Improved Real-time Multiple Face Detection and Recognition from Multiple Angles

Preeja Priji[1], Rashmi S Nair[2]

Mtech Student, Department of Computer Science, Mohandas College of Engineering, and Technology, Anad, Trivandrum India [1]

Asst Prof, Department of Computer Science, Mohandas College of Engineering, and Technology, Anad, Trivandrum, India[2]

Abstract— The proposed thesis is a real-time multiple face recognition system from an uncontrolled domain such as a surveillance camera or drone camera. The traditional methods make use of edge detection for face detection and pixel matching for recognition. These methods produce a lot of false positives. For accurate detection, Artificial neural networks are used. These techniques require more resources and is slower since convolutions has to be applied to every pixel subset. The proposed method uses a combination of Viola Jones algorithm to detect faces and Convolutional Neural network to refine the detected faces by removing false positives. The method is more accurate than traditional methods and faster than Artificial Neural Networks since only the detected subset is subjected to ANN. The ROI is automatically detected for all faces Each detected face is then recognized using eigen face recognition method. The proposed method can also learn similar faces automatically by referring previous frames. Each time a face is not recognized by the system in an angle yet it is known in other angles, the system auto learns the face to be the frequently recognized face. The system is built on Emgu CV and C# which makes it highly reliable and faster in Microsoft Platforms.

Index Terms— *Haar face detection, multiple faces, real-time face detection multiple angles, eigen face recognition, self learning.*

1 INTRODUCTION

Image Processing is a very important field comprising thousands of useful and valuable applications such as face detection, recognition and processing. It is easy to determine a face if the individual is sitting at a normal frontal angle, but it is difficult to detect face from different angles. The haar cascading uses cascading of classifier trained by Adaboost method. The haar cascading method is faster when compared with other neural networking detection methods. Hence it can be used in real-time detection.

The goal in face detection is to identify and extract faces visible in an image [1]. Reliable face detection is one of the most studied research topics in the field of computer vision and precursor to face identification or matching. Haar Cascade classifier method is the commonly used method for face detection and detection of other objects. Haar Cascade XML contains the values for possible faces. Frontal xml contains values for frontal face and profile face file contains profile faces values form left side.

The problem of haar cascading is the increased number of false positives in uncontrolled scenario. This can be rectified by further refining the detected faces by the Convolutional Neural Networks (CNN).

In [2] the files from Face Detection Data Set and Benchmark (FDDB) dataset designed for studying the problem of unconstrained face detection were used and got very high accuracy whereas with Open-I database, the result contained many false positives since Open I has uncontrolled scenario of faces.

According to [2] the method suggest is to use deep learning technique to the detected faces, so that the overhead of deep learning to scan entire image is reduced and scanning is confined to the region that contains faces or false positive which are detected from Haar classifier.

After face detection recognition has to be done on the detected faces. The basic idea of recognition is to consider the images as vectors of pixel. The image vector is compared with a set of trained vectors. This process is called recognition. All the values in the vector does not constitute face, so we use methods like eigen faces or fisher faces to convert the faces into a mean value by passing the faces through a mean value generator function, so that each face can be represented as the combination of the mean faces. We use eigen faces here because its faster and more versatile than fisher faces. There are ANN that can recognize faces but they are not suitable for real-time recognition.

This paper is organized into six chapters, Chapter II deals with related works. chapter III deals with extraction, detection

and refining of faces, chapter IV deals with the training, recognition and self-leaning, Chapter V deals with result analysis and chapter VI is the conclusion.

2 Related Works

The paper [3] present two methods for tracking pedestrians in videos with low and high density of crowds. For videos with low density crowd, [3] first individuals in each video frame is detected using a part-based human detector where occlusion is handled. In the second method, a global data association method based on Generalized Minimum Clique Graphs is used for tracking each. In First method, scene layout constraint is captured by learning Dynamic Floor Field, Static Floor Field and Boundary Floor Field along with flow of crowd and it is used to track individuals in the crowd. In second method, the tracking is performed utilizing contextual and salient information.

