INTERNATIONAL JOURNAL OF AUTOMOTIVE
SCIENCE AND TECHNOLOGY
2020, VOL. 4, NO:3, 132-137
www.dergipark.gov.tr/ijastech

Comparison of the Edge Detection Methods According to Fake Edge and Area Calculation Performances

Murat Alparslan Güngör[1*] and Seyfi Polat[2]

0000-0001-7446-7808[1] , 0000-0002-7196-3053[2]

[1] *Electrical and Electronics Engineering Department, Faculty of Engineering, Hitit University, Çorum, 19030, Turkey*
[2] *Mechatronics Engineering Department, Faculty of Engineering, Hitit University, Çorum, 19030, Turkey*

Abstract

Determining the borders of the objects in the images and detection of the edges is an important step for image processing applications. In this study, the areas of different objects were calculated using edge detection methods. The edges of real-time images taken from the camera were detected with traditionally used Sobel, Prewitt and Canny filters. In addition, these filters were applied to the images obtained using Basic Global Thresholding (BGT) pre-processing and Mean Thresholding (MT) pre-processing to detect the edges in the image. After eliminating the fake edges in the edge images, the areas of the obtained images were calculated. The methods used in this paper were compared according to the fake edge and the area calculation performances.

Keywords: Area calculation; Edge detection; Fake edge; Thresholding

* Corresponding author
Murat Alparslan Güngör
alparslangungor@hitit.edu.tr
Adress: Electrical and Electronics Department, Faculty of Engineering, Hitit University, Çorum, Turkey
Tel:+903642274533
Fax: +903642274535

Research Article

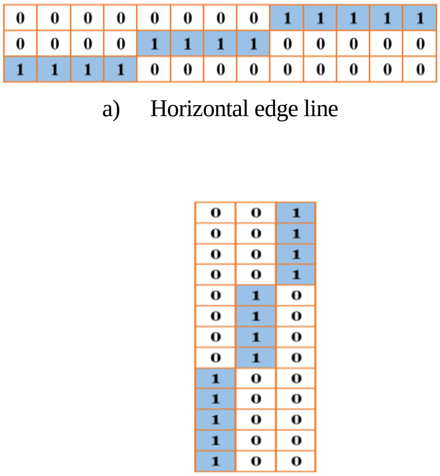
Manuscript
Received 24.04.2020
Revised 20.05.2020
Accepted 10.07.2020
Doi: 10.30939/ijastech..726332

1. Introduction

Image processing plays an increasingly important role in a wide range of disciplines and fields, in many areas of our daily lives, in science and technology, with applications such as television, photography, robotics, remote sensing, medical diagnostics and industrial inspection. Furthermore, one of the most important areas where image processing is commonly used is autonomous vehicles. The object identification from the images obtained with autonomous vehicle cameras is basically done by the edge detection algorithms.

Determining the edges of the objects in the images is an important step for image processing applications. The goal of edge detection algorithms is to find transitions that show the edges in the images. Sobel [1], Prewitt [2] and Canny [3] are basic edge detection algorithms used in the literature. There are many studies on images using these popular algorithms [4-16].

Problems such as environment, transmission channels, and imperfections in the image acquisition system, especially cheap cameras, cause poor quality of the images taken from the camera [17-18]. Thus, the edges in edge images obtained as a result of edge detection algorithms may not continue on a single line as shown in Fig. 1.



0	0	0	0	0	0	0	0	1	1	1	1	1
0	0	0	0	1	1	1	1	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0	0	0

a) Horizontal edge line

0	0	1
0	0	1
0	0	1
0	0	1
0	1	0
0	1	0
0	1	0
0	1	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0

b) Vertical edge line

Fig. 1. Edges in edge images are not on a single line

The problem shown in Fig.1 is the spatial deviation of the edge pixels indicated by "1". If the real figure is a straight line, the edge must be obtained on a single line. If the real figure is curved, the edge is expected to be the pattern to give the curved shape. Deviations in the ideal edge image cause errors in operations such as area calculation.

Another problem as a result of edge detection algorithms is the formation of parallel line in the edge image obtained as shown in Fig. 2. In this study, the parallel line formed in the edge image was named as fake edge.

0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1

Fig. 2. The parallel line formed in the edge image

Deviations and fake edges in the edge images of Fig. 1 and Fig. 2 are undesirable. The large number of fake edges in the edge image causes low performance of image processing applications. In addition, the fake edges may cause the images to be misinterpreted.

