Eye Detection using Haar Cascade Classifier

Jyotsna More[1], Divyani Sutar[2], Remica Sequeira[3], Vedant Chavan[4]

[1] Professor, Dept. of Information technology, Xavier Institute of engineering, Mumbai, India
[2]Dept. of Information technology, Xavier Institute of engineering, Mumbai, India
[3]Dept. of Information technology, Xavier Institute of engineering, Mumbai, India
[4]Dept. of Information technology, Xavier Institute of engineering, Mumbai, India

---***---

Abstract - *This paper proposes a method that detects the eyes from an image which is acquired from a video that is recorded from the webcam of the laptop/computer. Using this measured data eyes from the image is detected. They firstly convert the video into frames that is an image and then save the images respectively. Here the image is saved in a jpg format. Then the features of the eyes are extracted from the image. Using Haar features of eyes, they detect the position of the eyes. Similarly, the face and other facial parts can be detected using the features that focuses the image. This system is cost efficient as it requires a web camera or personal laptop. It gives the result in less amount of time thus proving to be very time efficient too.*

Key Words: OpenCV, Haar Cascade Classifier, Viola jones algorithm, eye detection,.

1. INTRODUCTION

Real time detection or tracking of the eyes are very popular in the areas of research in computer vision. Detection of eyes is used for gaze tracking and human-computer interaction. Our focus would be in human-computer interaction. This paper is based on an image algorithm to detect eyes in the visible spectrum. The main advantage of such a method is that it is cost efficient and it does not require any other additional hardware. It can work with regular low-cost webcams [6]. The several different methods used are as follows: a) Model-based method, b) Featured-based method, c) Hybrid method and d) Learning based method. In the feature-based method, the eyes detection is done by using the image features related to the position of the eyes. This process is easier and quicker and provide high performance. In the model-based approach, it does not detect the features in the image but rather selects the right model that provides high performance. This paper uses the feature-based model. A video will be recorded of the subjective person using a webcam. Then the video will be converted into several number of frames depending on the frame rate per second. Then by using Haar cascade classifier the eyes detection will be performed. This paper explains the following parts: 1) Capturing a video using webcam, 2) converting the video into frames(image) using OpenCV and finally, 3) Detecting the eyes using Haar cascade classifier.

2. RELATED WORK

1. Eye detection based on OpenCV

This paper includes algorithm based on OpenCV. It also has image processing and computer vision creation of framework to detect driver fatigue eye features. The algorithm used in this paper localizes the eye region and detects its state on the basis of rough to precise thinking, it can accurately locate eye pupils in an eye-open state, which is critical for reducing traffic accidents [4]. This paper uses the Canny operation for eye-area extraction of edges. Canny's operation is focused on thinking about optimization, multiplying SNR and localization which balances the edge operation and cancels noise interference. The Canny operation is identical to the Marr operation. Through the template matching process, this paper implements eye localization and state detection and Hough variant round transformation from rough to precise thinking [4].

2. Eye detection using viola jones algorithm

To classify the eyes in an image, this paper presents and locates the facial characteristics in an image [1]. In this process, the identification of facial parts such as the eyes, is an important activity. This method is used to recognize and identify the components of the human facial variables in an image. The issue of recognition of facial expression may differ between many variables, such as lighting, invariant pose and rotation, etc. The research in this paper includes the Viola-Jones Cascade Object Detector algorithm that offers different combinations of filters and methods to detect these facial expressions [1].

3. Eye detection on Hard Datasets

In this paper, they created a low-light and long-distance image dataset. This faced some problems by eye detectors that solve problems in the real world. The dataset they created consists of re-imaged images (photo head) and semi-synthetic heads that are aged under various low-light, atmospheric blur conditions and 3 m, 50 m, 80 m, and 200 m distances [2]. The analyzation of the efficiency in localization of eye algorithms are paired with the detector of Viola Jones. This paper provided a performance appraisal on a variety of hard datasets of face detection algorithms[2].