This works [4]. focus on detecting and segmenting out crowds of humans from still photos. The goal is to determine if there is a crowd in a sample photo and if so, which portions of the image to be included in the crowd. The detection of a crowd form a uncontrolled image environment is useful task in itself. Crowd formation can cause delay in underground passages, shopping centers and pedestrian paths, or can an indication of civil unrest.

A crowd can be defined as a group of spatially proximate objects of a certain class. The work specifically considers human crowds, as the type that is usually of most concentrated in practice.

There are many reasons which makes crowd detection challenging. First, limited resolution of images that decreases the possibility of detection. Partial occlusions are prevalent in crowds, and the variation in dress, pose, light makes it difficult. Detection of individuals as the basic building element is not a promising approach [4]. Where as a method that directly looks for multiple people, faces problems of modelling an increased range of variability in their combined appearance, also crowd specific factors such as the spacing of individuals in the crowd that is its density.

The work [5] follows the above-mentioned line of work and extends it to the detection and tracking of people in high-density crowds instead of modelling individual interactions of people, the work uses information at a more global level provided by the geometry of scene and crowd density. Some crowd detection method avoids the hard detection task and attempt to infer person counts directly from low-level image measurements. These methods provide person counts in image regions but is uncertain about the location of detected faces.

The goal and contribution of the analyzed work [5] is to combine these two sources of complementary information for improved person detection and tracking. The prediction behind the method, the constraints of person counts in local image regions helps improvement of the standard head detector.

The method is formulated in an energy minimization framework which combines crowd density estimates with the strength of individual face detections. This energy is minimized by jointly optimizing the density and the location of individual faces in the crowd. The work demonstrates optimization of such leads to significant improvements of state-of-the-art person detection in crowded scenarios with varying densities.

With crowd density cues, the constraints provided by scene geometry and temporal continuity of person tracks in the video is explored and demonstrate further improvements for person tracking in highly crowded scenario. The approach is validated on challenging crowded scenes from multiple video datasets.

The work [6] focus on developing effective features and robust classifiers for unconstrained face detection with arbitrary facial variations. Firstly, a simple pixel-level feature, called the Normalized Pixel Difference (NPD) is proposed. An NPD is the ratio of the difference between any two intensity values of pixel to the sum of the values. The NPD has several desirable properties, such as scale invariance, boundedness, and ability to reproduce the source image. It is easy to compute, involving only one addition, one subtraction, and one division between two values pixels per feature computation.

Secondly a method to construct a single cascade classifier that can effectively deal with complex face contour and handle different pose and occlusions. The weak discriminative ability of NDP is solved by indicating that a subset of NPD features can be optimally selected by Ada Boost learning and combined to create discriminative features in regression tree which is a "divide and conquer" strategy to face and optimized unconstrained face detection in a single classifier, without labelling the views in the training set of the face images. The proposed face detector is robust to pose variation, occlusion problem, and angular illumination, also to blur and low resolution image.

This work [7] mainly relies on a head detector to count people from a source image. For detecting the heads from the source image first the point of interest is detected using gradient information from the grey scale image. This approximately locates top portion of the head region to minimize the search region. The points of interest on the source image are masked using a foreground regional space obtained using background subtraction techniques including Vibes and Idiap. Then a sub-window is placed covering the points of interest based on information on perspective calibration and classifies as head or non-head region making use of a classifier. Multiple nearby detections are finally merged to obtain result which is the no of faces.

3 Extraction, Detection and Detection of Faces

The face extraction from an image is an easy process. We make use of 3 pass method to detect faces from front, left and right. The detection of faces from different angles from video is explained in 3 steps. Extraction, detection of face using Haar cascading and removal of false positives using Convolution Neural Network.

3.1 Extraction of image from video and preparation

First the first frame from a video file is extracted using Query Image function. The image file is first converted to grayscale, now the image is subjected to histogram equalization. The output image is ready to be analysed and is passed for detection. After detection, the next frame is extracted and the whole process is done for all frames. Fig 1 shows block diagram of image extraction and Fig 2 shows the image extraction stages.

