar regions of an image is known as an edge. Usually, the boundaries form due to the different surfaces of objects or due to the light and shadow falling on a single surface [12]. In literature, different edge detection algorithms have been proposed which are being used to detect cracks in tiles. Some of the widely used methods are based on gradients that use predefine kernel for convolution, for example, Prewitt [13], Canny [14], Sobel [15], and Robert [16]. In our proposed algorithm, we used morphological dilation and erosion operations for highlight cracks and edges. These operations are natively robust for noise and are able to correct imperfections in structure by connecting any disjoint pixels [17]. These operation uses a small matrix structure, called structuring element [18]. The important properties of structuring element are its origin, shape, and size. Origin represents a pixel whose value is going to be updated after applying the operations, shape defines the pattern of 0 and 1 for pixels falling within a window, and size represents a window used to process number of neighbourhood pixels at a time. Structuring element can be used with different shapes such as line, disk, square, diamond, pair, and rectangle. In the proposed algorithm, we used disk shape structuring element with the radius 2 [19], as shown in Fig. 2.

$$\text{SE} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Fig. 2. Disk shape structuring element of size 2

After defining the structuring element, the algorithm performs dilatation and erosion operation on grayscale image. The dilation operator [18] is used to expand the foreground of image up to the extent to the maximum neighbourhood value in structuring element. The implementation of Dilation process is just like spatial convolution. Mathematical procedure of dilation operation is represented in Eqs. (1) and (2).

$$(Q \oplus SE)(i,j) = \max\{Q(i - i`, j - j`) \mid (i`, j`) \epsilon D_{SE}\} \quad \ldots(1)$$

$$d = Q \oplus SE \quad (2)$$

where SE is the structuring element, D_{SE} is domain of structuring element, Q is the enhanced grayscale image, $Q(i,j)$ is supposed the $-\infty$ outside the image domain, $(i`,j`)$ are domain of SE which belongs to D_{SE}, this symbol $\oplus$ represent the dilation operation, and d is the eroded image.

The Erosion Morphological operation is used to shrink the foreground of image [20]. In other words, erosion operation is the complement of dilation Process. The erosion operation also works like dilation operation, but it changes the value of origin pixel with minimum value found in neighbourhood window. Mathematical procedure of erosion is represented in Eqs. 3 and 4.

$$(Q \ominus SE)(i,j) = \min\{Q(i + i`, j + j`) \mid (i`, j`) \epsilon D_{SE}\} \quad \ldots(3)$$

$$e = Q \ominus SE \tag{4}$$

where SE is the structuring element, D_{SE} is domain of structuring element, Q is the enhanced grayscale image, $Q(i,j)$ is supposed the $+\infty$ outside the image domain, (i`,j`) are the domain of SE which belongs to D_{SE}, this symbol $\ominus$ represent the erosion operation, and e is the eroded image.

In Eq. 5, d is intermediate dilated image and e intermediate eroded image and N is the final edge highlighted image.

$$N = [double(d)] - [double(e)] \tag{5}$$

2.3 Crack Detection

For crack detection, the proposed algorithm binarization is perform on both grayscale image and edge detected image because of its ability of detecting the different type of defect, object separation, pattern recognition, and reduce the computation complexity. Afterwards, the algorithm calculates the magnitude of binary images individually. If magnitude of binary grayscale image equals to the binary edge detected image, then the tile is a perfect tile.

3. Results and Discussion

To test the performance of the proposed algorithm, we perform a series of experiments on dataset containing sixty images of different tiles. The dataset we used is not available publicly and so we collected images from multiple locations [21-22] because of the need of a dataset containing colour tile images. Some datasets are available publicly that contain pre-processed black & white images which are not suitable to test the proposed algorithm. The dataset used to test the algorithm is categorised in two categories, as shown in Table 1. Category 1 contains 29 perfect images i.e. images of faultless tiles and category 2 contains 31 images of defected tiles containing cracks, missing patterns, and surface scratches. Images in category 1 and category 2 are shown in Fig. 3 and Fig. 4 respectively.