In this study, the images of two different objects were obtained with the camera under three different environmental conditions which are dark, light dark and bright, and then the edges were detected with above mentioned edge detection methods. The performances of the fake edge and area calculation of the methods were compared based the obtained edge images.

2. Material and Method

As the first order derivative methods, Sobel, Prewitt and Canny filters are popular methods used to detection of the edges in the image. The Canny filter uses a multi-step algorithm for detection of the edges. In this method, the image is first smoothed using a Gaussian filter. After applying the gradient operator on the smoothed image, the two thresholds are used to detect the edges in the image. Sobel and Prewitt filters usually use the 3x3 matrices shown in Fig. 3 and Fig. 4 for detection of edges as respectively. The difference of the Sobel filter from the Prewitt filter is that it has a value of 2 in the window.

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

Fig. 3. Sobel filter

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

Fig. 4. Prewitt filter

To detect the edges in the image, the edge detection filters can be applied directly to the image or to the pre-processed image. In this study, MT and BGT algorithms were used as pre-processing. The difference between the two algorithms is the method of calculating the threshold value. In the MT algorithm, the threshold value is calculated by dividing the total pixel value in the image by the number of pixels in the image [19]. In BGT algorithm, the threshold value is calculated as a result of a multi-step algorithm. In this algorithm, the procedure used to calculate the threshold value (T) is defined as follows [20]:

1- Initial T value is determined.

2- The image is divided into two groups according to whether each pixel value in the image is bigger or smaller than T value.

3- Average gray level values (μ_1 and μ_2) of the pixel values in each group are calculated.

4- New T value is calculated using the formula $T=0.5 (\mu_1+\mu_2)$.

5- Steps 2-3-4 are repeated with iterations until the difference in T is small enough.

In both algorithms, the aim is to convert the gray level image into a black-white image according to the calculated threshold value. In this study, Sobel, Prewitt and Canny filters were applied to both directly and pre-processed images to detect the edges in the images. The applied nine methods and the abbreviation for each method are given in Table 1.

Table 1. The methods and abbreviations

Method	Pre-processing	Abbreviation
Sobel	-	S
Prewitt	-	P
Canny	-	C
Sobel	BGT	SBGT
Prewitt	BGT	PBGT
Canny	BGT	CBGT
Sobel	MT	SMT
Prewitt	MT	PMT
Canny	MT	CMT

After obtaining edge images using the any of nine methods shown in Table 1, we determined the fake edges in these images before calculating the area of the geometric shape. In this paper, the following procedure was applied to determine the fake edges:

•One edge pixel of the corresponding geometric shape in the edge image of the object is selected as the starting pixel.

•Each pixel is connected to one pixel after it.

•The boundary determination for the corresponding geometric shape ends when the starting pixel is reached.

•Thus, each pixel that form the edges of the geometric shape has two connections.

The fake edges of a sample image are indicated in Fig. 5.

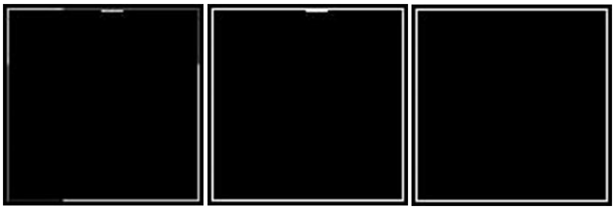


Fig. 5. Fake edges of a sample image. a) Sample image, b) Edge image, c) Edge image after eliminating the fake edges

Fig. 5-a shows a sample gray level image. Fig. 5-b is the edge image of the image shown in Fig.5-a. As mentioned in Fig. 2, there are fake edges on the upper line of the geometric shape shown in Fig. 5-b. After determining and eliminating the fake edges using the procedure mentioned above, Fig. 5-c is obtained. For any geometric shape which has many fake edges or invisible fake edges, above-mentioned procedure can be used to detect the fake edges.

After the fake edges were eliminated and the boundaries of the geometric shape were determined, the area of the geometric shape was calculated. Pixel-based method was applied for the area calculation. In this method, the number of pixels of the geometric shape with certain boundaries is determined. Then, the area of the corresponding geometric shape is calculated by multiplying the area occupied by a pixel by the determined number of pixels.

