

Face detection using OpenCV Android - Test distance

Rodolfo Romero Herrera¹, Juan Jesús García Centeno², Gabriela De Jesús López Ruiz³

Profesor Investigador, Departamento de Posgrado, Instituto Politécnico Nacional (IPN) Escuela Superior de Cómputo (ESCOM), México D.F.¹

Estudiante, Programa Delfín, Instituto Tecnológico Superior de Pátzcuaro (ITSPA), Michoacán Mexico¹

Profesor, Departamento de Ingeniería de Software, Instituto Politécnico Nacional (IPN) Escuela Superior de Cómputo (ESCOM), México D.F.³

ABSTRACT: In this article the implementation of face detection method shown by Viola and Jones in mobile devices. The intention is to show the simplicity of the process; but also prove that they are not sufficient for the functions implemented Open Cv, if we want to implement security measures, it is easy to cheat the system. The maximum and minimum distances with the intention to test the system response to these parameters were measured. The midpoint in the distances is when the complete head of a person is detected. These tests discern the intended utility and algorithms cellphone. As a result two graphs were obtained away, and some parameters such as lighting and the use of makeup were determined.

KEYWORDS: Face detection, OpenCV, Android, Recognize.

I. INTRODUCTION

Currently, intended that machine can not only recognize their owners also learn through face detection and other elements of it; to do this we use the OpenCV computer vision library which allows detection of these elements thanks to they use algorithms that work together to face detection [1].

Using OpenCV library with android is the best choice when it comes to face detection because it provides better performance for real-time image processing; many systems can be used as a basis opencv and / or you can improvise a reasonable solution for many low-level functions OpenCV [1].

To recognize the face, eyes and other objects that want to detect of a face, we use the file CascadeClassifier; which serves to draw a box the color you want or any other figure in the face. This file uses two specific functions, which are detectMultiScale, load and the class cascadeclassifier used to detect objects in a video sequence [2].

All models of facial recognition in OpenCV are derived from the abstract base class FaceRecognizer, which provides a unified Access to all face algorithms for detection in OpenCV [3]. Within opencv package version 2.4.9 which is one of the most stable apart from the 2.4.6 can be observed with face-detection; it is the most functionality detection in Android. Support 2 execution modes: default Java wrapper available to the classifier cascade, and manually JNI that call to a native class that supports monitoring. Even the version of Java is able to show in real time the face detection [4] [5].

By using separate opencv face detection can make another type of detections based on computer vision and object detection for feature extraction [6]. The file haarcascade_lefteye_2splits which are coordinates used to detect the elements to be emphasized by a square [7].

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2015

The use of these applications on a mobile device has the advantage that the android operating system is free and allows developers to use libraries that are free to detect the human face gestures and process for that a robot simulate emotions in an economical manner [8].

When importing the project will have to consider some additional steps: download the api which was developed and add NDK library to native C ++ code that can operate the next step, identify and use the correct version of the file haarcascade_lefteye_2splits.xml; which contains the algorithm for face detection [5].

Mobile applications are a market that has not been fully harnessed into what may be in matters of technological development; in this case is due to the use of the native language c ++ which is very heavy and increases the difficulty of the application [9]. From here the necessary amendments are made to detect other elements of the face [10].

II. RELATED WORK

The Viola-Jones algorithm works by finding features. All functions are rectangles. Inside the box are smaller rectangles and then you have pixels. The rectangle is formed in the face in the Viola-Jones algorithm is very small [11].

The Viola-Jones algorithm provides the first face detection in “real time”, this algorithm is composed of 3 elements for rapid and accurate detection which are the integration of the image to calculate the function, the AdaBoost algorithm for selection features and attentional last cascade for efficient calculation [12]. Viola-Jones algorithm training is slow but detection is very fast because each image consists of 10,000 to 50,000 pixels [13].

A technique proposed by Paul Viola and Michael J. Jones was analyzed [14]. This method is basically the face location by using a group of rectangular features, which operates faster than a pixel-based system [14]. The two, three and four rectangles: three kinds of characteristics are used. Said rectangular features (CRs) are 10 different types shown in Figure 1.