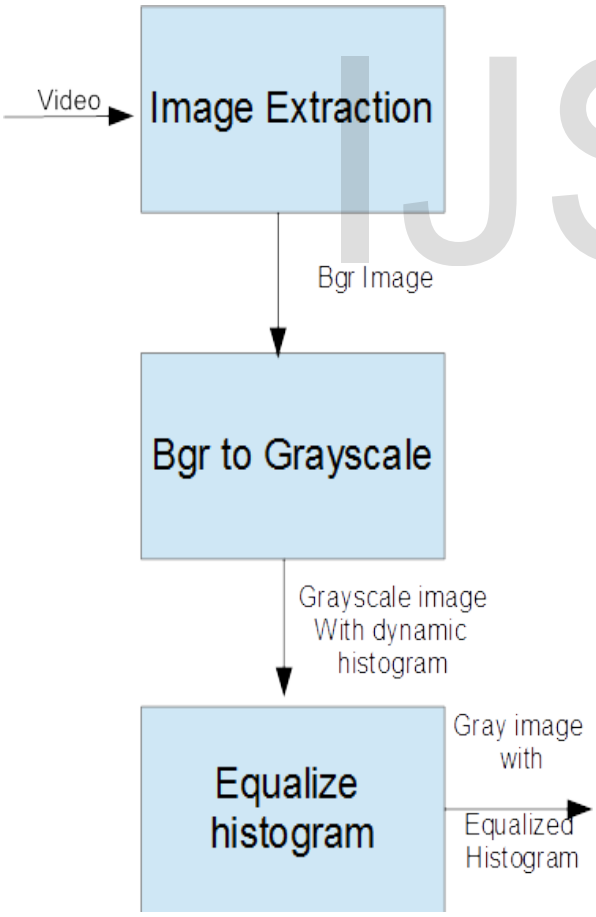

Fig. 1. Image extraction and preparation block diagram

Fig 2. (a) Extracted Image

Fig 2. (b) Gray Scale Image,

Fig 2. (c) Histogram Equalized Image

3.2 Detection of faces using haar cascade classifier.

The detection process is a 3-stage process and is explained below. Fig 3 shows front, left and right profile faces.

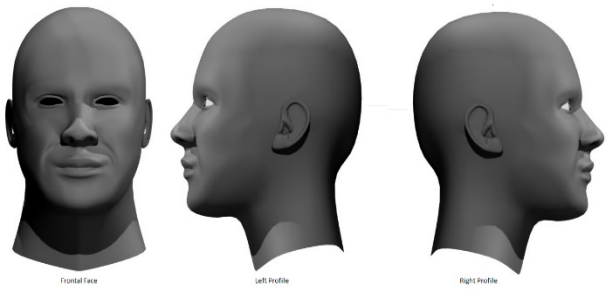

Fig 3. Front, Left and Right Profile faces

First, we detect all frontal faces using frontal face values. The detected face values are stored as rectangles in a list.

Fig 4 shows the detected frontal faces shown by red rectangle

Fig 4. Detected frontal faces shown by red rectangle

Now the same image is subjected profile face values. The left profile faces are detected in this step, the rectangles are compared with front faces list and checked for intersection. If intersection area is > 0, the face is discarded since it is the duplicate of the frontal face already detected. The remaining values are appended to output face list. Fig 5 shows left profile faces detected in green rectangle.

Fig 5. Detected Left faces.

Now the image is flipped horizontally, and subjected to values in profile face again. Here right profile faces are detected, the detected rectangle's x coordinates are inverted using [8].

$$xinvert = width\_of\_rect - (x-1) \quad (1)$$

The rectangles are checked for intersection with faces list and non-intersecting faces are added to list. Now output list contains all faces and false positives from the original image. This list is refined using CNN.