While the nine methods shown in Table 1 directly cause fake edges, they do not calculate the area. The area is calculated based on the outputs of these methods. These methods cause the deviations in the edge image as shown in Fig. 1. These deviations cause errors in operations such as area calculation. Thus, we compared these methods according to both the fake edge and area calculation performances.

3. Results and Discussion

In this study, one camera, platform and two objects were used as shown in Fig. 6.

The first object shown in Fig. 6-a consists of eight different geometric shapes. The nine area calculations were made for this object, including the object itself. The second object has a larger area than first object shown in Fig. 6-b. Thus, the performances of the applied method according to the size of the area were analyzed. Fig. 7 shows sample processed images of the object taken from the camera. After converting the RGB image taken from the camera into gray level image, the image shown in Fig. 7-a was obtained by cutting the gray level image. Black-white image was obtained by applying MT or BGT algorithm to the image shown in Fig. 7-a (Fig. 7-b). The edge image was obtained by using the methods expressed in Table 1. The image in Fig. 7-c was obtained by applying filter directly to the image in Fig. 7-a and the image in Fig. 7-d was obtained by applying filter to the pre-processed image in Fig. 7-b.

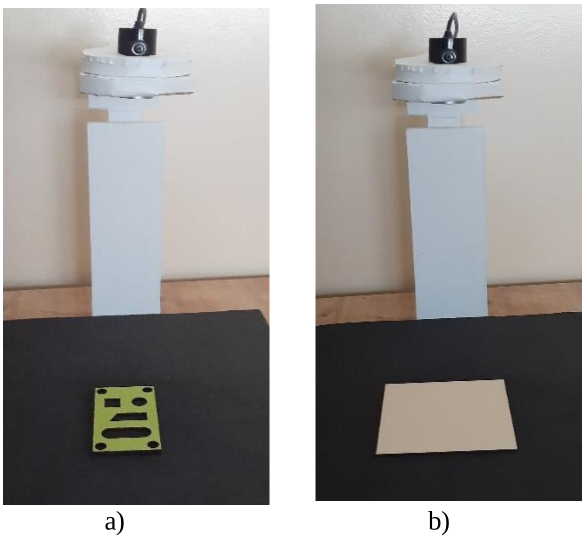
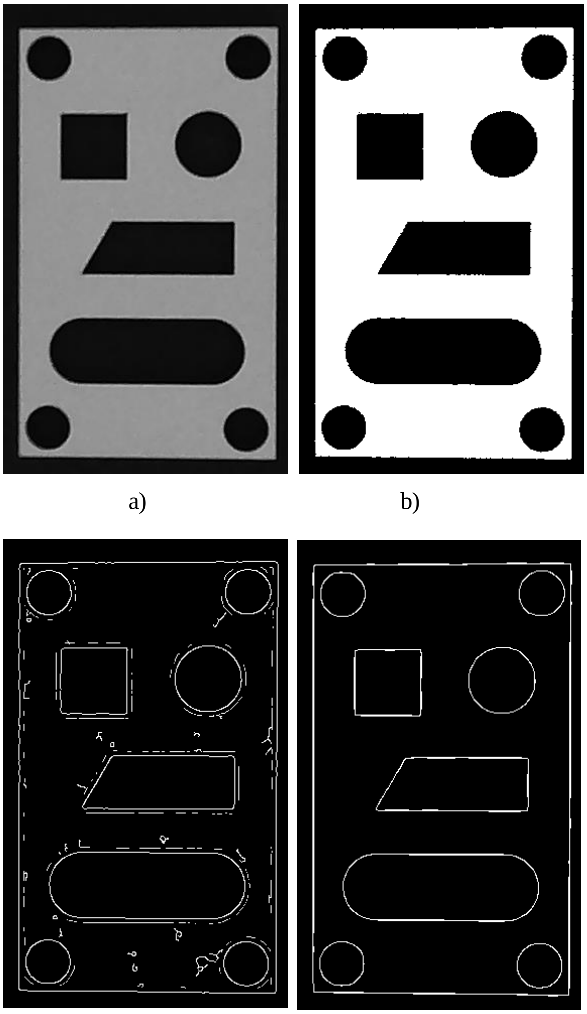


Fig. 6. Camera, platform and objects a) First object with different geometric shapes, b) Second object with larger area than first sample

For the sample processed images shown in Fig. 7, we used MT algorithm to obtain Fig. 7-b, C and CMT methods to obtain Fig. 7-c and Fig. 7-d, respectively. As shown in Fig. 7, CMT method caused less fake edge in the image than C method for this sample.

