

Image Processing for Rapidly Eye Detection based on Robust Haar Sliding Window

Fitri Utamingrum, Renaldi Primaswara, Yuita Arum Sari

Faculty of Computer Science, Brawijaya University, Indonesia

Article Info

Article history:

Received Dec 6, 2016

Revised Feb 15, 2017

Accepted Mar 3, 2017

Keyword:

Detection

Eyes

Haar cascade classifier

Sliding window

ABSTRACT

Object Detection using Haar Cascade Classifier widely applied in several devices and applications as a medium of interaction between human and computer such as a tool control that utilizes the detection of eye movements. Obviously speed and precision in the detection process such as eyes, has an effect if implemented on a device. If the eye could not detect accurately, controlling device systems could reach bad detection as well. The proposed method can be used as an approach to detect the eye region of eye based on haar classifier method by means of modifying the direction of sliding window. In which, it was initially placed in the middle position of image on facial area by assuming the location of eyes area in the central region of the image. While the window region of conventional haar cascade scan the whole of image start from the left top corner. From the experiment by using our proposed method, it can speed up the the computation time and improve accuracy significantly reach to 92,4%.

Copyright © 2017 Institute of Advanced Engineering and Science.

All rights reserved.

Corresponding Author:

Fitri Utamingrum,

Faculty of Computer Science,

Brawijaya University,

Veteran Road No. 8 Malang, East Java, Indonesia.

Email: f3_ningrum@ub.ac.id

1. INTRODUCTION

In digital image processing, there are several methods that have been used and one of them is haar cascade classifier. Haar cascade has been widely applied for humans and computers interaction (HCI). This implementation frequently used for the detection of object such as facial area, mouth detection, and eye detection. Eye gaze tracking doing by Ghani et al. [1] for controlling mouse pointer, Gao et al. [2] developed a method based on haar classifier, detection of eye movement for controlling wheelchair present by Singh et al. [3] using EOG signal, and Coetzer et al [4] implement SVM and ANN for real time eye detection. The process of haar cascade in detecting eyes is using the technique of the sliding window or rectangular window on the whole image and find out if there is a part of the image that is shaped like an eye or not. Haar based on cascade models include a weight connection from the input to each layer and from each layer to the successive layers [5]. Before doing the detection of the eye area, there are some process of image processing in advance. RGB image initially converted into a grayscale image, the result will be calculate using integral image at the pixel of grayscale image. Haar features process grayscale image in a rectangular window with a certain size such as 24 x 24 pixels. Inside this window, the filtering process performed to find the eye region. This filtering was done using selected haar feature template. If the value of the haar features in a certain region below the threshold then it has to be clustered. However, if a feature in a region is not qualified, so there is no object to be detected in the area of the region and its region area relocates the area. However, if the area meets the feature, then the rules are performed to the next feature. If all requirements fulfilled feature, it means that there is an object in the area of the region.

In detecting an object, Haar feature has a fairly good accuracy rate, but using haar feature requires a long computation time. In accordance with the haar method, this happens because haar do processing on all parts of the image pixel sequentially. In training process haar used AdaBoost for classifier the object [6],[7]. However, eye detection can affect the accuracy control of device or application that utilizes the eye area accurately. Besides those methods, occasionally image come up with noise often affect the quality of the image. So that, it has an impact on the accuracy in eye detection [8]. Previous research and methods offered by Viola Jones [9] which use certain window sizes moving at each pixel from the upper left corner to the bottom right.

In this paper, we proposed a method by utilizing a sliding window and haar features, which the detection window starts from the center of the image which already indicates the area of eye to reduce the processing time and errors in eye detection. For classification, we also proposed method by using simply nearest distance compared haar with adaboost. It is also considering to use Open CV with nothing change of scanning windows region.

