FACE RECOGNITION USING CASCADE ALGORITHM

Vasavi Ravuri, S.Vasundra, A.Aryan, Somula Ramasubbareddy, K.Govinda

Abstract*: As one of the most successful application of Image processing, face recognition has received attention for quite a long time. Whether being used in CCTV cameras or home security, face recognition is an important application in today's metropolitan era. The face recognization strategy is implemented by using Haar cascade algorithm. Both are used differently and the results are then compared to know which one works better or is more accurate. In this paper, we have used Python programing language, since this is only a basic Face recognition system there is no database used but for future extension, we can add a database to expand this method to a larger scale.

Keywords: Deep Leaning, Face Recognization, Classifiers, Cascade.

I. INTRODUCTION

With the expanse of scale and sphere of human activities triggered by globalisation, face recognition i.e. the process to identify human faces and use their characteristics for further purpose, is being used extensively in the global economy. The framework for image recognition uses biometrics to outline highlights from a photo or video and stores it as data. It brings together this particular data and the dataset of already identified and characterized appearances to put forward a new match. Face recognition can help check individual character, yet it likewise raises protection issues.

High defination cameras in mobile phones have settled on Face recognition a plausible decision for verification as distinguishing proof. For eg, the iPhone x has included their new Face id technology helping clients to open the phone with a faceprint scanned by its front camera. Their source code for face recognition examines more than 30000 elements, which is further layered with 3D demonstrations to prevent being caricatured by photos or covers. However, Face IDs can further be implemented to validated purchases from cross applications in Apple Ecosystem for example iTunes, Appstore and iBooks Store. Apple encrypts and all the data for faceprint is stored in its cloud, yet the validation process happens directly on the device.

Amazon recognition can we used by planners which is form the Amazon AI suite, to incorporate and to examine the features to an application. Google provides a similar ability by using Google cloud vision API. This technology which is using neural networks is further being used in a variety of ways, for example amusement and exhibiting. If we take the example of the Kinect development gaming system it will implement this technology to differentiate among different players. Smart ads in air terminals and in social media are catered to take into account the persons features like sex, ethnicity, age, etc and gives solutions/ads on the basis of these facts.

Revised Manuscript Received on July 05, 2019.

Vasavi Ravuri, Asst. Prof, VNRVJIET, Hyderabad, Telangana,
Vasundra.S, Professor, JNTUA College of engineering, Ananthapuramu, Andhra Pradesh.
Aryan, Scope School, VIT University, vellore, hansy.aryan@gmail.com
Somula Ramasubbareddy, Asst. Prof, VNRVJIET, Hyderabad, Telangana, svramasubbareddy1219@gmail.com
K.Govinda, Associate Professor, VIT University, Vellore, Tamilnadu, kgovinda@vit.ac.in
Corresponding Mail: svramasubbareddy1219@gmail.com.

2. PROBLEM STATEMENT

The main purpose is to recognize or identify faces and track them by giving them a particular id number or name. This can be done by either giving a new id to every time a face is recognized by the software or by creating a folder or a database where the faces trained by the software are stored, in other words a database.

3. LITERATURE REVIEW

There have a lot of work and research done regarding face recognition. Various algorithms are used and are applied on detecting faces so that a better than ever face recognition software can be created. Existing software's have either modified or upgraded these algorithms or have combined the concept of many to create a better and more accurate one.

This paper addresses the non-minor issue of execution assessment of object tracking. The author proposes a rich arrangement of measurements to evaluate diverse parts of the execution of object tracking. The paper utilizes six distinctive video successions that speak to an assortment of difficulties to outline the functional estimation of the proposed measurements by assessing and comparing two object tracking algorithm.[2]

A centroid tracker has already been created and has been kept for implementation as an arrangement of laser beams in the Aurora Laser Fusion project which is at at Los Alamos National Laboratory. It entails the foundations as one of the 3 national inertial confinement fusion energy experiments. This pillar control framework will further be used to achieve angular beams with highest amount of accuracy control of optical mirrors.[3]

In a paper by Ronda Venkateswarlu and KV Sujata have discussed about the Autonomous fire and overlook weapons have picked up significance to accomplish exact first pass execute by hitting the object at a suitable point.