Each object was analyzed in three different environmental conditions (named as dark, light dark and bright). Thus, in this study, fifty-four edge images obtained for three different environmental conditions, nine different methods, two different objects were used to evaluate the performance.

While the boundaries are determined in the image, the line may show spatial deviations as seen in Fig. 1. These spatial deviations do not pose a problem for fake edge detection. The fake edge appears if there are lines parallel to each other as shown in Fig. 2. In this study, the edges of fifty-four images were examined manually. The fake edges were eliminated in cases where fake edges could be identified as shown in Fig. 2. However, situations were also encountered where fake edges could not be identified as shown in Fig. 8. In this case, fake edges could not be eliminated and these edge images could not be processed in the area calculations.



a) b)

c) d)

Fig. 7. Sample processed images of the object taken from the camera. a) The image of first object, b) The pre-processed image of first object, c) The image of first object obtained with direct filtering, d) The image of first object obtained by applying the filter after pre-processing.

0	0	0	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	0	0	0	0

Fig. 8. Situations where fake edges could not be identified.

In the study, firstly, the performances of the methods were compared according to obtaining fake edges. Table 2 shows the fake edge numbers of the images obtained for the first object according to the methods and environments mentioned above. Similarly, Table 3 shows the fake edge numbers for second object. The cells marked in Table 2 and Table 3 represent the images selected for the area calculation because the fake edges in these images could be identified and eliminated.

Table 2. The numbers of fake edges for first object with nine methods under three different environments

	Environment		
Method	**Dark**	**Light Dark**	**Bright**
SBGT	**0**	**0**	**0**
S	57	**4**	53
SMT	**3**	**1**	**0**
PBGT	106	29	**13**
P	38	4	53
PMT	124	56	**16**
CBGT	1550	1664	1827
C	1612	2759	4364
CMT	1379	1567	1847

Table 3. The numbers of fake edges for second object with nine methods under three different environments

	Environment		
Method	**Dark**	**Light Dark**	**Bright**
SBGT	**5**	**0**	**1**
S	832	194	670
SMT	1005	832	**0**
PBGT	141	**58**	57
P	786	192	654
PMT	1522	539	47
CBGT	1138	1564	1508
C	1017	3242	6743
CMT	460	446	1470

When Table 2 and 3 are analyzed; among the S, P, C methods, while S and P gave close results (even P gave slightly better results), C gave the worst results according to the fake edge performance. It was observed that the fake edge was absent or very few with the SBGT method in Tables 2 and 3. Area calculations were made by eliminating the few fake edges of these images. For the SMT method, it was observed that there was no fake edge for the second object only after the application of the bright environment as shown in Table 3 while the areas were calculated for the all images of the first object by eliminating the fake edges as shown in Table 2. Tables 2 and 3 show when S was applied directly, it was used only one image for the area calculation. Although the P method gave the same or slightly better results than the S method, only the fake edges in three images could be eliminated for the P, PBGT and PMT methods. When C, CBGT and CMT methods were evaluated; CBGT and CMT methods caused less fake edges in the image than C method but did not improve poor performance enough. Consequently, any image could not be used to calculation of the area with or without preprocessing using C application. As a result, Tables 2 and 3 show that SBGT method has the best performance among the above mentioned nine methods according to the fake edge performance. The small number of fake edges can make it easier to eliminate fake edges. However, the image in the light dark environment in Table 2 for the P method could not be included in the area calculation because the fake edges could not be identified although it has few fake edges.

The area calculations performances of the methods were analyzed after the fake edges of nine images for the first object and five images for the second object were eliminated. For the first object, the area calculation of nine geometric shapes, including the object itself shown in Fig. 9, was made separately. For the second object, the area of the object was calculated directly.