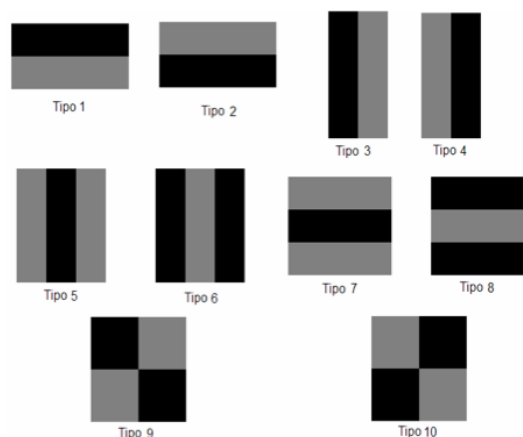


Figure1. Set of ten CRs used in the process of locating faces.

How you use these CRs, it is to place them in a certain position within the image and calculate the difference between the sum of the pixels within the light part and the dark part of the CR, obtaining an integer value that must exceed a to consider that threshold is above the facial feature you want. The training stage is to find the faces in a

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2015

given scale and position within a window of a given size located a characteristic feature of a face, this is called CR classifier. In Figure 2 show a set of classifiers in the classifier where only figure 1d located a feature of the face.

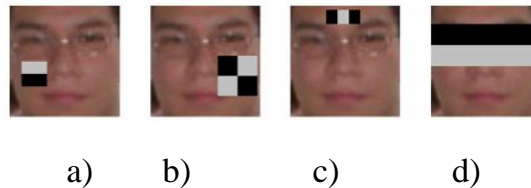


Figure 2. Four rectangular features (AAD) which is a useful feature that can be considered a classifier only d.

The Viola & Jones method is to train or find a set of classifiers, and their thresholds, testing each of the types of classifiers at each position and scale a set of test images using, for this, the learning algorithm AdaBoost[15]. This set consists of a group of images that standardized faces and a group of pictures in which faces not be found; sorters with the largest number of hits in the classification of the faces and the fewest false positives obtained are selected.

For CRs very quickly calculate various scales an intermediate representation of the image (whole image) is used. Once calculated, any CR is calculated at any scale and location in constant time. The integral image in a position x, y contains the sum of the pixels above and to the left of x, y. Equation 1.

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (1)$$

Where $ii(x, y)$ is the integral image and $i(x, y)$ is the original image (Figure 3). Using the following pair of recurrences:

$$s(x, y) = s(x, y-1) + i(x, y) \quad (2)$$

$$ii(x, y) = ii(x-1, y) + s(x, y)$$

Where $s(x, y)$ is the cumulative sum of the values of the pixels of the row, $s(x, -1) = 0$, and $ii(-1, y) = 0$, the integral image is calculated in a single pass over the original image. In figure 3, one can see the point corresponding to the sum of all the pixels located on the left and up. As expressed in the equations (1) and (2).

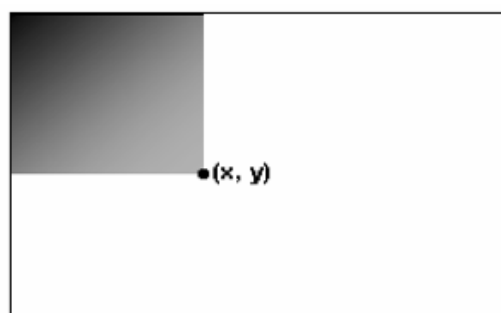


Figure 3. The value of an integral image at a point (x, y) is the sum of all pixels up and left.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2015

Using the integrated image any amount of a rectangle is calculated using an array of four references (see Figure 4). Also the difference between two sums of rectangles is calculated with eight references. Since the characteristics of two rectangles (types 1, 2, 3 and 4) defined above have adjacent rectangular sums is calculated with an arrangement of six references, eight for the characteristics of three rectangles nine for four rectangles.