Centroid of the picture exhibited by an object in the field of view (FOV) of a sensor is commonly acknowledged as the aimpoint for these weapons. Centroid trackers are pertinent just when the objective picture is of critical size in the FOV of the sensor yet does not flood the FOV. In any case, as the range between the sensor and the object reduce, the picture of the objective will develop lastly overflow the FOV at short proximities and the centroid point on the objective will continue changing which isn't attractive. And furthermore centroid need not be the most wanted/helpless point on the object. For solidified targets like tanks, appropriate aimpoint determination and direction up to right around zero territory is basic to accomplish maximum kill probability.[4]

Understanding the movement of object in a video rises an interest due to its increased automation in CCTV, controlling control and also the analysis of movements of pedestrians. In this paper, we are presenting a system which is further used for detection and differentiation of people and vehicles in a variety of different weather conditions upon using a static camera. Aside from this, it is also capable for tracking of more than one object despite object interaction. The outcome of this is then presented by an online application of the algorithm.[5]

The author lazyoracle, created a motion detector which records timestamps every time it detects a motion in the frame. But since even a little bit of motion is detected and recorded, it is not the most appropriate motion detector in many cases. [6]

The author Adrian rosebrock uses OpenCV library to create a motion detector with very high accuracy meaning that it won't detect the motion of small things, this was done by giving a threshold to the frame so that only things with significant changes in the pixel intensity values are detected. The program also shows the status of the room whether it is occupied or unoccupied when it detects motion.[7]

The accuracy of tracking objects depends on not only the program code but also the speed with which the object is moving. Adrian Rosebrock created a program that uses different algorithms in OpenCV to detect motion. Different object tracking implementations are used according to the speed and FPS throughput. The user can also select the object the they want to track the path of by drawing a bounding box around that object [8].

In this paper, a program is presented which can detect and identify human faces and works in real-time. The first step is to track the user's head and then it recognises the person upon comparing features of the face to the larger dataset of all individuals. Thus the face recognition problem is viewed by the computer as 2-D recognition problem. A feature space then takes those images which then encode the differences among the dataset of all the known images. This feature space is recognised as 'Eigen faces' having the eigen vector of the set of faces. The limitation is on the fact that isolated features like nose, ears, eyes, etc are not taken into account.[9]

Viola and Jones acquainted a technique to precisely and quickly distinguish faces inside a picture. This system can be adjusted to distinguish facial features. the zone of the picture being investigated for a facial component should be regionalized to the area with the most noteworthy likelihood of containing the element. By regionalizing identification region, false positives are dispensed and the speed of location is expanded because of lessen territory inspected. [10]

4. PROPOSED METHOD

The proposed method is divided into 2 parts

1) Face recognition with deep learning.

2) Face recognition using haar cascade

We will first understand how facial recognition with deep learning works:

For detecting an object in general in the scope of this project we would be using centroid tracking. This will work on the principle of calculating the Euclidean distance between

(a) Objects that the centroid tracker has already detected before

(b) The new emerging and to be calculated centroids between adjacent frames in a particular video

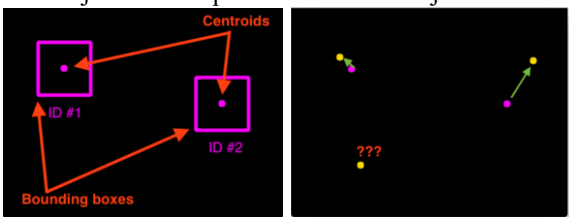
The centroid algorithm would be embedded in a python class, then have a .py script to run the program and finally have its applications to input datasets

For every distinct object in each and every casing when we would be using the centroid tracking algorithm, we would be having an arrangement of bounding box (x,y) coordinates.

The arrangement is created by an object locator of user's choice (for eg, Haar falls, Linear SVM and HOG's combined, Single Shot Detections, Faster R-CNNs, contour extraction with the help of colour thresholding, etc) which gives on the fact that they are figured to each frame in the video. After obtaining the bounding box, we should calculate the "centroid" (x,y) of the box.

For each resulting casing in the video stream we add the previous step of figuring object centroids; be that as it may, rather than relegating another one of a kind id to every distinguished question (which would in turn nullify the point of object tracking), we have to find a way for the connection of the current object centroid and the previous' frame centroid. To achieve this, the processing of Euclidean separation between each combination of now existing item centroids and the new input object centroids happens which in turn will tell us the main object or in this case the face of the individual.

The main assumption of centroid tracking algorithm is for two continuous frames Fx and Fx+1 where x is time, the centroid of the main object will move between this time whereas, the distance for all the other objects will be much smaller. So, the object tracker is created when we connect the centroid with distance between the frames being minimized for other objects as compared to the main object.



(a.) (b.)

