



Implementation of Template Matching on Detection of Stop Line Violations

Dwira Kurnia Larasati, Iwan Setyawan

Faculty of Electronic and Computer Engineering, Satya Wacana Christian University, Salatiga, Indonesia

ARTICLE INFORMATION

Received: April 06, 2021
 Revised: October 22, 2021
 Available online: November 30, 2021

KEYWORDS

Traffic violation detection, stop line, zebra crossing, template matching.

CORRESPONDENCE

Phone: +62 895-3873-53611
 E-mail: 612018043@student.uksw.edu

A B S T R A C T

Road marking is a sign located above the road surface that combines either a line or a symbol and provides information as visual guidance to road users. One example of road marking is the stop line behind the zebra crossing. Many violations of pedestrian rights are committed by motorized vehicles, especially because vehicles do not stop behind the stop line. This paper proposes a system to detect this type of traffic violation using the template matching method. This method is a technique in digital image processing to find small parts of the image that match the image template, so it can detect whether there is a traffic violation or not. This system uses the appropriate template images for each dataset (morning, afternoon, and evening). The system produces accuracy performances for the morning and afternoon dataset of about 100% and 78% respectively, and for the evening dataset is 70%. So, the overall average accuracy rate is 83%. The main factor affecting the overall accuracy performance is the inability to automatically extract the best template for the dataset from the morning and evening datasets.

INTRODUCTION

Road marking is a sign located above the road surface that combines either a line or a symbol and provides information as visual guidance to road users [1]. Road markings must be made as clear as possible so that road users can understand and the traffic will be safe.

One example of a road marking is the stop line behind the zebra crossing. Zebra crossing is a useful road marking as a crossing between pavements and the roadway [2]. Whereas, stop line behind the zebra crossing is a simple line to inform the drivers of the vehicle where they must stop [3]. It helps pedestrians cross the road safely when the red light is on. However, road users often ignore the road markings and cause many violations, namely by not stopping behind this line [4].

According to WHO's Global Status Report on Road Safety 2015, Indonesia ranks in fifth place for the highest number of road accidents in the world [5]. WHO also updated their data in 2018 Road Traffic Accidents Deaths that Indonesia reached 41,862 fatalities in road traffic accidents [6]. The analysis of road traffic accident data in Indonesia from 2013 to 2019 published by Statista Research Department reveals that the average number of fatalities reach over 28,000 [7]. Based on road fatality statistic data, it is estimated that road fatality in Indonesia will reach 40,000 in 2020 and up to 65,000 per year for 2035 [8]. Although <https://doi.org/10.25077/jnte.v10n3.898.2021>

not all road traffic accidents are caused by stop line violations, it is described in [9] that in Indonesia, a high motorcycle population can give a negative impact, such as violations, because of the accumulation of traffic violations by the motorcyclist, particularly at crossroads. One particular violation is not stopping behind the stop lines at zebra crosses, and therefore blocking or disturbing pedestrians crossing the road. This form of violations can be reduced by posting a police officer at every zebra crossing. However, this requires a large number of officers and is therefore impractical.

From the problem definition above, it is obvious that a solution is needed for minimizing stop line violations to ensure traffic safety. The main contribution of this paper is proposing a system that can detect stop line violations automatically using a template matching algorithm. The template matching is used for detecting whether the stop line is present. If it does, it can then show whether shown that there is a violation or not.

Applications of template matching for image-based pattern recognition can be widely found in the literature. One example of such implementation is a door lock security system using Raspberry Pi proposed in [10]. This study presented a system based on image processing to replace the use of RFID that gives less flexibility. The user can simply position themselves in front of the Raspberry Pi camera and its data will be matched using the template matching method with an image in database. The experimental result has a success rate of 96%. Another example

is presented in [11] in which template matching method is used for Optical Character Recognition (OCR) and obtain different accuracy rates for different fonts, namely 100% for Calibri, 100% for Verdana, 86.66% for Arial, 80% for Lucida Fax, and lower accuracy for Cambria and Times New Roman.