2. PROPOSED METHOD

After successfully detect objects in the facial area, our proposed method is placing a sliding window in the middle area of the face image, not from the top left corner as was done by the conventional haar cascade. Previously, images have been through grayscale process and change the pixel value into the integral image, determination ROI of eye, than in this area performed calculation to get features value of eye. In this case we aware that there are some factors can disturbing the detection of objects such as lighting, so that in testing phase, the images have good lighting conditions and taken pictures during in the morning or in the afternoon. To ease the understanding of eye detection process in general, diagram block in Figure 1 represents the method that we proposed.

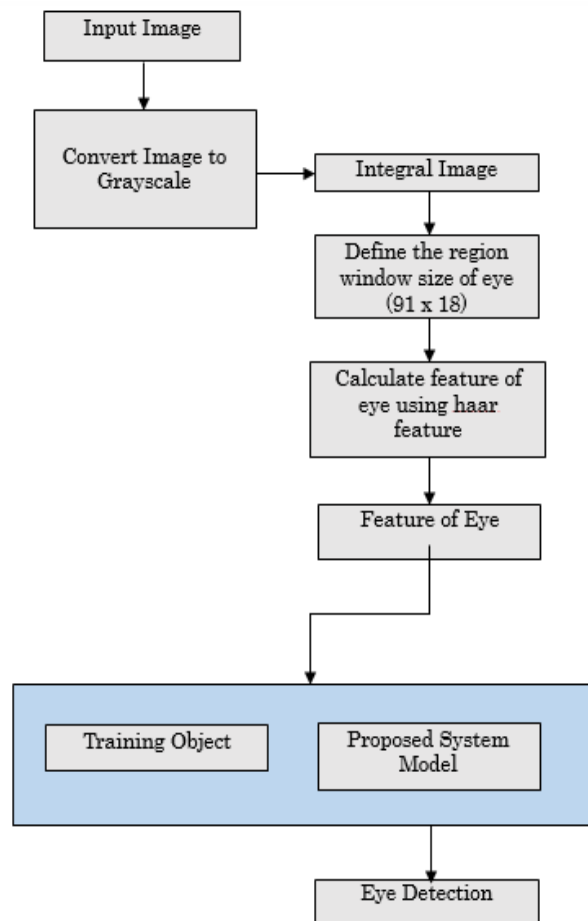


Figure 1. Eye Detection Block Diagram of Proposed System

2.1. Scaling

Scaling process is a way to resize a digital image, it is necessary to all images which are processed have the same size. In this system the scaling process performed by utilizing the scaling package provided on Java.

2.2. Grayscale Image

In preprocessing phase, we need to remodel color image into grayscale image, which applied in haar method as well [10]. So that, it need to transform from RGB image into grayscale image. To convert image into grayscale, *average* method is utilized, by adding up all the value of *R G B*, then divided by 3, in order to obtain an average value of *R G B*, the average value of that can be said as grayscale.

$$\text{Grayscale} = (R + G + B) / 3 \quad (1)$$

2.3. Integral Image

Integral image is a technique used in the haar method for calculating the value of the rectangle features quickly by changing the value of each pixel in the grayscale image. Then generating value summed area table so that the results regarded as a representation of a new image. This method has the advantage that relatively faster computing, because it depends on the number of pixels in a square instead of each pixel value of an image. Figure 2 may improve the integral calculation.

0	0	0	0	0	0
0	5	2	3	4	1
0	1	5	4	2	3
0	2	2	1	3	4
0	3	5	6	4	5
0	4	1	3	2	6

(a)

0	0	0	0	0	0
0	5	7	10	14	15
0	6	13	20	26	30
0	8	17	25	34	42
0	11	25	39	52	65
0	15	30	47	62	81

(b)

Figure 2. The calculation result of rectangle feature have the same value; (a) original image and (b) represent the integral image

The region result value of original image shows in Figure 2(a): $5+2+1+5+2+2 = 17$. By using that calculation, it needs five steps calculation. In Figure 2(b), region result value of integral image: $17-0-0+0 = 17$, so it is only need four steps calculation. This explains that the calculation of the integral image requires fewer steps than the calculation of each pixel in a region. The calculation of integral image performs faster in large-sized image rather than in small-sized image.