Fig. 1 How Centroid tracking works

If a case arises that there are multiple input detections need to be tracked aside from the main object, we would have to register those objects in an order ie the new object will also be added to the list of all the objects being tracked. This would be done by-

o The object would be assigned a new ID

o The centroid of coordinates of box of the object would also be stored

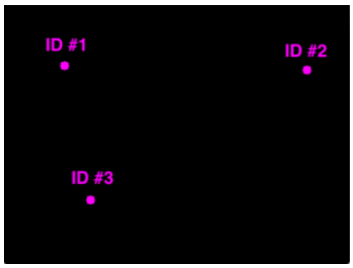


Fig. 2. Giving new ID to every new face recognized

To use this object tracker specifically to detect the face we have used two files under the project folder - .prototxt and .caffemodel[1] which come under the library of OpenCV deep learning algorithms for the purpose of face recognition. However, we can also use another form of detection.

Now we'll see how the haar cascade algorithm, a machine learning based algorithm works for face recognition:

Haar Cascade algorithm works when the training set has a cascade function which in turn is used to classify images as positive or negative. The algorithm further detects finer details in the images. In mathematical topic, cascade algorithm is used for calculating function value of basic scaling using an iterative algorithm.

The iterative algorithm creates successive approximation to ψ(t) or φ(t) from {h} and {g} filter coefficients.

The iterations are defined by:

$$\varphi^{(k+1)}(t) = \sum_{n=0}^{N-1} h[n]\sqrt{2}\varphi^{(k)}(2t-n)$$

For the kth iteration, where an initial φ(0)(t) must be given.

$$\Phi^{(k+1)}(\omega) = \frac{1}{\sqrt{2}} H\left(\frac{\omega}{2}\right) \Phi^{(k)}\left(\frac{\omega}{2}\right)$$

Wavelet function is given as:

$$\psi(t) = \sum_{n=-\infty}^{\infty} g[n]\sqrt{2}\varphi^{(k)}(2t-n).$$

The algorithm in our project goes through 3 stages. Stage 1 is first entering an ID number and then capturing the face and storing them in a dataset, around 20 positive and negative images are captured. Stage 2 is then training the algorithm to detect the face, this is done by detecting the basic facial features. And the final stage is detecting the actual face in real time. This entire process can either be done by creating a separate database and storing the images in it with the ID number and then linking the entire database to the program or by creating a new statement every time a new ID or face is detected.

5. RESULTS

After running both the algorithms Below is the output obtained from both

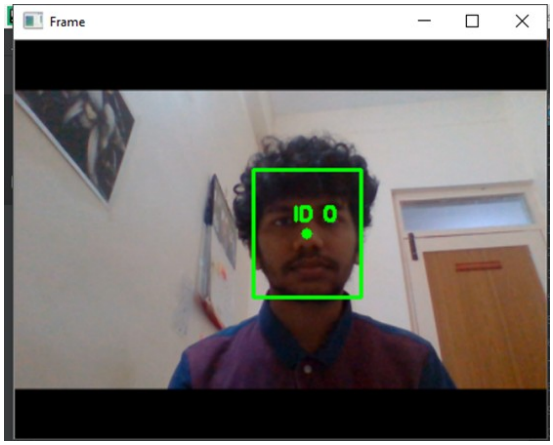


Fig. 3. Face recognition using deep learning

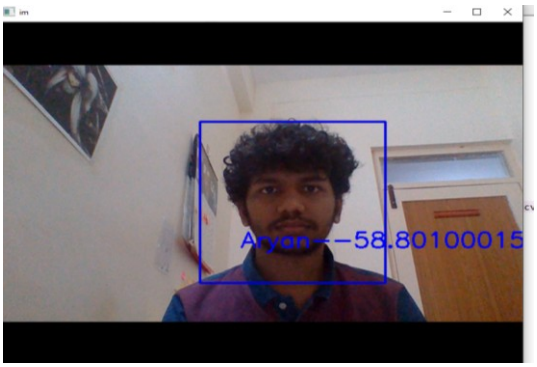


Fig. 4. Face recognition using Haar cascade

In both the algorithms, the instant it starts the face is detected and both are very accurate although some environmental factors such as bad lightning might have an impact on the algorithm when detecting the face.

In the deep learning one after the face is detected an ID is given to it. The object tracker can be used to detect more than one face. However, if an object is removed from the frame the tracker will remain till the object has existed, if the object is outside the scope of viewing tracker for more than 50 frames, then that particular object will be de-registered.

On the other hand, in case of haar cascade algorithm after the face is detected the name of the person detected and the confidence level is shown on the screen. In this case also multiple faces can be detected.