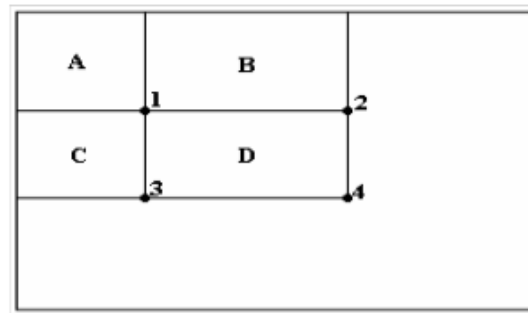


Figure.4-The sum of the pixels within the rectangle D can be calculated with an arrangement of four references. The value of the integral image at position 1 is the sum of the pixels in the rectangle A. The value at position 2 is $A + B$, in position 3 is $A + C$, in position 4 is $A + B + C + D$. The sum within D can be calculated as $4 - 1 - (2 + 3)$.

The face detection example that comes standard in the package opencv which is free and allows various modifications other than face detection also allows us to highlight other elements of the face such as eyes, eyebrows, nose and mouth in the more stable version 2.4.9 android contains several elements and files for operation between these files should be noted that the algorithm used opencv viola-jones.

As it relates to the face detection algorithm viola-jones is extremely fast due to work in conjunction with the AdaBoost algorithm which selects a small number of critical visual characteristics of a large set of potential features [16].

The next element used in this application to face detection is the file lbpcascade; which is responsible for the classification algorithm cascade; this file was trained with some thousands of tests of views on particular objects such as cars or faces. Examples can be colored or negative always the same size; after it was trained and qualified can be applied to the region of interest. This algorithm is designed to be easily resized [17].

Speaking specifically, the files used lbpcascade_frontalface.xml consists of a local binary pattern for face detection [18].

The AdaBoost algorithm is to focus on a set of points, the set of points that need to be qualified with a previous classification algorithm; in this case the algorithm is given the data on which the distribution is created; which is selected weak with the lowest error rate to throw us a rated output signal classification [19].

In the pseudo-code of AdaBoost is trained with examples $(x_1, y_1), \dots, (x_m, y_m)$, where x_i are in some domain X , and labels $y_i \in \{-1, +1\}$. In each round $t = 1, \dots, T$, one D_t distribution is calculated as in Figure 1. Learning algorithm is applied to find a weak hypothesis $h_t: X \rightarrow \{-1, +1\}$, where the goal is to find a weak hypothesis with low weighted error ϵ_t . The final hypothesis H combine signal calculating a weighted combination of weak hypothesis [20].

Finally java file called Detection Based Tracker creates the parameters to be used to track the object. Most of these parameters are the same as those needed to Haar [21].

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2015

This specific file contains several methods which are used for better performance when detecting the face; which are stop, start, setMinFaceSize, detect and Release in particular setMinFaceSize methods and detect; They are those responding to the detection face based on the parameters sent from the lbpcascade_frontalface.xml file [22].

III. EXPERIMENTAL RESULTS

The results based on tests conducted were obtained as shown in Figure 5; where, Men are easier to recognize than women to lesser and greater distance. Also, color of the skin and the use of makeup face affect the detection. Other factors affecting is the light in the room and the visibility of the face by the camera. In Figure 5 the shortest distance is indicated in Table 1 and blue number 20 the maximum distance. This test was performed by placing a computer versus cell with different faces. It can be seen that the best results are obtained at lower distances. When the distance is increased the result is not proportional or exponential. In all cases it care that the face was completely detected.