Fig 6 shows detected right faces in blue rectangle

Fig 6. Detected right faces from flipped image

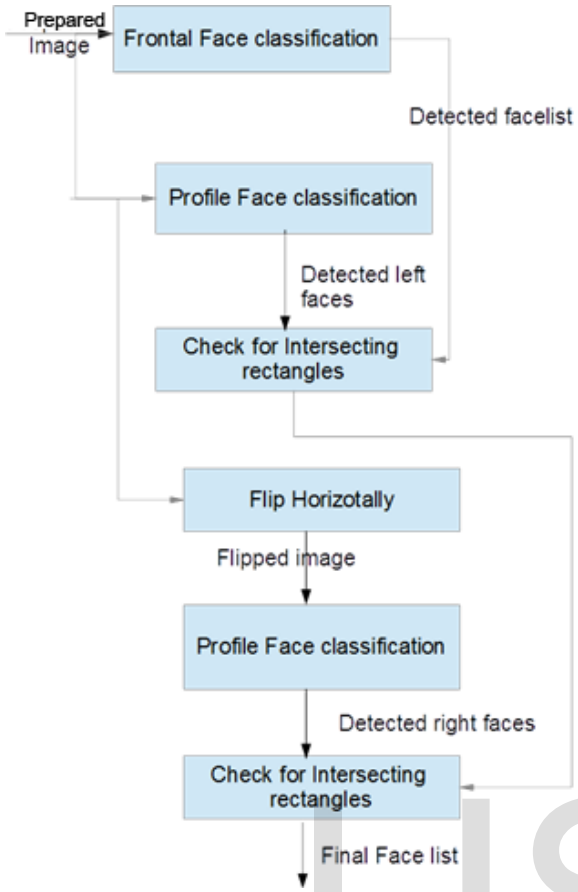

Fig. 7 Haar face recognition block diagram

.

3.3 Removal of false positives using CNN

The convolutional neural networks (CNN) are a special kind of multi-layer neural network designed for image processing. CNN explores spatial relationships of pixels in images to decrease of parameters in the neural network that must be trained [9] [10].

There are four major idea used by CNNs:

1. Local connections: each unit in one layer is connected to a spatially-connected local subset of units in the previous layer;
2. Shared weights: all units in each of the feature maps in one layer have the same set of weights;
3. Pooling: a spatial sub-sampling step is applied to reduce the dimensions of the feature maps;
4. Many layers: the network may have more than 10 layers.

A CNN architecture consists of a number of convolutional and sub-sampling layers and several layers that are fully connected [11]. The convolutional layer has several feature maps. Each unit connected to a local subset of units in the previous layer. That is each feature map is obtained by convolving the input with a linear filter then adding a bias and then passing through a non-linear function. The units in each feature map in the convolutional layer are calculated by sub-sampling layer [12] [13]. The process reduces the computational complexity for subsequent layers and provides a certain degree of shift-invariance. Multilayer perceptron (MLP) are fully connected network. The parameters of CNN are trained through back propagation algorithm.

Open source application cuda-convnet which uses NVIDIA GPUs to accelerate the computation speed can be used. [14] In cuda-convnet, the schemes of local normalization and overlapping pooling are used in a layer to improve generalization. A regularization method called dropout, whose key idea is to randomly drop units from the neural network during training is employed to reduce over fitting in the fully connected MLP layers. For the details of the architecture and the training protocol, refer to [15, 16].

Each of the faces detected are cropped from the source image and is fed to the CNN. CNN accepts if the given image values match the cumulative thresholds of then many to many neural connections the CNN rejects the face, the rectangle is removed from output list.

The Fig 8 shows the input set of CNN.

Fig 8 Input of CNN

The block diagram shown below shows the working of CNN.