2.4. Training Phase for Eye Feature

To detect eye feature on facial area, it is necessary to train on many eye images dataset by various forms or types of a person's eyes [11]. The dataset containing eye images with a look ahead (frontal) and the variation of the left and right views obtained at random from people around the research. Training phase is done by the viola Jones et al using haar template. The window size used was 92×19 pixels in accordance by the size of the dataset of eye, while the size of the box feature is the 4×3 . Figure 3 will clarify eye dataset.



Figure 3. Some training sample of eye with some condition of lighting and skin color

2.5. Direction of Sliding Window

In this paper, window region is placed on the top center area of the face image show in Figure 4(b). The size of the window region that use is same as the size of the window that is used during the process of training data of eye region. The size that we use is 92 x 19 pixels, it is rectangular and in accordance with a window that would indicate that the area contained of eye region.

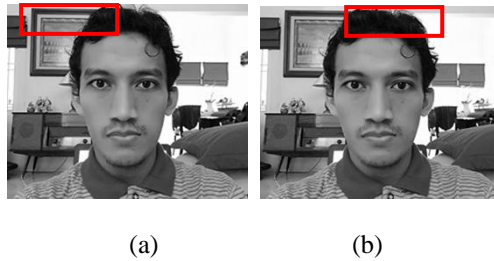


Figure 4. Illustration of Window Region, (a) The window region of conventional haar cascade scan the image start from the left top corner, (b) Our proposed method start scanning the image from top center.

To determine the midpoint of the starting of scanning window, it is necessary to know the width of the pixels in facial area and the size of the box window used due to the determination of the midpoint derived from the width of the image. In this case we use 360 x 240 pixels of the image size or the otherwise, while the size of the box window haar is 92 x 19 pixel. Figure 5 illustrates the method of determining the starting point of scanning window.

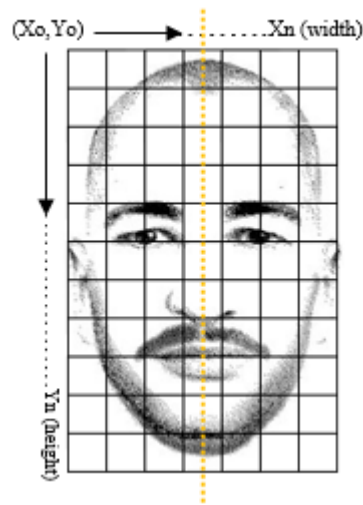


Figure 5. Determine the position of the window region midpoint of the in X point and Y point.

$$(Xn, Yo) = X \frac{(W_{image} - W_{window})}{2}; Yo \quad (2)$$

Image has width (W-image) containing a number of pixel, in which pixel (0,0) located in the top left corner of an image. To determine the pivot point of rectangle window, image width is minus by width of the rectangle window (W-window) then divided by two. So it figures out the coordinates for (Xn, 0) as a reference point of rectangle window. For in case of image with 360 pixel width and 92 pixel width of rectangle window, x point calculated from $X = (360 - 92) / 2 = 134$. Rectangle window starting from the dot pixels (134;0) until the width of the window region ends (226;0) with a window height that adjusted.

Detection process of eye region is obtained from the difference between the values of each feature on the testing data with the values of each feature on the training data in a window rectangular area, and then

summed to generate a single value features and compared whether the value of these features is less than a specified threshold value, illustrated in Figure 6. The features are obtained from black and white squares of haar features.

$$Value(V) = \text{sum}/|Test_{ij} - Train_{ij}| \quad (3)$$

$Test_{ij}$ is a pixel value of testing data features at position i, j , $Train_{ij}$ is a pixel value of train data features at position i, j .