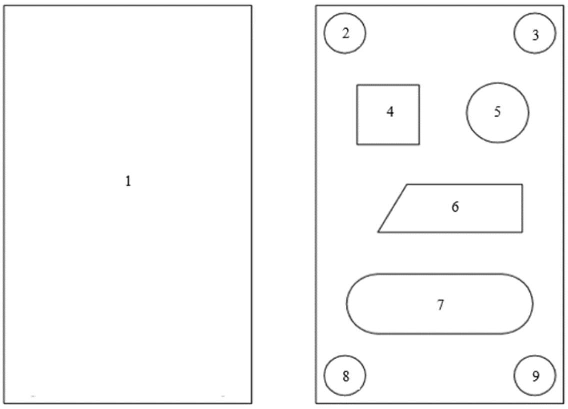


Fig. 9. Nine geometric shapes for first object.

The absolute error shown in Eq. 1 was used to analyze the performance of the area calculation.

$$\text{Absolute Error (\%)} = \frac{|\text{Real Area}-\text{Calculated Area}|}{\text{Real Area}}100 \quad (1)$$

Firstly, real areas of geometric shapes were found. Then, the areas of geometric shapes were calculated with the pixel-based method from the edge image. While calculating the area with pixel based method, 4 reference points (midpoints of the right edge, left edge, bottom edge and upper edge) were determined from the edge image. The distance between each of these reference points was matched with the real geometric shape and it was found how many “mm” long a pixel corresponds. The area was calculated using the total number of pixels remaining in the frame from the edge image. Finally, the absolute error between the real area and the pixel-based calculated area was calculated using Eq. 1. The absolute error calculated for both objects is shown in Table 4 and Table 5.

Table 4. Absolute errors for the first object

Environment	Method	Absolute Error (%)								
		The number of geometric shape								
		1st	2nd	3rd	4th	5th	6th	7th	8th	9th
Dark	SBGT	0,43	10,96	13,90	7,86	9,09	6,79	4,93	11,38	12,22
	SMT	0,23	12,85	16,00	8,64	10,06	7,30	5,42	12,71	13,34
Light Dark	SBGT	1,63	0,03	3,89	1,62	3,30	3,02	1,68	0,32	3,11
	S	2,02	2,63	1,43	0,65	1,78	2,55	0,80	3,05	0,10
	SMT	1,45	2,20	6,34	2,36	4,83	5,16	2,53	2,41	5,29
Bright	PBGT	1,43	2,63	1,57	2,65	1,62	2,86	1,16	3,33	0,67
	PMT	1,35	0,74	2,83	2,77	2,43	3,44	1,45	1,79	1,01
	SBGT	1,43	2,63	1,57	2,65	1,62	2,86	1,16	3,33	0,67
	SMT	1,35	0,74	2,83	2,77	2,43	3,44	1,45	1,79	1,01

Table 5. Absolute errors for the second object

Environment	Method	Absolute Error (%)
Dark	SBGT	0,27
Light Dark	SBGT	0,32
	PBGT	0,32
Bright	SBGT	0,65
	SMT	0,02

Each geometric shape shown in Table 4 corresponds to geometric shape numbered in Fig. 9. The first geometric shape, which shows the object itself in Fig. 9, has a larger geometric area than the other geometric shapes in the first object. The second object, which gives absolute error values for different environments in Table 5, has the largest area of all geometric shapes including first and second objects. When the absolute error values for the 1st geometric shape shown in Table 4 were analyzed: the absolute error values obtained for the SMT method for the dark and light dark environments were better than the SBGT method. In addition, when the absolute error values obtained for 1st geometric shape with the bright environment were compared, PMT method performed better than PBGT method and SMT method performed better than SBGT method. When Table 5 was examined, the SMT method for the bright environment had a better absolute error value than the SBGT method. As a result of the comparison of MT and BGT thresholding algorithms according to the area calculation performances, it is observed that the MT algorithm gives better results than the BGT algorithm if the calculated area is large.

When comparing the methods for Sobel and Prewitt filters shown in Table 4 for the bright environment: for all geometric shapes, PBGT method gave the same results with SBGT method and PMT method gave the same results with SMT method. When the light dark environment for the second object in Table 5 was examined, it was observed that SBGT method and PBGT method had the same results. Thus, if the Sobel and Prewitt filters are compared according to the area calculation performance, these two filters have the same performance regardless of the shape size and the MT and BGT thresholding algorithms used for preprocessing.

