International Conference on Intelligent Technologies & Science - 2021
(ICITS-2021)

LBPH based Enhanced Technique For Multiple Face Detection

Neeraj Saxena[1], Shivam Shukla[2]
Computer Science and Engineering,
Goel Institute of Technology & Management, Lucknow, India
neesaxena@gmail.com, er.shivamshukla@goel.edu.in

Abstract: In today's technology dependent era the technology has changed the world in and around us. Nowadays each and every field is using the technology to make the work easy and fast. Each and every technology is trying to use the machine learning, deep learning based models to automate the process. The AI has been developed by using the various machine learning and deep learning based models. The AI based products majorly depends on the text input and output the text is processed by using the field of natural learning process. In this paper we will develop multiple face recognition based presence monitoring system using the LBPH based face recognition algorithm. We also have other models for same purpose like Eigen face recognition, fisher face recognition models. The LBPH uses local binary histograms for the detection of the multiple face in a image. all the models are trained by using the face base dataset and then the perdition accuracy result is used to measure the best algorithm.

Keywords: Face recognition, feature extraction, OpenCV, Local Binary Pattern Histogram (LBPH)

1. Introduction:

The traditional method for taking attendance was laborious and time taking in case of crowded class rooms. The manual attendance is expensive, but due to new handy software's the records are preserved and by using advance tools we can generate the report of each student. To save the time taken by manual attendance system previously many real time camera based attendance system were proposed but due to the low prediction accuracy the models were not considered by the institutions. The previous models contained different machine learning and neural network matching models. The geometric feature matching method depends on the calculation of a set of geometric features of the facial image. The general specification is defined by a vector showing the position and size of the main facial features, for example, the eyebrows, nose, mouth and facial contours. SIFT based feature matching uses a multiple layers of Gaussian filter on the images as a part of preprocessing and then uses difference of Gaussians and the image pyramid. SURF feature matching is better than SIFT as it firstly recognizes the important local features and then matches with the fitted data set. CNN is better than the previous models as it is an excellent mathematical tool for complex calculations especially in 2D images. The neural network is so much simplified face recognition approach because of its non-linear architecture in the net system. Therefore, the features extraction phase is more effective than the linear technique, it selects a dimensionality reducing linear projection that increases the scatter of all expected models.

Face recognition is a non-invasive identification system and faster than other systems since multiple faces can be analyzed at the same time. The difference between face detection and identification is, face detection is to identify a face from an image and locate the face. Face recognition is making the decision "whose face is it?" using an image database. In this project both are accomplished using different techniques and are described below. The report begins with a brief history of face recognition. This is followed by the explanation of HAAR-cascades, Eigenface, Fisherface and Local binary pattern histogram (LBPH) algorithms. Next, the methodology and the results of the project are described.

2. Methodology:

Computer vision is the transformation of data from a still or video camera into either a decision or a new representation. All such transformations are done for achieving some particular goal. The input data may include some contextual information such as "the camera is mounted in a car" or "laser range finder indicates an object is 1 meter away". The decision might be "there is a person in this scene" or "there are 14 tumor cells on this slide". A new representation might mean turning a color image into a grayscale image or removing camera motion from an image sequence. Because we are such visual creatures, it is easy to be fooled into thinking that computer vision tasks are easy. How hard can it be to find, say, a car when you are staring at it in an image? Your initial intuitions can be quite misleading. The human brain divides the vision signal into many channels that stream different kinds of information into your brain. Your brain has an attention system that identifies, in a task-dependent way, important parts of an image to examine while suppressing examination of other areas. There is massive feedback in the visual stream that is, as yet, little understood. There are widespread associative inputs from muscle control sensors and all of the other senses that allow the brain to draw on cross-associations made from years of living in the world. The feedback loops in the brain go back to all stages of processing including the hardware sensors themselves (the eyes), which mechanically control lighting via the iris and tune the reception on the surface of the retina. In a machine vision system, however, a computer receives a grid of numbers from the camera or from disk, and that's it. For the most part, there's no built-in pattern recognition, no automatic control of focus and aperture, no cross-associations with years of experience. For the most part, vision systems are still fairly naïve. Figure 1-1 shows a picture of an automobile. In that picture we see a side mirror on the driver's side of the car. What the computer "sees" is just a grid of numbers. Any given number within that grid has a rather large noise component and so by itself gives us little information, but this grid of numbers is all the computer "sees". Our task then becomes to turn this noisy grid of numbers into the perception: "side mirror". Figure 1-2 gives some more insight into why computer vision is so hard.