Previous research on automatic detection of road marking violations can also be found in the literature. In [12], the Boost-CNN method and automatic hard mining method are used for the stop line detector. The Boost-CNN method is a combination of an AdaBoost classifier and a CNN. It achieves 91.5% in accuracy. Whereas, in [13] the detection of stop line violation system is made using Haar Cascade Classifier to identify the vehicles that commit the violation. It can characterize vehicles well and obtain an accuracy of 91.5% at 720p resolution. Another research made a prototype system to identify a vehicle whether it is halted before the stop line or not by infrared sensor module [14]. Later, its signal is sent to Raspberry Pi to capture the license plate number of vehicles that commit violations. The number in the image will be identified and converted to audio to be voiced by Google Voice as a warning. This system produces an accuracy of 80%. However, to the best of the authors' knowledge, no system based on template matching has been found.

In this paper, the authors propose the use of a template matching method to detect this type of traffic violation. Compared to the previously mentioned methods, template matching offers a much simpler mathematical model and computation. This method also relies only on the captured camera image and does not require any additional hardware or sensors. In this paper, the system is implemented using MATLAB R2017a.

The paper is organized as follows. The METHOD section presents data acquisition, the method, and analysis. The result of the experiments is discussed in the RESULT AND DISCUSSION section. Finally, the CONCLUSIONS section provides some conclusions and pointers for future work.

METHOD

Data Acquisition

In this research, the author used closed-circuit television (CCTV) video footage in Simpang Empat Bareng Arah Solo, Klaten as input data. The video is accessed online and live at [15]. In this video, the red light lights up for about 1 minute and has a frame rate of about 30 fps in size of 1366x768. The video was taken over several days at three distinct time ranges: morning, afternoon, and evening. All the data were taken when the red light was on. The morning dataset is collected at about 10:00 to 12:00 WIB, the afternoon dataset is collected at about 14:00 to 15:30 WIB, and the evening dataset is around 19:15 to 20:30 WIB. For each period, 50 frames were obtained containing 25 frames indicating stop line violation and 25 frames indicating stop line violation. The examples of the images in each dataset are shown in Figure 1, Figure 2, and Figure 3.