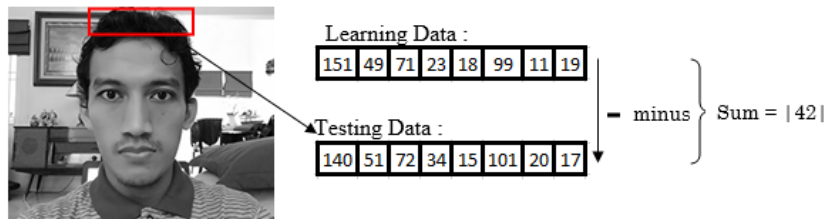


Figure 6. Window region sliding down and generate some feature of eye, calculate and compared each data with learning data

Window boxes shifted every single pixel in the image of the picture. Then through haar features, it requires to examine whether the image area in a window box detected as eye area or not. If the value is less than the threshold, then the window box will continue to run until it stops. In this system we have modified the rules in detecting eye region. If the difference in value features do not have any value which less than the specified threshold then it has to find out the smallest difference value. So, the value is detected as an area of eye.

3. EXPERIMENTAL RESULT AND DISCUSSION

Testing data was taken from the people around the university in Malang city. The dataset captured in bright lighting conditions only and we collected 30 dataset retrieved using MacBook Air with resolution 900 pixels and also from mobile device camera Asus Zenfone Selfie with resolution 1920 x 1080 pixels. As a limitation in this system, the facial image position should be in the middle. We also add several conditions of image testing to be tested in a system that has been created. In this section, we did some experiments for detecting eye area with scaling image size into around of 360 x 240 pixels and 3 x 4 pixel for size of template haar feature. This experiment implemented on Intel Core i5-5200U processor 2,7GHz using Java Netbeans 8.1. The following Figure 7 shows the results of our proposed method.

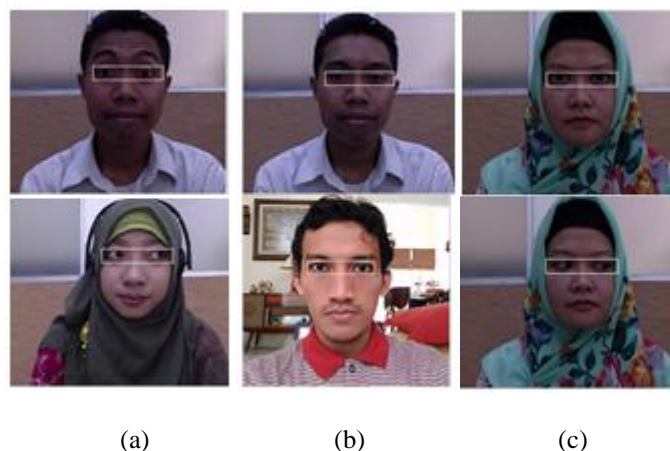


Figure 7. Experimental result from different condition: (a) Eye detection result with different skin type; (b) Eye detection result with different background; (c) Eye detection result with different of eye gaze

Experimental result obtained from calculation using Percentage of Successful Prediction (PSP). Accuracy calculation is important to measure the ability of a system that has been made in recognizing objects. In experimental, it reaches 71.4% detection rate and 11.1% false rate as shown in Table 1 that based on the formula 4.

$$Detection\ Rate : \frac{TP}{TP+TN} \times 100\% \tag{4}$$

TP = person / facial image and succeeded to recognize the eye.
 TN = image is not the eye area, and detected as an eye.

$$Accuracy : \frac{TP+TN}{Total\ Data} \times 100\% \tag{5}$$

Table 1. Eye Detection Result of our proposed method

Method	Detection Rate	False Rate	Processing Time
Haar Cascade + AdaBoost	69,2 %	25 %	3,4 s
Proposed Method	71,4 %	11,1 %	2,9 s

As we can see, the experiment result from Table 1, the proposed method could improve the detection rate and false rate by changing the direction of sliding window. While, the haar cascade method achieves less detection rate and higher false rate. Moreover, the proposed method have faster processing time than haar cascade method.