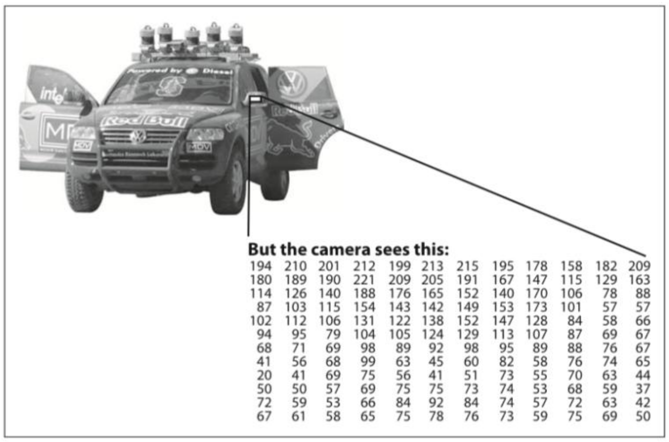

Fig 1: To a computer, the car's side mirror is just a grid of numbers

OpenCV grew out of an Intel Research initiative to advance CPU-intensive applications. Toward this end, Intel launched many projects including real-time ray tracing and 3D display walls. One of the authors working for Intel at that time was visiting universities and noticed that some top university groups, such as the MIT Media Lab, had well developed and internally open computer vision infrastructures code that was passed from student to student and that gave each new student a valuable head start in developing his or her own vision application. Instead of reinventing the basic functions from scratch, a new student could begin by building on top of what came before. Thus, OpenCV was conceived as a way to make computer vision infrastructure universally available. With the aid of Intel's Performance Library Team,* OpenCV started with a core of implemented code and algorithmic specifications being sent to members of Intel's Russian library team. This is the "where" of OpenCV: it started in Intel's research lab with collaboration from the Software Performance Libraries group together with implementation and optimization expertise in Russia. Chief among the Russian team members was Vadim Pisarevsky, who managed, coded, and optimized much of OpenCV and who is still at the center of much of the OpenCV eff ort. Along with him, Victor Eruhimov helped develop the early infrastructure, and Valery Kuriakin managed the Russian lab and greatly supported the eff ort. There were several goals for OpenCV at the outset:

• Advance vision research by providing not only open but also optimized code for basic vision infrastructure. No more reinventing the wheel.
• Disseminate vision knowledge by providing a common infrastructure that developers could build on, so that code would be more readily readable and transferable.
• Advance vision-based commercial applications by making portable, performance optimized code available for free with a license that did not require commercial applications to be open or free themselves.

3. Face Detection using Haar cascades

A Haar wavelet is a mathematical fiction that produces square-shaped waves with a beginning and an end and used to create box shaped patterns to recognize signals with sudden transformations. An example is shown in figure 10. By combining several wavelets, a cascade can be created that can identify edges, lines and circles with different color intensities. These sets are used in Viola Jones face detection technique in 2001 and since then more patterns are introduced [10] for object detection as shown in figure 10. To analyze an image using Haar cascades, a scale is selected smaller than the target image. It is then placed on the image, and the average of the values of pixels in each section is taken. If the difference between two values pass a given threshold, it is considered a match. Face detection on a human face is performed by matching a combination of different Haar-like-features. For example, forehead, eyebrows and eyes contrast as well as the nose with eyes as shown below in figure a single classifier is not accurate enough. Several classifiers are combined as to provide an accurate face detection system as shown in the block diagram below in figure 2.