5.1 COMPARATIVE STUDY

Below is a comparative study of haar cascade algorithm depicted through a graph:

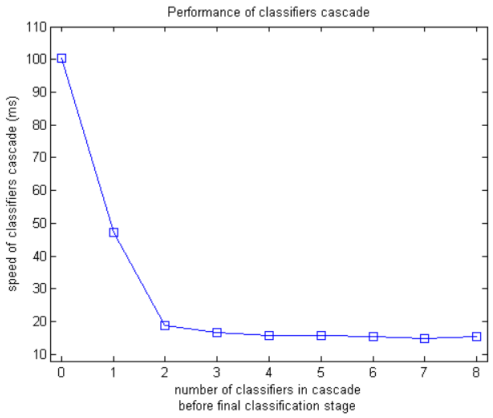


Fig. 5. A graph between speed and number of classifiers showing the performance of classifier.

The graph shows that as the speed of classification of cascade decreases the number of classifiers in cascade increases meaning that the slower the speed more the information is passed on in groups before it reaches final classification stage resulting in better accuracy while detecting faces.

Another one is a comparison of different deep learning classifiers for their accuracy:

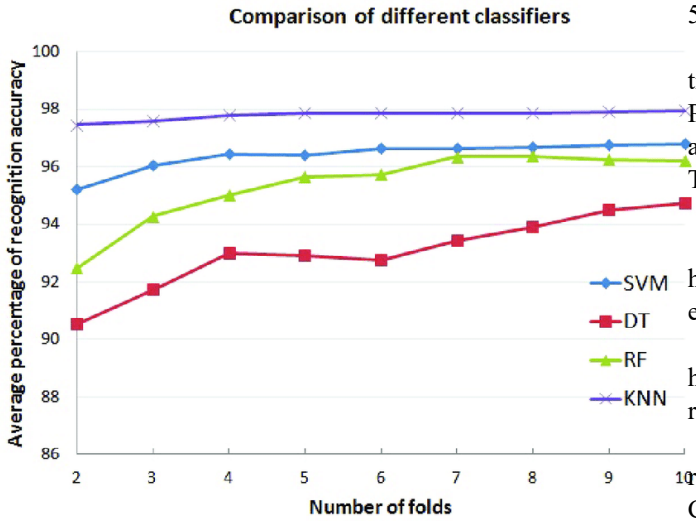


Fig. 6. Graph between accuracy percentage and number of folds depicting the comparison of different classifiers.

SVM, DT, RF and KNN are part of machine learning and deep learning algorithms. The graph shows how accurate each algorithm is on the basis of the folds they have. Holds are basically the number of packets the data can be divided into. Each fold is individually used for the validation set and the remaining part is used for the training set

6. CONCLUSION

Both algorithms work in a similar manner they are meant to and are very accurate excluding the factor of bad lightning. The drawback for centroid tracking algorithm (deep learning) is that there are high chances of ID's being overlapped if the objects are very close by or one is standing behind the other, since it only detects the face and does not store each face it detects.

On the other hand, in case of haar cascade, lightning plays a very big role since because of bad lightning the program can show false data of the face it detects, however it is very reliable since it remembers every face by a different ID and does not give it a new one every time a face arrives.

Both the algorithms are good in different ways and their performance depends on how they are applied in an application, while centroid tracking (deep learning) focuses more on detecting the path which is followed by the object, the haar cascade focuses on detecting each face and its confidence level.

7. REFERENCES

[1] https://github.com/soeaver/caffe-model

[2] Yin, F., Makris, D., & Velastin, S. A. (2007, October). Performance evaluation of object tracking algorithms. In IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, Rio De Janeiro, Brazil (p. 25).

[3] Anderson, F. (1990, September). Real time, video image centroid tracker. In Acquisition, Tracking, and Pointing IV (Vol. 1304, p. 82). International Society for Optics and Photonics.

[4] Venkateswarlu, R., Sujata, K. V., & Rao, B. V. (1992, November). Centroid tracker and aimpoint selection. In Acquisition, Tracking, and Pointing VI (Vol. 1697, pp. 520-530). International Society for Optics and Photonics.

[5] Johnsen, S., & Tews, A. (2009, May). Real-time object tracking and classification using a static camera. In Proceedings of IEEE International Conference on Robotics and Automation, workshop on People Detection and Tracking. ERENCES:

[6] https://github.com/lazyoracle/motion-detection

[7] https://www.pyimagesearch.com/2015/05/25/basic-motion-detection-and-tracking-with-python-and-opencv/

[8] https://www.pyimagesearch.com/2018/07/30/opencv-object-tracking/

[9] Turk, M. A., & Pentland, A. P. (1991, June). Face recognition using eigenfaces. In Proceedings 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (pp. 586-591). IEEE.

[10] Wilson, P. I., & Fernandez, J. (2006). Facial feature detection using Haar classifiers. Journal of Computing Sciences in Colleges, 21(4), 127-133.