In this paper, we also applied haar cascade method which generated manually for comparing with our proposed method. To obtain the comparable results, testing process was conducted with the same database. The comparison result already presented in Table 2. The system was measured quantitatively using that has been done by the equations of accuracy [12],[13] in formula 5. The accuracy of proposed method is 93.3% and conventional haar cascade method is 86.67 %. From the Figure 8(a), we can see the difference that conventional haar cascade method missed some of eye detection, and our proposed method can improve the eye detection.

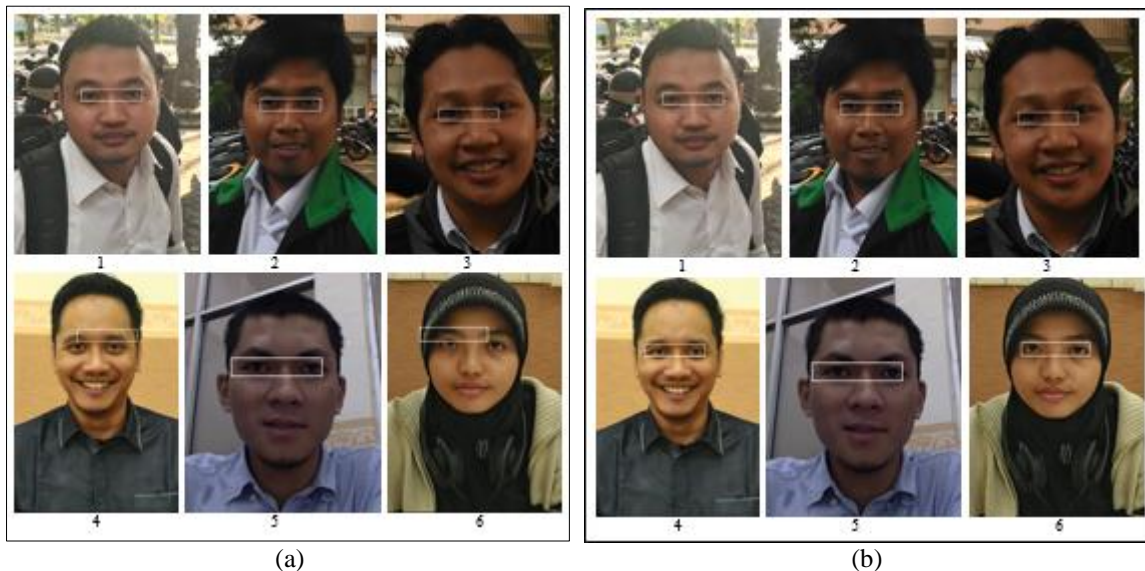


Figure 8. Some of eye detection result from comparison method: (a) Haar cascade method can not detect the fourth and sixth image; (b) The proposed method can fix the eye detection

Table 2. Comparison result using Haar Cascade method and proposed method

Method	Accuracy
Haar + AdaBoost	90,8%
Proposed Method	92,4%

4. CONCLUSION

Minimizing errors and speeding up the detection process can be handled using the proposed system in this paper. Some of testing images can not be detected by using haar conventional methods, for instance, several images with a different eye gaze sometimes also can not be detected. By using the scheme of changing the position of the scanning window in the area of detected face, the detection of eye can be improved effectively, as well as the computation time can be improved. The drawback of this proposed method is come up with any different lighting condition of captured image. Thus for further research, it will improve the performance of image in different lighting condition in order to run more quickly and accurately. Besides that, the approach of the angle of the face can also be performed.

ACKNOWLEDGEMENTS

This work was performed in Computer Research Group, Faculty of Computer Science, Brawijaya University. It was supported by Unggulan Research Grant Faculty of Computer Science, Brawijaya University research fund. We also would like to thank for people who keen to provide pictures as dataset.