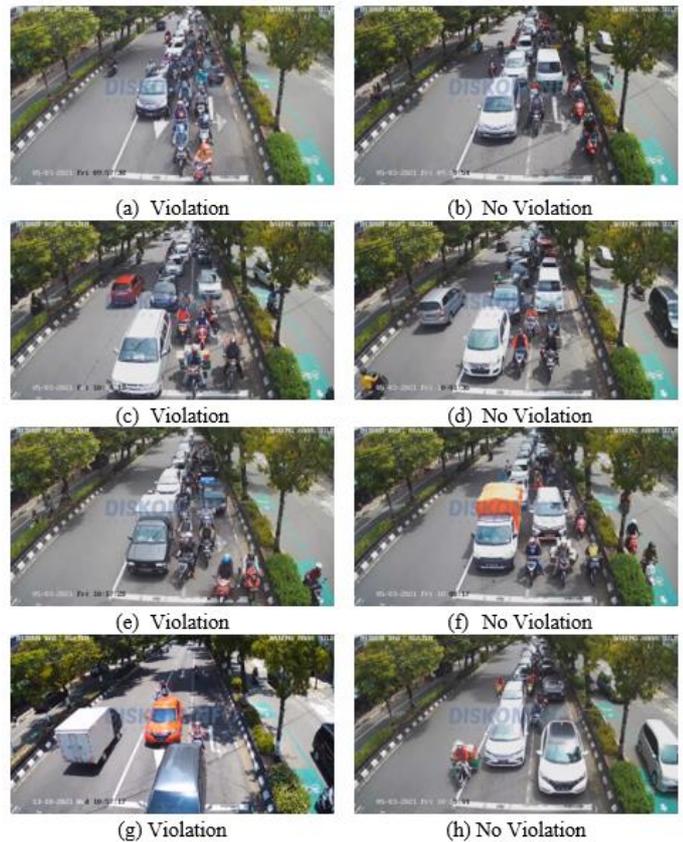


Figure 1. Examples from The Morning Dataset

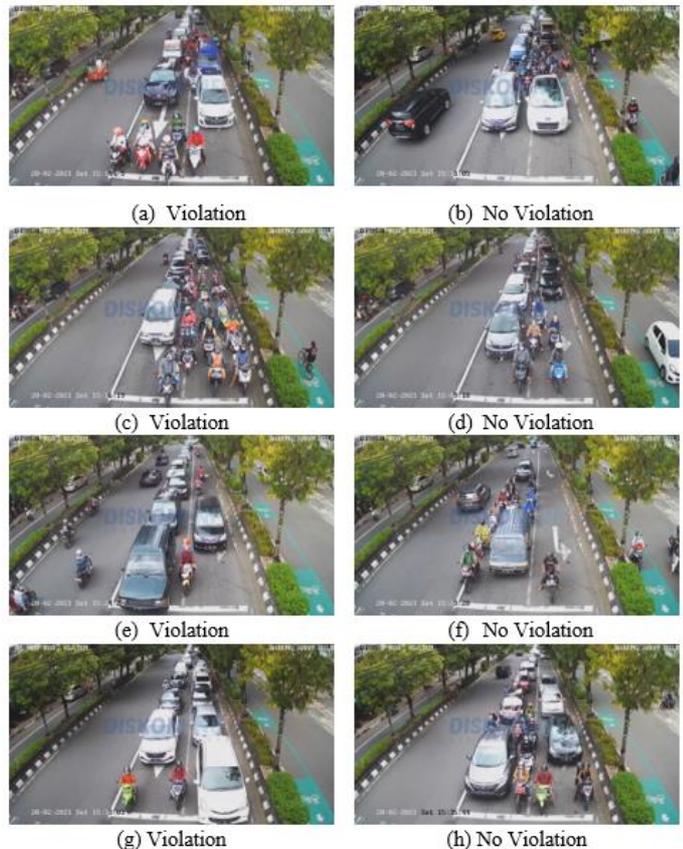


Figure 2. Examples from The Afternoon Dataset

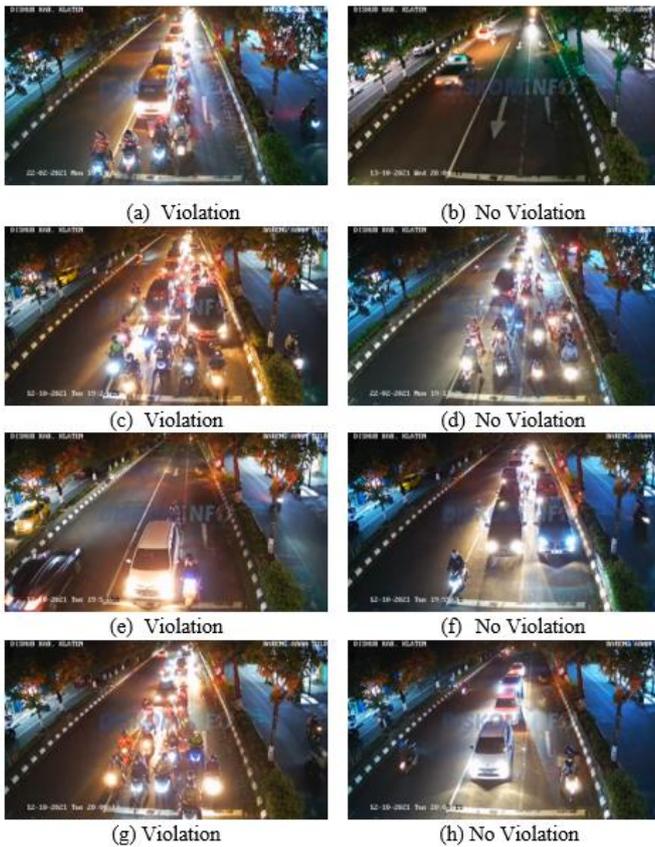


Figure 3. Examples from The Evening Dataset

Template Matching

The method used in this paper is template matching, which is one of the most frequently used techniques in digital image processing that can measure the similarity of all pixels (or features) within the template and a candidate window in the target image [16]. A template is defined as representative local features selected from images [17]. Template matching algorithms work by comparing input image patterns with template image patterns in datasets. The template is moved through every position in the input image to identify whether images/patterns are similar to the template present in the input image.