Table 1

Categories of dataset used to test the algorithm

Category	Number of Tiles
Category 1 (Faultless Images)	29
Category 2 (Faulty Images)	31

Fig. 3. Category 1, faultless images in dataset

Fig. 4. Category 2, faulty images in dataset

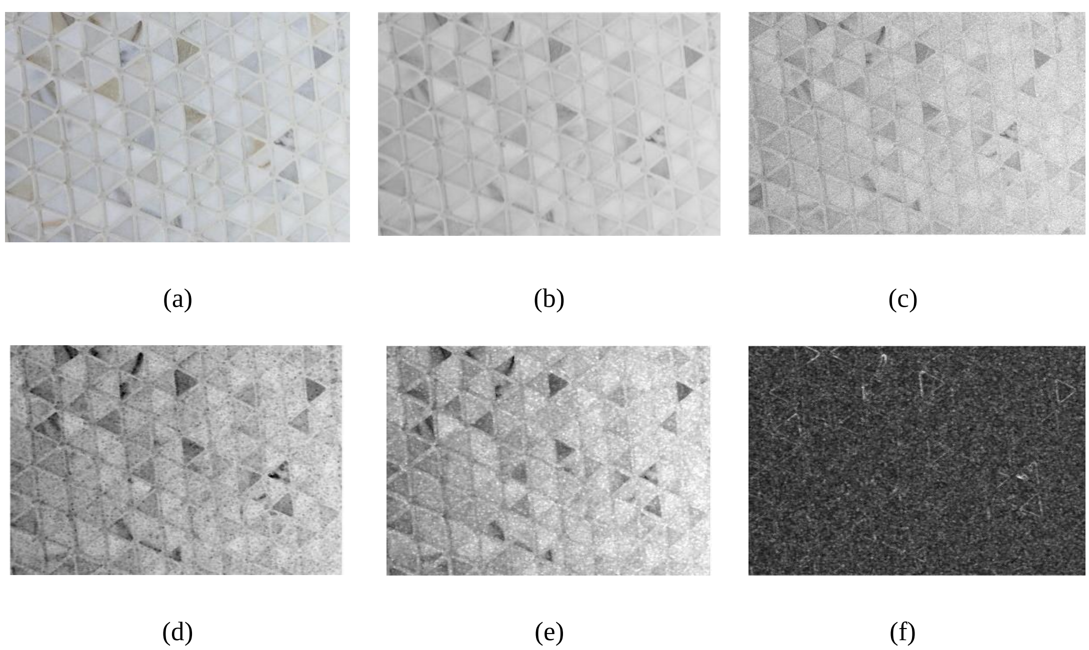

Fig. 5. Visual result of proposed algorithm applied on perfect tile image of category 1 (a) Input image (b) Pre-process image (c) Noisy image, (d) Eroded image, (e) Dilated image, (f) Final edge detected image

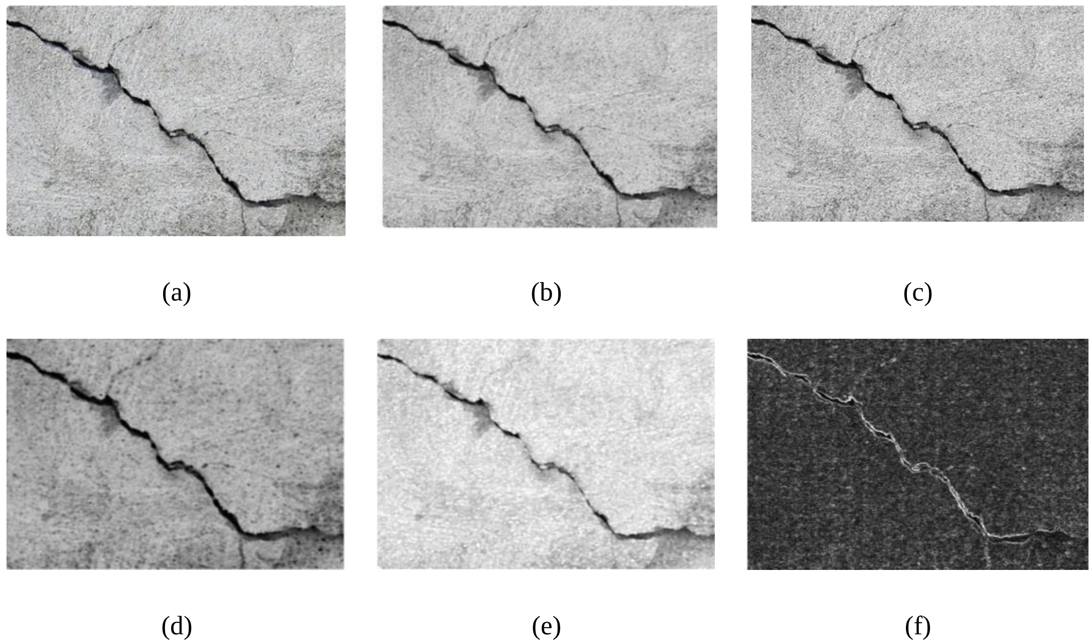

Fig. 6. Visual result of proposed algorithm applied on defective tile image of category 2 (a) Input image (b) Pre-process image (c) Noisy image, (d) Eroded image, (e) Dilated image (f) Final edge detected image

For the purpose of demonstration and to save the space, only one experimental detail for each category is presented in this section. The visual results of each stage produced by the proposed algorithm are shown in Fig. 5 and Fig. 6 for image no. 6 from category 1 and image no. 9 from category 2, respectively. The details of complete experimental outputs are given in Table 2.