REFERENCES

- [1] M. U. Ghani, *et al.*, "Gaze Pointer: A real time mouse pointer control implementation based on eye gaze tracking" 2013 16th Int. Multi Top. Conf. INMIC 2013, pp. 154–159, 2013.
- [2] C. Gao and S. L. Lu, "Novel FPGA based haar classifier face detection algorithm acceleration" Proc. - 2008 Int. Conf. F. Program. Log. Appl. FPL, pp. 373–378, 2008.
- [3] M. Singh, *et al.*, "Eye movement detection for wheelchair control application" *Int. Conf. Electr. Electron. Signals, Commun. Optim. EESCO 2015*, pp. 4–8, 2015.
- [4] R. C. Coetzer and G. P. Hancke, "Eye detection for a real-time vehicle driver fatigue monitoring system," no. IV, 2011.
- [5] S. Goyal and G. K. Goyal, "Cascade Modelling for Predicting Solubility Index of Roller Dried Goat whole Milk Powder," vol/issue: 2(1), pp. 23–28, 2013.
- [6] S. Zhang, *et al.*, "Informed Haar-like Features Improve Pedestrian Detection," *Cvpr*, 2014.
- [7] D. Zhu, *et al.*, "Hybrid Method for Human Eye Detection," pp. 5368–5373, 2014.
- [8] F. Utamingrum, *et al.*, "Mixed gaussian and impulse noise removal based on kernel observation and edge direction," *Int. J. Innov. Comput. Inf. Control*, vol/issue: 11(5), pp. 1509–1523, 2015.
- [9] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," presented at 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. I-511-I-518, 2001.
- [10] M. Mehrubeoglu, *et al.*, "Real-Time Eye Tracking Using a Smart Camera," pp. 1–7.
- [11] Y. M. Cheung and Q. Peng, "Eye Gaze Tracking with a Web Camera in a Desktop Environment," *IEEE Trans. Human-Machine Syst.*, vol/issue: 45(4), pp. 419–430, 2015.
- [12] S. W. K. b. C. F. T. Wei-Chao Lin a, "CANN: An intrusion detection system based on combining cluster centers and nearest neighbors," *Knowledge-Based Systems*, vol. 78, pp. 13-21, 2015.
- [13] R. B. V and R. K. Senapati, "Bright Lesion Detection in Color Fundus Images Based on Texture Features," vol/issue: 5(1), pp. 92–100, 2016.

BIOGRAPHIES OF AUTHORS



Fitri Utamingrum, was born in Surabaya, East Java, Indonesia. She received her Bachelor Degree in Electrical Engineering (BEng.) from National Institute of Technology from Brawijaya University, Malang, Indonesia in 2007. In addition, she obtained the degree of Doctor of Engineering in the field of Computer Science and Electrical Engineering from Kumamoto University, Japan. She has been working as part time lecturer in several institution, such as Sekolah Tinggi Teknik Angkatan Darat (STTAD) from 2006 until 2007. Sekolah Tinggi Teknik Atlas Nusantara (STTAR) from 2006 until 2007. In addition, she also has been teaching the students of Vocational Education Development Center (VEDC) Malang-Indonesia in 2007 and Malang Joint Campus (MJC) in 2007. She has been full time lecturer in Brawijaya University starting from 2008.



Yuita Arum Sari, is currently working at Laboratory of Intelligent System in Brawijaya University. She obtained Bachelor Degree in Computer Science from Brawijaya University, Malang, Indonesia in 2011. Her Master Degree obtained from Department of Informatics, Institute Teknologi Sepuluh Nopember, Surabaya, Indonesia in 2015. Her research are in fields of computer vision, digital image processing, and pattern recognition.



Renaldi Primaswara Praetya, is currently as a magister student in Faculty of Computer Science Brawijaya University. She obtained Bachelor Degree in Computer Science from Brawijaya University, Malang, Indonesia in 2012. Now He is currently finalizing his Master Degree in Computer Science Brawijaya University. His research are in fields of computer vision, speech recognition, and digital image processing.