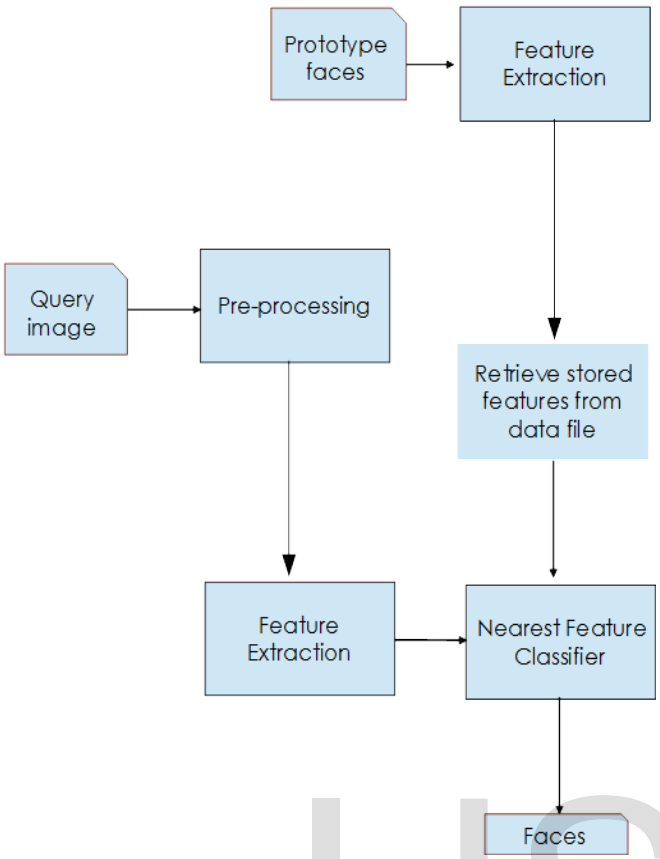

Fig. 9 CNN block diagram

The rectangles in output list are drawn to the source image and displayed. Fig 10. Shows the refined image

Fig 10. Image after removing false positives

4 EIGENFACE TRAINING, RECOGNITION AND SELF LEARNING

Not that the faces haves been detected, the detected we can use eigen face method to recognize the faces, but before recognizing the faces has to be trained. After recognizing we can use same training method for self-learning faces too.

4.1 Training

The training process is simple and straight forward. We use supervised training. First the sample image or a video frame of the person is subjected to detection as explained in section III. The detected face is going to be trained. The trained files are just the cropped down sampled portion of faces stored as 200x200 bitmaps and a file containing the number of trained faces and names in the order of images in our trained dataset separated by a suitable delimiter as shown in Fig. 11

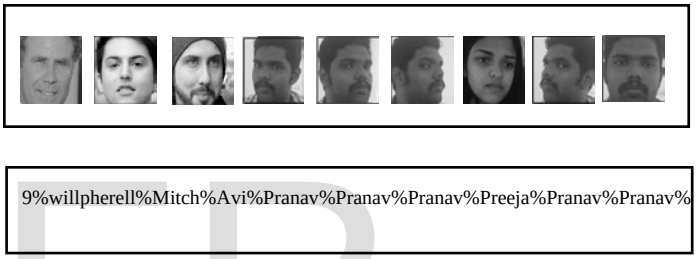

Fig. 11 Trained faces and Content of information file.

Before cropping and storing these images are subjected to gray scale conversion and histogram equalization. The block diagram of the process is shown in Fig 12.

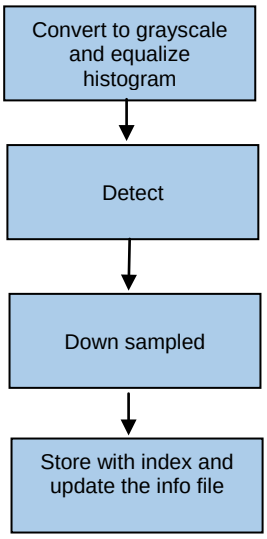

Fig 12. Training Block Diagram

4.2 Recognition

The basic idea of face recognition is to treat pixels of the sam-

ple as a vector and the trained images as set of vectors. Recognition is done by comparing each vector and finding the nearest vector.

When images are considered as vectors of pixel values, images with faces are extremely high-dimensional. It takes a lot of time and recursive matching for recognition which is a resource overhead. For example, a 100x100 image = 10,000 dimensions Making the process slow and increasing storage. But very few 10,000-dimensional vectors are valid face images In order effectively model the subspace of face images, Eigenfaces are used. Eigen faces construct a low-dimensional linear subspace that perfectly explains the variation in the set of images containing faces, known as face space.