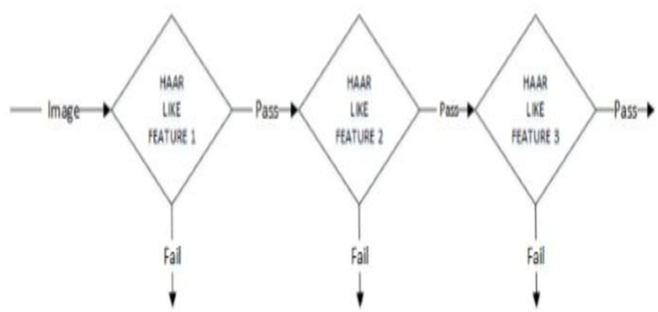

Fig 2: Haar-cascade flow chart

4. Local Binary Pattern Histogram

Local binary patterns were proposed as classifiers in computer vision and in 1990 By Li Wang [4]. The combination of LBP with histogram oriented gradients was introduced in 2009 that increased its performance in certain datasets [5]. For feature encoding, the image is divided into cells (4 x 4 pixels). Using a clockwise or counter-clockwise direction surrounding pixel values are compared with the central as shown in figure 3. The value of intensity or luminosity of each neighbor is compared with the center pixel. Depending if the difference is higher or lower than 0, a 1 or a 0 is assigned to the location. The result provides an 8-bit value to the cell. The advantage of this technique is even if the luminosity of the image is changed as in figure 4, the result is the same as before. Histograms are used in larger cells to find the frequency of occurrences of values making process faster. By analyzing the results in the cell, edges can be detected as the values change. By computing the values of all cells and concatenating the histograms, feature vectors can be obtained. Images can be classified by processing with an ID attached. Input images are classified using the same process and compared with the dataset and distance is obtained. By setting up a threshold, it can be identified if it is a known or unknown face. Eigenface and Fisherface compute the dominant features of the whole training set while LBPH analyse them individually.

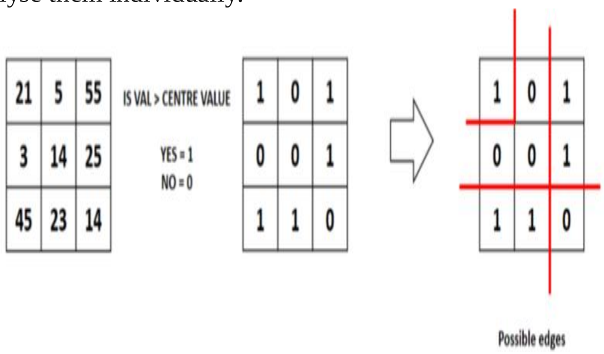

Fig 3: Local binary pattern histogram generating 8-bit number

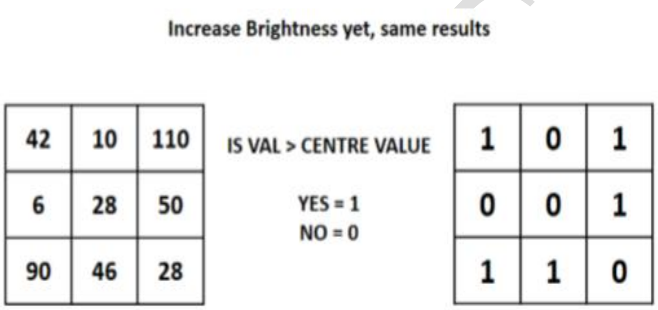

Fig 4: The results are same even if brightness is changed

5. Proposed Work:

Collecting classification images is usually done manually using a photo editing software to crop and resize photos. Furthermore, PCA and LDA requires the same number of pixels in all the images for the correct operation. This time consuming and a laborious task is automated through an application to collect 50 images with different expressions. The application detects suitable expressions between 300ms, straightens any existing tilt and save them. The Flow chart for the application is shown in figure 5.