4. Automatic Eye Detection and Its Validation

In this paper, they have presented an automated technique for eye detection. Firstly, they have proposed a new automatic eye detection system in real time in this paper. This process is then checked using the FRGC database. The identification of the following results is comparable with manually marked tests for fully automatic eye localization and face recognition [3]. However, as the face alignment is mostly carried out using eye positions, an algorithm with significant accuracy for eye localization is also important for face accuracy. Given that sub-space techniques are typically vulnerable to misalignment, the proposed method of eye localization is very effective [3].

5. Eye localization using Viola-Jones method

In this paper, they show how the eye detection takes place with the help of Viola-jones technique [5]. In the first step Shi-Tomasi detector and K-means clustering is used to determine the eye candidate regions. The second step is the localization of the eye. Using the eye template, you can obtain the following result [5].

3. METHODOLOGY

The Haar cascade classifier is an effective way to detect various objects in the surroundings. This method is also used in detection of face and eyes. The main objective of the Haar cascade classifier is a collection a of a lot of positive images and negative images which are later on to train the classifier. Basically, positive images are the images that you want our classifier to identity and negative images are images of everything they don’t want our classifier to detect.

1. Capturing a video using webcam

This is the first step where the subjective person video is being recorded. A webcam is used for this process. The camera resolution is supposed to be high and of good quality. In such a manner that the image processed in the second step must be clear and recognizable.

2. Converting the recorded video into images(frames) using OpenCV

Using OpenCV the recorded video is converted into several number of frames depending upon the frame rate per second. The frame rate can be set according to the requirements. OpenCV has many functions used for editing and manipulating videos. OpenCV libraries can be used to perform multiple functions on videos. Taking a video as an input and then converting it into several number of frames. The following functions can be used to do the respective.

Functions used

cv2.VideoCapture("path_name"): used to extract recorded video for further processing.

os.makedirs('folder_name'): used to create a folder.

cam.read(): used to read frames.

cv2.imwrite(frame_name, frame): used to save the read frame(image).

3. Eye detection using Haar cascade classifier

The initial step is to train the classifier. This method requires a set of two images. First set consist of positive images. The second set consist of negative images. Positive images will capture the eye features and negative image will capture the images without eye features. Next the features of the eyes will be extracted from the images. Haar features are shown below in the figure are used for detection. Next, for every feature this method subtracts the sum of white rectangle pixels from the sum of black rectangle pixel that is shown in the figure.

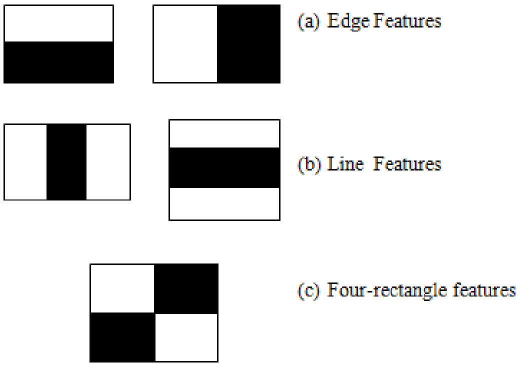

Fig 1: Haar feautres

For every feature calculation, the sum of pixels of white and black rectangles are needed. Here all the features are applied on all the training images [7]. It finds the best threshold for every feature that can be used to classify the images in positive and negative [8]. The main classifier is a sum of many weak classifiers. They are called weak classifiers they alone cannot classify the image but together they can classify. In Cascade of classifiers, instead of applying all the features on a frame, these features related are grouped together and then applied one at a time. If an image fails the first set of features, then discard it and won't considered the remaining features.

OpenCV provides a training method like Haar Cascade model which can be used to detect eyes in an image. First you import OpenCV and then load the required XML files. These XML files contains a set of features that are going to applied on the image. Later function detectMultiScale is used to detect the eyes and returns rectangle for the detected eyes.