For recognition process, a set of eigenfaces generated by PCA from a large set of images. Here we use the images in our trained set.

Considered a set of standardized face ingredients derived from statistical analysis of many pictures of faces. All other faces are combination of these standard faces.

As an example, one's face might be composed of the average face plus 10% from eigenface 1, 55% from eigenface 2, and even -3% from eigenface 3

Eigen faces are stored as a list of values. The steps for eigen value face recognition are given below

1. Compute covariance matrix of face images
2. Compute the principal components ("eigenfaces")
   - K eigenvectors with largest eigenvalues
3. Represent all face images in the dataset as linear combinations of eigenfaces
   - Perform nearest neighbor on these coefficients

During recognition, the detected face that is subjected is first converted to eigen values by eigen value generator, these eigen values are then compared with face space using a cluster mean classifier. The face id with the most confidence level is found out and displayed on screen. Faces below critical confidence are discarded.

.

4.3 Self-Learning.

Self-leaning is the process of recognizing faces that are trained but is not recognized in some intermediate frames due to angle or light differences. During recognition process, a stack is maintained to store faces recognized in the previous frames. That is when faces are recognized in an arbitrary frame, they are stored to stack, now if the faces are detected but if the confidence level is too low to recognize the face then the region is compared to the stack, if the low confidence face id matches the one in the stack, the face is recognized and also the unrecognized frame is cropped and subjected to training so that the face in that particular angle or lighting is recognized next time. The overall process for face recognition and self-leaning is shown in figure 13.

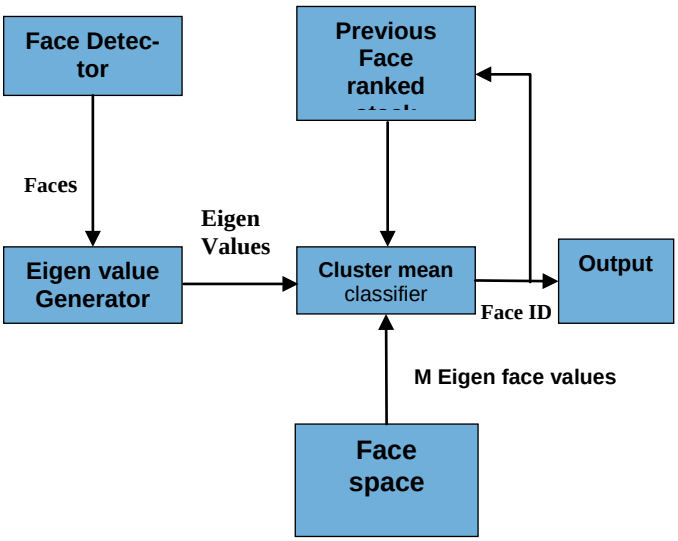

Fig. 13 Face Recognition and Self-leaning

5 Results

The faces were detected and false positives were removed successfully. In picture containing 19 faces, 20 faces were detected, of these 3 were false positives. The CNN removed the three and 17 faces were successfully detected and marked [17] [18].

The accuracy of the method is estimated to be 94.53% on average in any scenario and number of false positives was 0. Speed of detection is significantly 10 times faster than CNN on whole picture [19]. Although detection of frontal face only is 3 times faster than detection of left right and frontal. The performance can be increased by using CUDA Libraries. Fig 14 shows Statistics graph of Haar detection Vs Refined technique and Fig 15 show the speed comparison, Fig 16 shows speed comparison chart, Fig 17,18 and 19 shows Screenshots of the processes.

Below is the accuracy estimation table, accuracy statistics and screenshot of an image detected.

Using the equation

$$\text{Accuracy} = \frac{\textit{No of detected faces}}{\textit{No of faces in samples}} * 100 \quad (2)$$

No of Faces in samples	Detected Faces	Accuracy
19	17	89.47
7	7	100
24	22	91.6
12	11	91.6
6	6	100

Table. 1. Accuracy statistics for refined method

The recognition process is faster than normal CNN and more accurate than traditional viola jones. Self-learning improves each time a face is detected. Over time the system will get maximum efficiency.