The author takes the stop line template behind the zebra cross from Figure 1(h), Figure 2(b), Figure 3(b) in subsection Data Acquisition as shown in Figure 4, Figure 5, and Figure 6. The template images are taken in these figures because there is no violation, so the stop line behind the zebra cross is visible. Each template image is used at the appropriate time (morning template image for morning dataset, etc.).



Figure 4. The Template Image from Morning Dataset



Figure 5. The Template Image from Afternoon Dataset



Figure 6. The Template Image from Evening Dataset

The algorithm of stop line detection using the template matching method is as follows. The first step is pre-processing of the input image. The pre-processing process is used to improve computation efficiency and processing speed as well as noise removal in the image. This process consists of:

- a. Input image resizing
The input image is an RGB image with a size of 1366×768 which is then cropped and resized to a size of 591×69 which focuses on the zebra cross stop line. This step aims to improve the computation efficiency and speed.
- b. Input image conversion to Grayscale
In the proposed system, color information is not needed, and therefore the processed image is converted from color to grayscale. A grayscale image is an image that only has shades of gray colors [18]. The grayscale intensity is stored as an 8-bit integer giving 256 possible shades from black to white.
- c. Padding
The padding process is the process of adding pixels on each left, right, top, and bottom side of the image data. It is often required when part of the filter extends beyond the input image [19]. Here the author uses zero padding, i.e., the value of the added pixels is 0. Padding is used for the affixed template area to the input image that is not outside the image. The size of padding that the authors use is 10×10. The authors choose the small size of padding because there is a possibility of the camera movement due to wind.

The next stage is classification using a template matching algorithm by calculating the similarity value of the template image with the input image. This value is calculated by the normalized correlation formula shown as follows [20]:

$$c_{ij} = \frac{\sum_u \sum_v [T(u,v) - \bar{T}] \sum_u \sum_v [I(i+u, j+v) - \bar{I}_T]}{\sqrt{\sum_u \sum_v [T(u,v) - \bar{T}]^2 \sum_u \sum_v [I(i+u, j+v) - \bar{I}_T]^2}} \quad (1)$$

The equation above produces the value of $-1 \leq c_{ij} \leq +1$. This value will be compared to the threshold value t_c , if the value $c_{ij} \geq t_c$ then the template image resembles the input image in that position. However, if the value $c_{ij} \leq t_c$, then that particular area of the input image does not resemble the template image.

Threshold value t_c is determined empirically, as shown in Figure 7. The authors chose two values for t_c : 0.8 and 0.9 because the values produce better accuracy. If the value $c_{ij} \geq t_c$, then the template is detected in the input image so that no violation occurs. Otherwise, if the value $c_{ij} \leq t_c$, then the template is not detected in the input image. This may be due to the stop line being obstructed by a vehicle not stopping behind it properly and is thus an indication that there is a violation.