Table 2

Details of experimental results of the proposed algorithm

Total number of tested tiles	60
Total number of perfect tiles	29
Total number of defected tiles	31
Number of perfect tiles detected as defected	2
Number of defective tiles detected as perfect	3
Total number of correct outcomes	55
Total number of wrong outcomes	5

From the results, it is found that the accuracy of the proposed algorithm is almost 92% and total processing time to process whole dataset is approximately 30sec. To find the accuracy of the proposed algorithm, following equation is used.

$$Accuracy\ Rate = \frac{Number\ of\ correct\ outcomes}{Total\ number\ of\ tile} \times 100 \quad \ldots(6)$$

3.1 Comparison with Existing Algorithm

We compared our proposed algorithm with six existing and widely used algorithms in the same domain. The algorithms we selected for comparison includes Prewitt [23], Sobel [15], canny [24], Roberts [15], Kirsch [25], and Robinson [25]. Fig. 7 represents visual results of edge detection after applying existing and proposed algorithm on tile 15 of category 2. It can be observed that the edge detected image produced by the proposed algorithm is clearer, sharper, and provides smooth and noiseless visual information that helps in clearly identifying cracks and tile texture. Normally, visual results of any defect detection algorithms are not important as humans are not involved in the process of identifying defects. The software, implementing the algorithm, is responsible of detecting and highlighting defects in underlying images. Edge detected image provides the base for detecting defects therefore, for the sake of comparison, we also compared the visual quality of the edge detected image of the proposed algorithm with existing algorithms. During the experiments, we observed that if defect detection algorithm produces good quality edge detected image, it ultimately provides better accuracy in defect detection stage. Visually, the proposed algorithm produces better edge detected image but, visual comparison is just one aspect. Sometimes, visual comparison alone is not enough as different persons may have different opinions regarding information available in the same image. Therefore, to validate our claim, we also compared objectively the edge detected images produced by the proposed algorithm and existing algorithms using three widely used image quality assessment matrices, i.e. RMSE [26], PSNR [27] and SSIM [28]. The mathematical equations and output results of RMSE, PSNR, and SSIM are given in Eqs. 7-10 and Table 3, respectively. Accuracy and processing time comparison of the proposed defect detection algorithm with existing defect detection algorithms are given in Table 4, respectively.

$$MSE = \frac{\sum[O(i,j)-e(i,j)]}{i*j} \quad (7)$$

$$RMSE = \sqrt{MSE} \quad (8)$$

where i and j are number of rows and columns of an image O is consider as an original image and e is edge detected which is also called the compressed image. And, root mean square error, *RMSE*, is actually the square root of the mean square error, *MSE*.

$$PSNR = 10log_{10}\left(\frac{R^2}{MSE}\right) \quad (9)$$

where R is maximal value of input image.

$$SSIM\ (i\ ,j) = [l(i,j)]^{\alpha}.[c(i,j)]^{\beta}.[s(i,j)]^{\gamma} \quad (10)$$

where l is luminous used to compare the brightness factor, c is contrast used to compare the brightness and darkness region, and s is structure used to compare the pattern. α, β and γ are the positive constants.

As shown in Fig. 11, the proposed edge detection algorithm is given the high accuracy rate. The Proposed Defect Detection process give the 92% accuracy which is an achievement of our proposed system.