4. Conclusions

In this study, images of two different objects with different sizes and geometric shapes in three different brightness environments were taken from the camera and their edges were detected with nine different methods. In the obtained edge images, firstly fake edges were examined and methods were compared according to fake edge performance. After eliminating the fake edges in the edge images, the area calculation performances of the methods were evaluated. The results show that the best method according to the fake edge performance is to use the Sobel filter after applying BGT as a pre-processing. In addition, MT pre-processing has a better area calculation performance for objects with large area than BGT pre-processing. Finally, it has been observed that Sobel and Prewitt filters used after MT or BGT preprocessing have the same area calculation performance.

References

[1] Sobel, I. (1970). Camera models and perception. *Ph.D. thesis*, Stanford University, CA.

[2] Prewitt, J. (1970). Object Enhancement and Extraction. *Picture Processing and Psychopictorics*, NY, Academic Pres.

[3] Canny, J. (1986). A Computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8, 679-700.

[4] Jin, S., Li, X., Yang, X., Zhang, J. A., and Shen, D. (2019). Identification of tropical cyclone centers in sar imagery based on template matching and particle swarm optimization Algorithms. *IEEE Transactions on Geoscience and Remote Sensing*, 57(1), 598-608.

[5] Rafati, M., Arabfard, M., Zadeh, M. R. R., and Maghsoudloo, M. (2016). Assessment of noise reduction in ultrasound images of common carotid and brachial arteries. *IET Computer Vision*, 10(1), 1-8.

[6] Kalra, A., and Chhokar, R. L. (2016). A Hybrid approach using sobel and canny operator for digital image edge detection. *International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE)*, pp. 305-310.

[7] Guiming, S., and Jidong, S. (2016). Remote sensing image edge-detection based on improved Canny operator. *8th IEEE International Conference on Communication Software and Networks (ICCSN)*, pp. 652-656.

[8] Ye, H., Shen, B., and Yan, S. (2018). Prewitt edge detection based on BM3D image denoising. *IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pp. 1593-1597.

[9] Attivissimo, F., Cavone, G., Lanzolla, A. M. L., and Spadavecchia, M. (2010). A technique to improve the image quality in computer tomography. *IEEE Transactions on Instrumentation and Measurement*, 59(5), 1251-1257.

[10] Furnari, A., Farinella, G. M., Bruna, A. R., and Battiato, S. (2017). Distortion adaptive Sobel filters for the gradient estimation of wide angle images. *Journal of Visual Communication and Image Representation*, 46, 165-175.

[11] Menaka, R., Janarthanan, S., and Deeba, K. (2020). FPGA implementation of low power and high speed image edge detection algorithm. *Microprocessors and Microsystems*, 103053.

[12] Biswas, S., and Ghoshal, D. (2016). Blood cell detection using thresholding estimation based watershed transformation with Sobel filter in frequency domain. *Procedia Computer Science*, 89, 651-657.

[13] Vardhana, M., Arunkumar, N., Lasrado, S., Abdulhay, E., and Ramirez-Gonzalez, G. (2018). Convolutional neural network for bio-medical image segmentation with hardware acceleration. *Cognitive Systems Research*, 50, 10-14.

[14] Aybar, E. (2008). Edge detection on color images using sobel operator. *Afyon Kocatepe University Journal of Science*, 8(1), 205-217.

[15] Aybar, E. (2014). Edge detection using connection MAP. *Anadolu University Journal of Science and Technology B - Theoritical Sciences (AUJST-B)*, 3(1), 21-32.

[16] Lalimi, M. A., Ghofrani, S., and McLernon, D. (2013). A vehicle license plate detection method using region and edge based methods. *Computers & Electrical Engineering*, 39(3), 834-845.

[17] Hsia, C. H., Wu, T. C., and Chiang, J. S. (2017). A new method of moving object detection using adaptive filter. *Journal of Real-Time Image Processing*, 13(2), 311-325.

[18] Szczepanski, M. (2019). Fast spatio-temporal digital paths video filter. *Journal of Real-Time Image Processing*,16(2), 477-489.

[19] Ahmad, N. S., Zaki, Z. M., and Ismail, W. (2014). Region of adaptive threshold segmentation between mean, median and otsu threshold for dental age assessment. *International Conference on Computer, Communications, and Control Technology (I4CT)*, pp. 353-356.

[20] Rongwen Lu (2020). Basic global thresholding (https://www.mathworks.com/matlabcentral/fileexchange/38390-basic-global-thresholding), *MATLAB Central File Exchange*. Retrieved.