Functions used

cv2.CascadeClassifier("XML file name"): used to import Cascade classifier file (XML file).

img.copy(): used to copy an image.

cv2.imread('image_name'): used to extract image.

plt.imshow(variable_name): used to plot the rectangle boundary for the detected eye.

CascadeClassifier.detectMultiScale(): used to detect eyes.

The below images are the end output of this system.

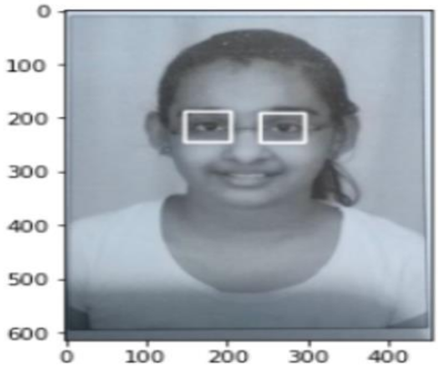

Fig 2: Result 1

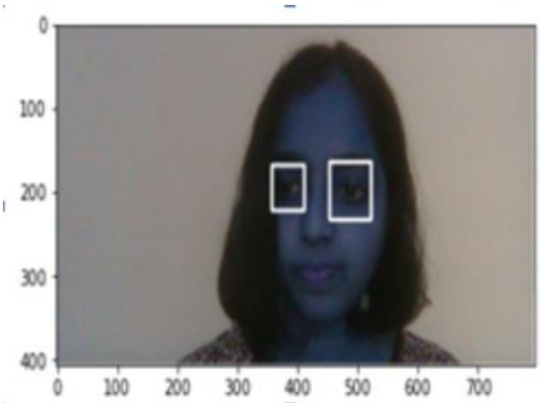

Fig 3: Result 2

4. CONCLUSIONS

In this study you learn to perform eye detection using haar cascade classifier and OpenCV. Haar Cascade classifier is an effective object detection approach, but it can be challenging to accurately specify the parameter value of scaleFactor and the parameter value of minNeighbors of detect Multiscale function. By accurately specifying these parameter values you can effectively detect eyes from a given image.

REFERENCES

[1] K. Vikram; S. Padmavathi, "Facial parts detection using Viola Jones algorithm", 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS), 6-7 Jan. 2017.

[2] Jon pPrris, Michael Wilber, Brian Heflin, Ham Rara, ahmed el-barkouky. aly farag, Javier Movellan, Modesto castrilón-santana, Javier lorenzo-navarro,

Mohammad Nayeem teli, sébastien marcel, Cosmin Atanasoaei, t.e. boult, "face and eye detection on hard datasets", 2011 international joint conference on biometrics (ijcb), 11-13 Oct . 2011.

[3] Peng Wang; M.B. Green, Qiang Ji, J. Wayman, "Automatic Eye Detection and Its Validation", 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops, 21-23 Sept. 2005.

[4] Shifeng Hu; Zuhua Fang; Jie Tang; Hongbing Xu; Ying Sun,"R esearch of Driver Eye Features Detection Algorithm Based on OpenCV", 2010 Second WRI Global Congress on Intelligent Systems, 04 February 2011.

[5] S. El Kaddouhi, A. Saaidi, M. Abarkan, "Eye detection based on the Viola-Jones method and corners points", Springer Science+Business Media New York 2017.

[6] Anjith George, Aurobinda Routray. "Fast and accurate algorithm for eye localisation for gaze tracking in low-resolution images", IET Computer Vision, 2016.

[7] S. S. Kulkarni, A. D. Harale, A. V. Thakur. "Image processing for driver's safety and vehicle control using raspberry Pi and webcam", 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), 2017.

[8] Viknesh Kumar, Boon-Chin Yeo, Way-Soong Lim, Joseph Emerson Ra, Kim-Boon Koh. "Development of Electronic Floor Mat for Fall Detection and Elderly Care", Asian Journal of Scientific Research, 2018.