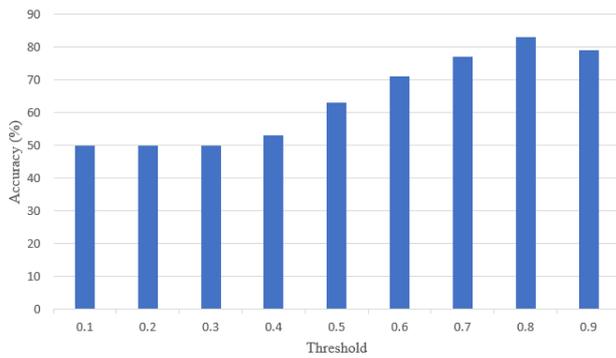


Figure 7. Graph of Accuracy Percentages at Various Threshold Value

Analysis

The output of the process carried out by the system for both the threshold value t_c 0.8 and 0.9 is the classification that predicts input into one of two classes, namely the “violation” class and “no violation” class. There are four possible outputs, namely true positive (TP), false negative (FN), true negative (TN), and false positive (FP). True Positive (TP) is when an input image belonging to the positive class is classified into the positive class by the system. If such an image is classified into the negative class by the system, we have a False Negative (FN). When the input image belongs to the negative class and is classified by the system into the negative class, we have a True Negative (TN), otherwise when such image is classified into the positive class then we have a False Positive (FP) [21]. These relationships are shown as the confusion matrix shown in Figure 5. In this paper, an input image is said to belong to the positive class when it contains a stop line violation. Conversely, an image is said to belong to the negative class when it does not contain a stop line violation.

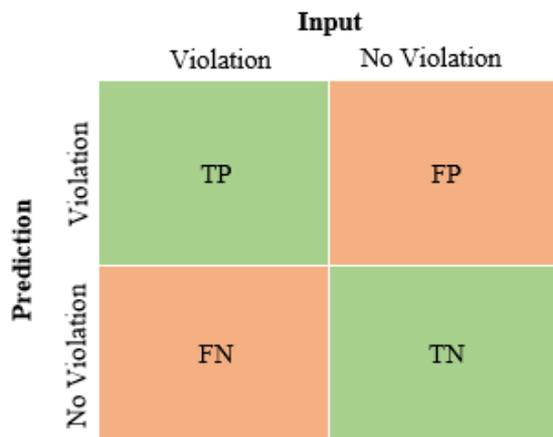


Figure 5. Confusion Matrix

The most common indicators (accuracy, precision, and recall) are obtained by using the formulas (2), (3), (4) [22].

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \times 100\% \tag{2}$$

$$\text{Precision} = \frac{TP}{TP+FP} \times 100\% \tag{3}$$

$$\text{Recall} = \frac{TP}{TP+FN} \times 100\% \tag{4}$$

RESULTS AND DISCUSSION

The number of classified images for each class as well as the percentage of accuracy, precision, and recall are shown in Table 1 and Table 2 for threshold 0.8; Table 3 and Table 4 for threshold value of 0.9.

Table 1. Detection Result for Violation Detection with Threshold Value of 0.8

Time	Total	TP	FP	TN	FN
Morning	50	25	0	25	0
Afternoon	50	14	0	25	11
Evening	50	21	11	14	4

Table 2. Accuracy, Precision, and Recall Percentage for Violation Detection with Threshold Value of 0.8

Time	Accuracy	Precision	Recall
Morning	100%	100%	100%
Afternoon	78%	100%	56%
Evening	70%	66%	84%
Average	83%	89%	80%

Table 3. Detection Result for Violation Detection with Threshold Value of 0.9

Time	Total	TP	FP	TN	FN
Morning	50	25	9	16	0
Afternoon	50	22	0	25	3
Evening	50	25	20	5	0

Table 4. Accuracy, Precision, and Recall Percentage for Violation Detection with Threshold Value of 0.9

Time	Accuracy	Precision	Recall
Morning	82%	74%	100%
Afternoon	94%	100%	88%
Evening	60%	56%	100%
Average	79%	77%	96%

Table 2 and Table 4 show that the average accuracy percentage of the system for threshold values of 0.8 and 0.9 are 83% and 79%, respectively. The authors chose the threshold value of 0.8 because the average value of accuracy is higher compared to the threshold value of 0.9. However, it should be noted that while the overall average accuracy drops, increasing the threshold to 0.9 raises the accuracy rate for the afternoon dataset.

In the dataset of the variation in the morning time, many images are classified as FP (shown in Table 3) because the images taken in the time where there are shadows of trees (effects of sunlight) that cover the observed stop line, so it is classified as a violation, even if it is just a shadow.

In the dataset of variations in the evening time, some images show no violation but the program detects the violations. This is because of the color-changing of the asphalt (effects of the vehicle’s headlamp) so that the input image is not similar to the template image.

From the results above, the template matching method has the potential to be used in the detection of violations in the stop line behind the zebra crossing.