Table 3

Comparison of Edge Detection Algorithm

Algorithm Name	RMSE	PSNR	SSIM
Prewitt	120.3038	6.73	0.0514
Sobel	133.6235	5.89	0.0334
Canny	129.3551	7.19	0.0382
Robert	123.2008	5.71	0.0366
Kirsch	89.1963	9	0.0111
Robinsion	114.9114	6.9	0.0430
Proposed Algorithm	88.97	8.89	0.213

Table 4

Comparison Using Existing Defect Detection Process

Algorithm	Correct Result	Incorrect result	Accuracy	Time
Prewitt	37	23	63%	27sec
Canny	44	16	74%	29sec
Sobel	38	22	65%	29sec
Roberts	41	19	70%	28sec
Kirsch	46	14	76%	33sec
Robinson	50	10	87%	35sec
Proposed Algorithm	55	5	92%	30sec

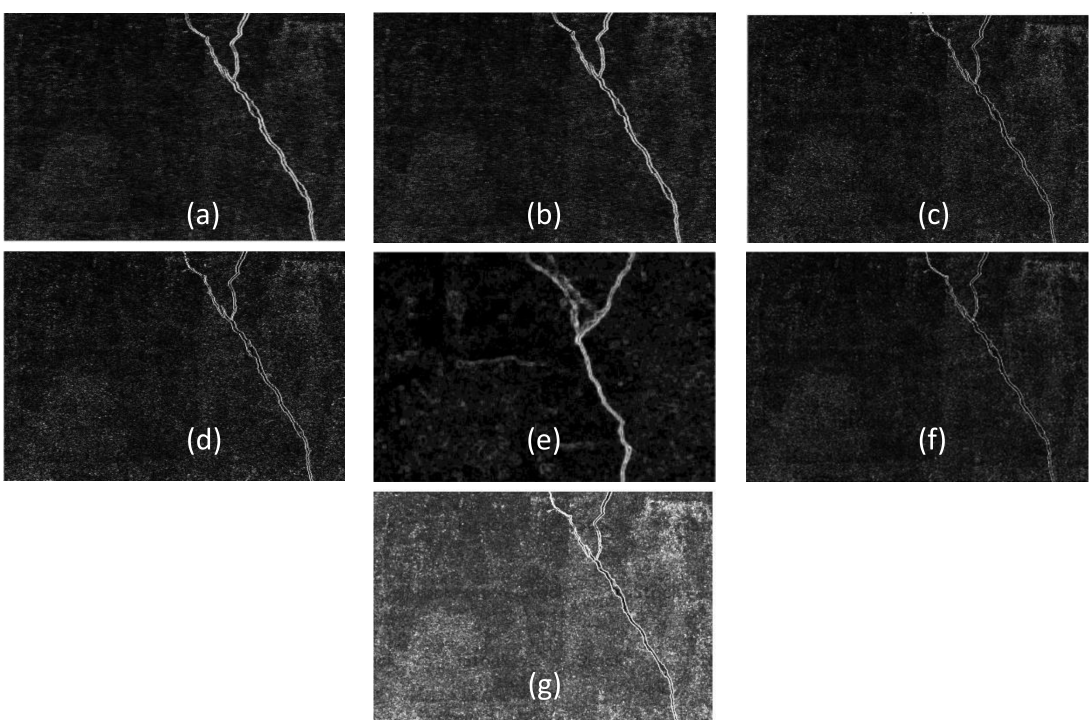

Fig. 7. Comparison of edge detected images. (a) Prewitt (b) Sobel (c) Canny (d) Robert (e) Kirsch (f) Robinson (g) proposed