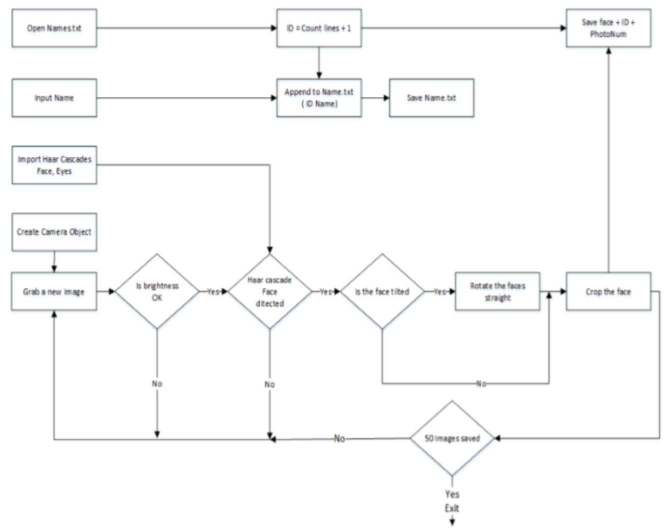

Fig 5. The Flowchart for the image collection

Application starts with a request for a name to be entered to be stored with the ID in a text file. The face detection system starts the first half. However, before the capturing begins, the application check for the brightness levels and will capture only if the face is well illuminated. Furthermore, after the face is detected, the position of the eyes are analyzed. If the head is tilted, the application automatically corrects the orientation. These two additions were made considering the requirements for Eigenface algorithm. The Image is then cropped and saved using the ID as a filename to be identified later. A loop runs this program until 50 viable images are collected from the person. This application made data collection efficient.

The proposed method includes two steps. Generating the dataset and applying the matching algorithm application for matching the face query set with the face base dataset.

The face base data set generation includes collection of data and processing the image by using various filters and extracting the face from all the images by using Haar cascade classifier which is based on the Adaboost algorithm. The extracted face data is converted to histogram and the face, information and the histogram discerption is stored in the dataset.

6. Result and Discussion:

The proposed model is implemented using the python language. Python is a dynamically typed interpreted high level programming language. Python is a trending language now a days. For implementation of the work we use python based library like numpy, opencv,

Tkinter, matplotlib, jupyter notebook. Opencv is a c++ based library for image processing which provides an image API. We have used face base dataset to implement the face recognition technique. The haar-cascade based classifier is used to extract the face out of the images in the face base dataset. We have used the frontal face haar cascade file to train the cascade classifier model. Then the face detected is used to train the different models like LBPH model, Eigen face model, fisher face models. The trained models are used to detect the number of faces in the images and we can clearly see that the LBPH models performance is better as compared to the other available models. The accuracy obtained by each model is as shown in the figure 6. For plotting the graph we have used the matplotlib library.

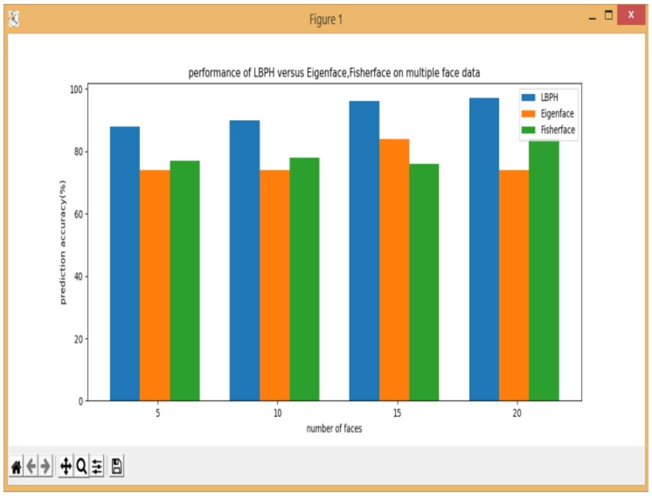

Fig 6: the results of the face detection models

In the figure 7 a test data is predicted to show the multiple faces detected in the image .it can be clearly seen that the students faces with the frontal view is easily detected. but if the students don't look towards the camera then the face is not detected so we have to capture images of students with the frontal view.

Fig 7: the test data with faces detected in the class room.

By using the LBPH and haar cascade based face detection and face reorganization system we have developed a automated student attendance system. the automate attendance system has been divided into three different modules. The first module include the enrollment of the student for which we have designed the user interface which takes all the details of the students. And open the camera and takes 30 snaps of the students with the frontal face. The face is extracted from the images by using the cascade classifier which is trained by using the frontal face haar cascade file.

The faces along with the student enrollment id is trained to the model for which we have used the LBPH model. The trained model is saved in yml format. The student details are stored by using the sqlite3 database engine which is already available in the python environment. The data base client and server both are already installed in the python environment.