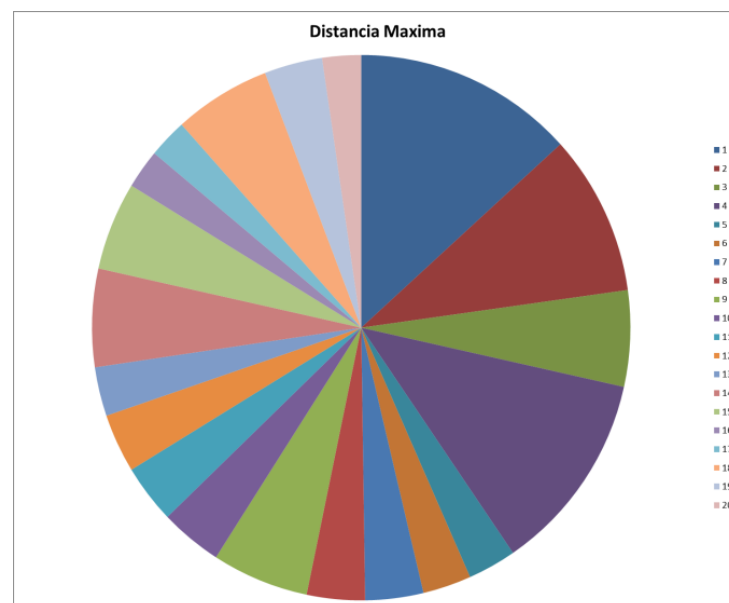


Figure 5: ResultsFace Detectionmaximum distance.

If we now approach the camera cell from such a distance to detect head and full torso of a human and diminish the distance to the maximum limit where it is not possible that the face is captured in full results you will get the figure 6. The best results are obtained when no longer enters the camera capturing the hair of the person; that is, the middle part of the graph between the bars 8 and 14.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2015

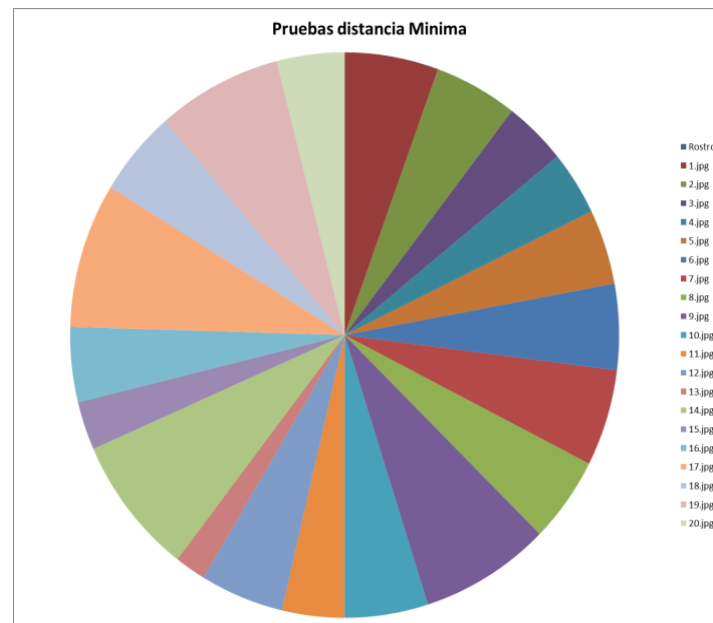


Figure 6: Face Detection Results minimum distance.

The system was tested to detect face images shown in computer and likewise were recognized without problem. However the screen brightness only slightly hinders the recognition.

IV. CONCLUSION

The faces were detected based on the minimum, maximum brightness of a room; an application that can recognize the faces of the people framing it.

The correct detection of a face depends on factors such as distance and the full or partial face detection. You may recognize the pattern of the human face if it is incomplete which does not benefit if we want that the process to be incorporated a biometric security system. Factors such as light or wearing makeup can affect face detection as well as the Owner; therefore, it requires more robust algorithms, if what you want is applied to security systems that use mobile devices. However, the system can be applied with advantage in cases where what is needed is the location or counting faces.

V. ACKNOWLEDGMENT

We thank the ESCOM IPN and for their support for the realization of this research.