CONCLUSIONS

This research proposes a system to detect the stop line violation using the template matching method. The system classified whether vehicles halt over the stop line causing the violation or not by comparing the input image to the stop line template image. The system shows an accuracy performance for the morning and afternoon dataset of about 100% and 78% respectively, and for the evening dataset is 70%. So, the average accuracy performance is 83%. However, this research is limited to one place only, manually changing the template for each period, and depending on the good lighting condition.

Future work will continue to improve the system like using automatically extracting and changing the best template images in all conditions, places, and weather so a better level of accuracy performance is obtained.

REFERENCES

- [1] D. Babić, A. Šćukanec, D. Babić, and M. Fiolić, "Model for Predicting Road Markings Service Life," *Balt. J. Road Bridg. Eng.*, vol. 14, no. 3, pp. 341–359, Sep. 2019, Accessed: Oct. 12, 2021. [Online]. Available: <https://bjrbe-journals.rtu.lv/article/view/bjrbe.2019-14.447>.
- [2] T. Bjørnskau, "The Zebra Crossing Game – Using game theory to explain a discrepancy between road user behaviour and traffic rules," *Saf. Sci.*, vol. 92, pp. 298–301, Feb. 2017, doi: 10.1016/j.ssci.2015.10.007.
- [3] A. Abijith, B. Abishek, and K. S. P. V. S. Nihith, "Enforcing the stop line rule using radio frequency identification tags," *Proc. 2018 Int. Conf. Commun. Comput. Internet Things, IC3IoT 2018*, pp. 195–198, Mar. 2019, doi: 10.1109/IC3IoT.2018.8668162.
- [4] Y. Song, C. Wang, and L. Wang, "On the Relationship between Traffic Violation Behavior and Crash Involvement Using Locally Linear Embedding and Improved K-Means Algorithm," *CICTP 2019 Transp. China - Connect. World - Proc. 19th COTA Int. Conf. Transp. Prof.*, pp. 3646–3657, 2019, doi: 10.1061/9780784482292.316.
- [5] Herman and M. Tamosee, "Indonesia Ranked 5th in the World for Highest Number of Road Accidents," 2016. <https://jakartaglobe.id/news/indonesia-ranked-5th-world-highest-number-road-accidents/> (accessed Oct. 13, 2021).
- [6] "Road Traffic Accidents in Indonesia." <https://www.worldlifeexpectancy.com/indonesia-road-traffic-accidents> (accessed Oct. 13, 2021).
- [7] Statista Research Department, "Indonesia: number of fatalities in traffic accidents 2019," 2021. <https://www.statista.com/statistics/979068/indonesia-number-of-fatalities-traffic-accidents/> (accessed Oct. 13, 2021).
- [8] A. Jusuf, I. P. Nurprasetio, and A. Prihutama, "Macro data analysis of traffic accidents in Indonesia," *J. Eng. Technol. Sci.*, vol. 49, no. 1, pp. 133–144, 2017, doi: 10.5614/J.ENG.TECHNOL.SCI.2017.49.1.8.
- [9] A. M. Mulyadi, "Motorcycle Red Box Evaluation at Signalized Intersections in Bogor: Traffic Flow, Occupancy Rate and Stop Line Violation," *IPTEK J. Proc. Ser.*, vol. 3, no. 6, Dec. 2017, doi: 10.12962/J23546026.Y2017I6.3240.
- [10] B. Widiawan, S. Kautsar, F. Purnomo, and B. Etikasari, "Implementation of Template Matching Method for Door Lock Security System Using Raspberry Pi," *VOLT J. Ilm. Pendidik. Tek. Elektro*, vol. 2, no. 2, pp. 143–148, Oct. 2017, doi: 10.30870/VOLT.V2I2.2047.
- [11] M. A. Hossain and S. Afrin, "Optical Character Recognition based on Template Matching," *Glob. J. Comput. Sci. Technol.*, pp. 31–35, May 2019, doi: 10.34257/GJCSTCVOL19IS2PG31.