The user interface is shown in the figure 8. For building the user interface we have used the tkinter library. The validation of the input is done by using the regular expressions. The images are captured are in the BGR color combinations which is converted to gray scale. The machine learning is performed on the gray scale image. To rectify the image the processing of the images are done which includes the improving the light effect on the images by using the equalized histograms. Then the adaptive threshold is used to process the background of the image by using the Gaussian mean kernel and the image is converted to binary image. Then the image can be used for the training the model.

Fig 8. Student attendance enrollment page

7. Conclusion:

In our proposed work we have designed a LBPH based multiple face detection based automated attendance system. The results clearly shows that the LBPH model is better compared to Eigen face detection model and the Fisher face detection model on the test with multiple faces. Nowadays with the increasing use of the technology in day today life it is possible to manage simple daily works with the machines which reduces the human effort and the time consumed to perform the task. The traditional method of taking attendance consumed much time hence we have used the LBPH based model to build a automated attendance system with the multiple face detection feature. Which would minimize time taken for taking the attendance it would easily generate e-attendance of the entire class room which could be used for further processing of the attendance of individual students. The results could be easily shared with the parents and the faculty so that a proper monitoring of students could be done.

References:

[1]. Takeo Kanade. Computer recognition of human faces, volume 47. Birkh¨auser Basel, 1977.

[2]. Lawrence Sirovich and Michael Kirby. Low-dimensional procedure for the characterization of human faces. Josa a, 4(3):519–524, 1987.

[3]. M. Turk and A. Pentland. Eigenfaces for recognition. Journal of Cognitive Neuroscience, 3(1):71–86, Jan 1991.

[4]. Dong chen He and Li Wang. Texture unit, texture spectrum, and texture analysis. IEEE Transactions on Geoscience and Remote Sensing, 28(4):509–512, Jul 1990.

[5]. X. Wang, T. X. Han, and S. Yan. An hog-lbp human detector with partial occlusion handling. In 2009 IEEE 12th International Conference on Computer Vision, pages 32–39, Sept 2009.

[6]. P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7):711–720, Jul 1997.

[7]. P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, volume 1, pages I–511–I–518 vol.1, 2001.

[8]. John G Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. JOSA A, 2(7):1160–1169, 1985.

[9]. S Marˆcelja. Mathematical description of the responses of simple cortical cells. JOSA, 70(11):1297– 1300, 1980.

[10]. T. Kanade, ―Picture processing by computer complex and recognition of human faces, ―technical report, Dept. Information Science, Kyoto Univ., 1973

[11]. I.J. Cox, J. Ghosn, and P.N. Yianios, ―Feature Based face recognition using mixture distance,‖ Computer Vision and Pattern Recognition, 1996

[12]. B.S. Manjunath, R. Chellappa, and C. von der Malsburg, ―A Feature based approach to face recognition,‖ Proc. IEEE CS Conf. Computer Vision and Pattern Recognition, pp. 373-378,1992

[13]. P. Suja and S. Tripathi, ―Analysis of emotion recognition from facial expressions using spatial and transform domain methods,‖ International Journal of Advanced Intelligence Paradigms, vol. 7, pp. 57–73, 2015.

[14]. M. Lades, J.C. Vorbruggen, J. Buhmann, J.Lange, C. Von Der Malsburg, R.P. Wurtz, and M. Konen, ―Distortion Invariant object recognition in the dynamic link architecture,‖ IEEE Trans. Computers, vol. 42, pp. 300-311,1993.

[15]. Dharejo FA, Jatoi MA, Hao Z, Tunio MA. PCA based improved face recognition system. Frontiers in Artificial Intelligence and Applications. 2017. https://doi.org/10.3233/978-1-61499-785-6-429

[16]. KIRBY, M. AND SIROVICH, L.. Application of the Karhunen-Loeve procedure for the characterization of human faces. IEEE Trans. Patt. Anal. Mach. Intell. 12 ,1990.

[17]. E. Paul and A S Ajeena Beegom, "Mining images for image annotation using SURF detection technique," IEEE International Conference on Control Communication & Computing India, Trivandrum, 2015, pp.724728.