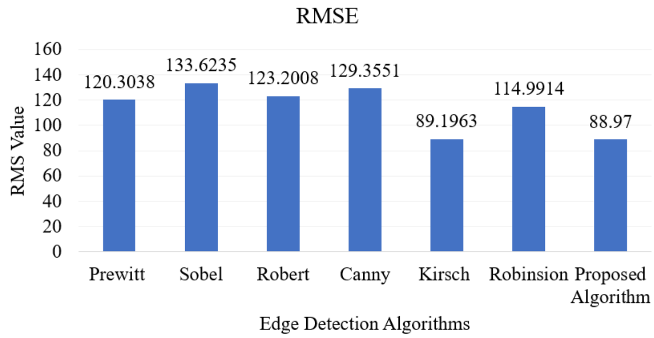

Fig. 8. Graphical representation of RMSE value in Table 3

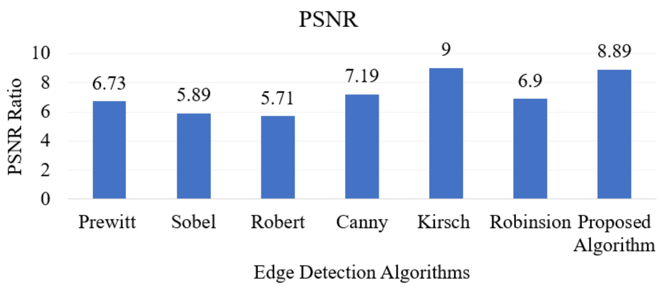

Fig. 9. Graphical representation of PSNR value in Table 3

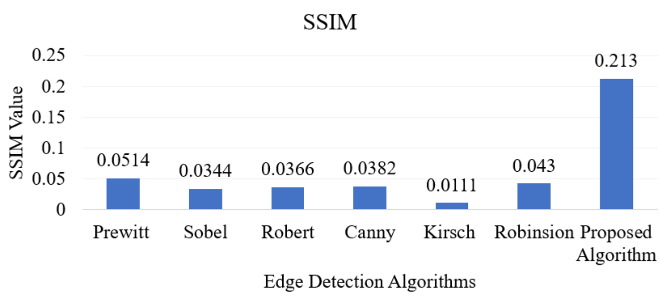

Fig. 10. Graphical representation of SSIM value in Table 3

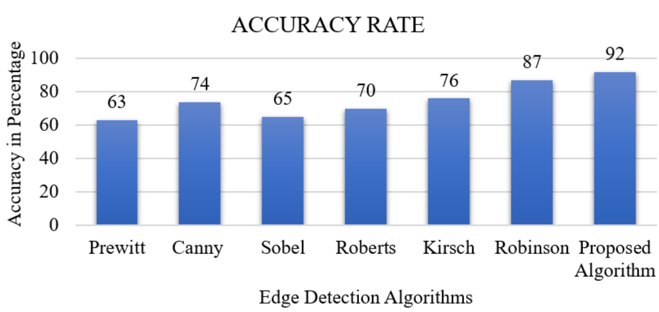

Fig. 11. Graphical representation of accuracy rate in Table 4

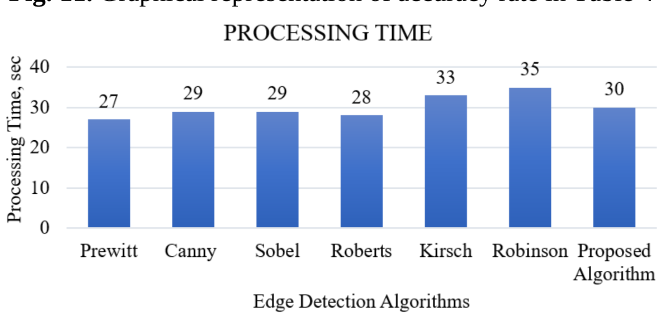

Fig. 12. Graphical representation of SSIM value in Table 4

4. Conclusion

By attaining 92% accuracy and taking 0.5 to 1 sec to process a single tile image, the proposed algorithm proven to be a better alternative to the existing defected detection algorithms. The reason of better accuracy rate is dependent on sequence of procedural steps, quality of input image, and quality of edge detected image. The less processing time makes the proposed algorithm very practical for implementation in real time defect detection systems. However, the proposed algorithm attained less accuracy in detecting hair line cracks and colour fading present in some of the tile images. Our future work will focus on making the proposed algorithm more efficient by adding the ability to detect small cracks, defects, and colour fading.

5. References

[1] "What is statistical process control? SPC quality tools | ASQ", Available from https://asq.org/quality-resources/statistical-process-control, [29 Jan 2021].

[2] H. Elbehiery, A Hefnawy, and M Elewa. "Surface defects detection for ceramic tiles using image processing and morphological techniques", Proceedings of 3rd World Enformatika Conference (WEC), 2005 vol. 5, no. 5, pp. 158–162.

[3] Y.C. Samarawickrama, and D.W. Chamira, "Matlab based automated surface defect detection system for ceremic tiles using image processing", 6th National Conference on Technology and Management (NCTM), Malabe, Sri Lanka, IEEE, 2017, pp. 34–39.

[4] N.A. Shire, M.M. Khanapurkar, and S.M. Rajashri, "Plain ceramic tiles surface defect detection using image processing", 4th International Conference on Emerging Trends in Engineering and Technology, Port Louis, Mauritius, IEEE, 2011, pp. 215–220.

[5] P. Kumari and S. Kumar Gupta, "Morphological image processing GUI using MATLAB", Trends Journal of Sciences Research, vol. 2, no. 3, pp. 90–94, 2015,

[6] R. M. Yousaf, H. A. Habib, H. Dawood, and S. Shafiq, "A comparative study of various edge detection methods", 14th International Conference on Computational Intelligence and Security, 2018 ,pp. 96–99.

[7] P. Ganesan, and G. Sajiv, "A comprehensive study of edge detection for image processing applications", International Conference on Innovations in Information, Embedded and Communication Systems, Janua, 2018, pp. 1–6.