REFERENCES

- [1] Author Unknown, OPENCV, "Caching on the World Wide Web", <http://opencv.org/platforms/android/opencv4android-samples.html>, 2014.
- [2] Kévin Letupe; Android application for Face Recognition; Université Lille1; 2013.
- [3] Hošek, R. romanhosek.cz. "Caching on the World Wide Web", [romanhosek.cz: http://romanhosek.cz/android-eye-detection-and-tracking-with-opencv/](http://romanhosek.cz/android-eye-detection-and-tracking-with-opencv/), 2013
- [4] Ferzli, R.; Khalife, I., "Mobile cloud computing educational tool for image/video processing algorithms," in *Digital Signal Processing Workshop and IEEE Signal Processing Education Workshop (DSP/SPE), 2011 IEEE*, vol., no., pp.529-533, 4-7, 2011.
- [5] Vazquez-Fernandez, E.; Garcia-Pardo, H.; Gonzalez-Jimenez, D.; Perez-Freire, L., "Built-in face recognition for smart photo sharing in mobile devices," in *Multimedia and Expo (ICME), 2011 IEEE International Conference on*, vol., no., pp.1-4, 2011.
- [6] Author Unknown, opencv. (s.f.). docs.opencv.org. "Caching on the World Wide Web", de docs.opencv.org: http://docs.opencv.org/2.4.9/modules/imgproc/doc/feature_detection.html, 2015

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2015

- [7] Author Unknown,opencv. (s.f.). docs.opencv.org. "Caching on the World Wide Web", de docs.opencv.org: <http://docs.opencv.org/2.4.9/modules/refman.html>, 2015
- [8] Author Unknown,opencv.org. (s.f.). "Caching on the World Wide Web", opencv.org: <http://opencv.org/platforms/android/opencv4android-samples.html>, 2015.
- [9] Joseph Howse; Android Application Programming with OpenCv; Packt publishing; 2013; UK.
- [10] SalilKapur, NisargThakker; Mastering OpenCV Android Application Programming; Pack publishing; 2015.
- [11] Paul Viola, Face Detector, "Caching on the World Wide Web", <http://www.cs.ubc.ca/~lowe/425/slides/13-ViolaJones.pdf>, 2015
- [12] Paul Viola, Michael J. Jones; Robust Real TimeFace Detection; International Journal of Computer Vision; 2004; Netherlands.
- [13] Borenstein, G. makematics.com. "Caching on the World Wide Web", de makematics.com: <http://makematics.com/research/viola-jones/>, 2015
- [14] Viola, Paul y Jones, Michael "Robust Real-Time Object Detection" En International Conference on Statistical and Computational Theories of Vision – Modeling, Learning, Computing and Sampling, Memorias. Canada, 2001.
- [15] FreundYoav, y Robert E., Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting". En ComputationalLearningTheory: Eurocolt '95, Springer- Verlag.pp. 23–37, 1995
- [16] Emer, E. (s.f.). math.mit.edu. "Caching on the World Wide Web" , de math.mit.edu: <http://math.mit.edu/~rothvoss/18.304.3PM/Presentations/1-Eric-Boosting304FinalRpdf.pdf>, 2015
- [17] Author Unknown,fossies.org. (s.f.). "Caching on the World Wide Web", de fossies.org: http://fossies.org/dox/opencv-3.0.0/classorg_1_1opencv_1_1samples_1_1facedetect_1_1DetectionBasedTracker.html, 2015.
- [18] Author Unknown,opencv. (s.f.). docs.opencv.org. "Caching on the World Wide Web", de docs.opencv.org: http://docs.opencv.org/modules/objdetect/doc/cascade_classification.html?highlight=cascadeclassifier#cascadeclassifier, 2015.
- [19] Ranjanritesh. (08 de agosto de 2012). bytesandlogics.wordpress.com. "Caching on the World Wide Web", de bytesandlogics.wordpress.com: <https://bytesandlogics.wordpress.com/2012/08/23/detectionbasedtracker-opencv-implementation/>, 2015
- [20] Schapire, R. E. (s.f.); ExplainingAdaBoost; Princeton; 2015.
- [21] Stackoverflow. (2014). stackoverflow.com. "Caching on the World Wide Web", de stackoverflow.com: <http://stackoverflow.com/questions/17904544/explanation-about-cascade-xml-file-lbpcascade-frontalface-xml>, 2015
- [22] Wang, Y.-Q. "Caching on the World Wide Web", de www.ipol.im: <HTTP://WWW.IPOL.IM/PUB/ART/2014/104/ARTICLE.PDF>, 2014