- [12] G. T. Lin, P. S. Santoso, C. T. Lin, C. C. Tsai, and J. I. Guo, "Stop line detection and distance measurement for road intersection based on deep learning neural network," *Proc. - 9th Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. APSIPA ASC 2017*, vol. 2018-Febru, pp. 692–695, Feb. 2018, doi: 10.1109/APSIPA.2017.8282121.
- [13] M. P. Anggadharma and Y. Widiastuti, "Breaches Detection in Zebra Cross Traffic Light Using Haar Cascade Classifier," in *Proceedings - 2nd International Conference on Informatics, Multimedia, Cyber, and Information System, ICIMCIS 2020*, Nov. 2020, pp. 272–277, doi: 10.1109/ICIMCIS51567.2020.9354275.
- [14] D. Hirawan, A. Hadiana, and A. Abdurakhim, "The Prototype of Traffic Violation Detection System Based on Internet of Things," in *IOP Conference Series: Materials Science and Engineering*, Nov. 2019, vol. 662, no. 2, p. 022084, doi: 10.1088/1757-899X/662/2/022084.
- [15] "CCTV Pemerintah Kabupaten Klaten." <https://cctv.klatenkab.go.id/?sel=all> (accessed Jun. 19, 2021).
- [16] J. Cheng, Y. Wu, W. AbdAlmageed, and P. Natarajan, "QATM: Quality-Aware Template Matching for Deep Learning," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11553–11562.
- [17] W. Lin *et al.*, "Multiscale Digital Porous Rock Reconstruction Using Template Matching," *Water Resour. Res.*, vol. 55, no. 8, pp. 6911–6922, Aug. 2019, doi: 10.1029/2019WR025219.
- [18] R. Fisher, S. Perkins, A. Walker, and E. Wolfart, "Glossary - Grayscale Images," 2003. <https://homepages.inf.ed.ac.uk/rbf/HIPR2/gryimage.htm> (accessed Oct. 13, 2021).
- [19] G. Liu *et al.*, "Partial Convolution based Padding," Nov. 2018, Accessed: Oct. 13, 2021. [Online]. Available: <https://arxiv.org/abs/1811.11718v1>.
- [20] M. H. Annaby, Y. M. Fouda, and M. A. Rushdi, "Improved Normalized Cross-Correlation for Defect Detection in Printed-Circuit Boards," *IEEE Trans. Semicond. Manuf.*, vol. 32, no. 2, pp. 199–211, May 2019, doi: 10.1109/TSM.2019.2911062.
- [21] S. Ruuska, W. Hämmäläinen, S. Kajava, M. Mughal, P. Matilainen, and J. Mononen, "Evaluation of the confusion matrix method in the validation of an automated system for measuring feeding behaviour of cattle," *Behav. Processes*, vol. 148, pp. 56–62, Mar. 2018, doi: 10.1016/J.BEPROC.2018.01.004.
- [22] Y. Xu, Y. Wang, J. Yuan, Q. Cheng, X. Wang, and P. L. Carson, "Medical breast ultrasound image segmentation by machine learning," *Ultrasonics*, vol. 91, pp. 1–9, Jan. 2019, doi: 10.1016/J.ULTRAS.2018.07.006.

NOMENCLATURE

Meaning of symbols used in the equations and other symbols presented in your article must be presented in this section.

c_{ij}	normalized cross correlation at position (i, j)
$T(u, v)$	template image T at position (u, v)
\bar{T}	means of the template image T pixels
$I(u, v)$	input image I at position (u, v)
\bar{I}_T	means of the input image I when it is covered by template image T at position (i, j)

AUTHOR(S) BIOGRAPHY

Dwira Kurnia Larasati

She is an undergraduate student of Electronic Engineering from Universitas Kristen Satya Wacana in her third year. The author's interest is in the field of computer vision and machine learning.

Iwan Setyawan

Iwan Setyawan is currently a lecturer at the Department of Electronic Engineering of Universitas Kristen Satya Wacana, Salatiga. His research interests include digital image and video processing, digital watermarking, and pattern recognition.