[8] B. R. K and S. A. Angadi, “Crack detection in ceramic tiles using zoning and edge detection methods”, Int. J. Trend Sci. Res. Dev., vol. 2, no. 4, pp. 2844–2847, 2018.

[9] W. S. AlAzawee, I. Abdel-Qader, and J. bdel-Qader, "Using morphological operations; Erosion based algorithm for edge detection", IEEE Int. Conf. Electro Inf. Technol. June 2015, pp. 521–525.

[10] B. Mohammad, S. Naidu, T. Ravikanth, P. Siva, "Identification and rejection of defective ceramic tile using image processing and Arduino", International Journal for Modern Trends in Science and Technology, vol. 2, no. 4 ,pp. 57–60, 2015.

[11] J. Li, and N. Wang, " A study of crack detection algorithm", Proc. - 5th Int. Conf. Instrum. Meas. Comput. Commun. Control, 2016, pp. 1184–1187.

[12] P. Selvakumar, and S. Hariganesh, "The performance analysis of edge detection algorithms for image processing", International Conference on Computing Technologies and Intelligent Data Engineering, 2016, pp. 1–5.

[13] J. Li and N. Wang, “A study of crack detection algorithm”, 5th International Conference on Instrumentation and Measurement, Computer, Communication and Control, Qinhuangdao, China, 2015, pp. 1184–1187.

[14] P. P. Acharjya, R. Das, and D. Ghoshal, "A study on image edge detection using the gradients", Int. J. Sci. Res. Publ., vol. 2, no. 12, pp. 2–6, 2012.

[15] G. N. Chaple, R. D. Daruwala, and M. S. Gofane, "Comparisons of robert, prewitt, sobel operator based edge detection methods for real time uses on FPGA", International Conference on Technologies for Sustainable Development, 2017, no.1 , pp.1–4.

[16] G. N. Chaple, R. D. Daruwala, and M. S. Gofane, "Comparisons of robert, prewitt, sobel operator based edge detection methods for real time uses on FPGA", International Conference on Technologies for Sustainable Development, 2017, no.1, pp.1–4.

[17] C. Rafael et al., “Digital image processing,” Second Edition, Prentice Hall, 2004, pp. 519-560 and 617-621.

[18] R. Srisha and A. Khan, “Morphological operations for image processing : understanding and its applications”, NCVSComs-13, December 2013, no. 13, pp. 17–19.

[19] P. Kumari and S. Kumar Gupta, “Morphological image processing GUI using MATLAB”, Trends Journal of Sciences Research, vol. 2, no. 3, pp. 90–94, 2015,

[20] C. Shagf, “Image edges detection using morphological operations”, J. Kerbala Univ., vol. 6, no. 4, 2008

[21] "Cracked tiles high res stock images | shutterstock", Available https://www.shutterstock.com/search/cracked+tiles, [11 Feb 2021].

[22] "Perfect ceramic tile images, stock photos & vectors | shutterstock", Available from https://www.shutterstock.com/search/perfect+ceramic+tile, [11 Feb 2021].

[23] P. Topno, and G. Murmu, "An improved edge detection method based on median filter", Proc. 3rd Int. Conf. 2019 Devices Integr. Circuit DevIC, 2019, pp. 378–381.

[24] P. P. Acharjya, R. Das, and D. Ghoshal, "A study on image edge detection using the gradients", Int. J. Sci. Res. Publ., vol. 2, no. 12, pp. 2–6, 2012.

[25] A. Abdullah, M. Ali, and H. Jawad, "Evaluation of the performance of kirsch and robinson methods on different color spaces", International Journal of Simulation: Systems, Science and Technology. vol. 20, no. 3, pp. 1-9, 2019.

[26] U. Sara, M. Akter, and M. S. Uddin, “Image quality assessment through FSIM, SSIM, MSE and PSNR—a comparative study”, JCC, vol. 07, no. 03, pp. 8–18, 2019.

[27] D. Sadykova and A. P. James, “Quality assessment metrics for edge detection and edge-aware filtering: A tutorial review”, International Conference on Advances in Computing, Communications and Informatics, Udupi, 2017, pp. 2366–2369.

[28] G.H. Chen, C.L. Yang, L.M. Po, and S.L. Xie, “Edge-based structural similarity for image quality assessment”, IEEE International Conference on Acoustics Speed and Signal Processing Proceedings, Toulouse, France, 2006, vol. 2, pp. 933-936.