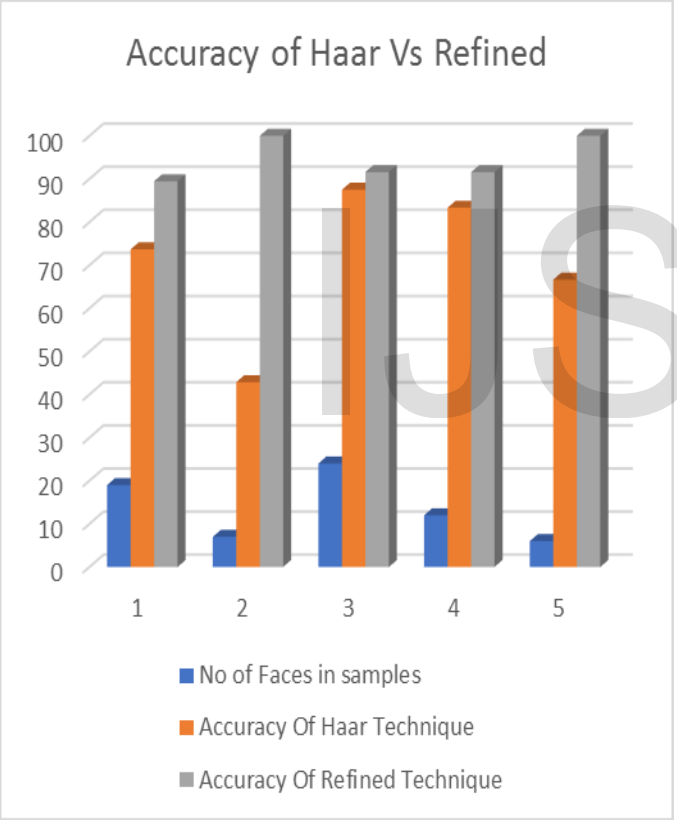

Fig 14. Statistics graph of Haar detection Vs Refined technique

No of faces	Haar	ANN	Combined
12 Faces	1.4s	8.3s	4.3s
2 Faces	0.9s	7.1s	2.1s
4 Faces	1.1s	7.2s	2.9s
1 Face	0.9s	6.9s	1.4s

Fig. 15 Speed analysis of Haar vs ANN vs Combined

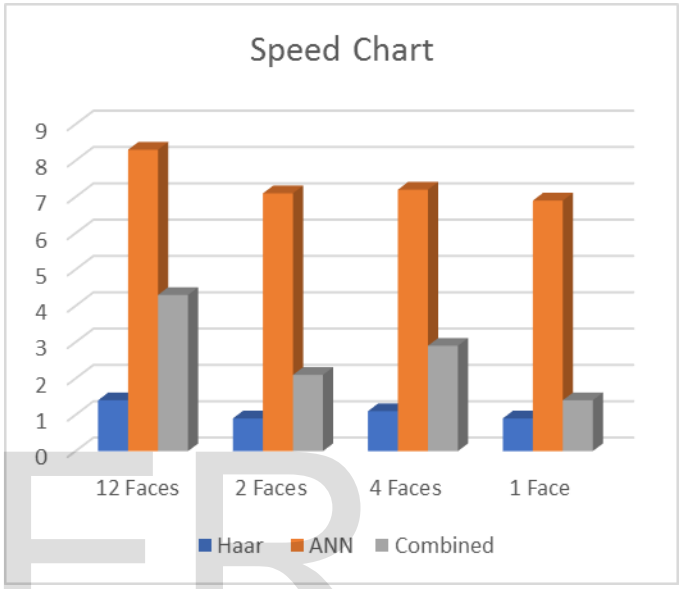

Fig 16. Speed chart showing processing rate for each technique

Fig. 17 Sample output of detection

Fig. 18 Screenshot of Training.

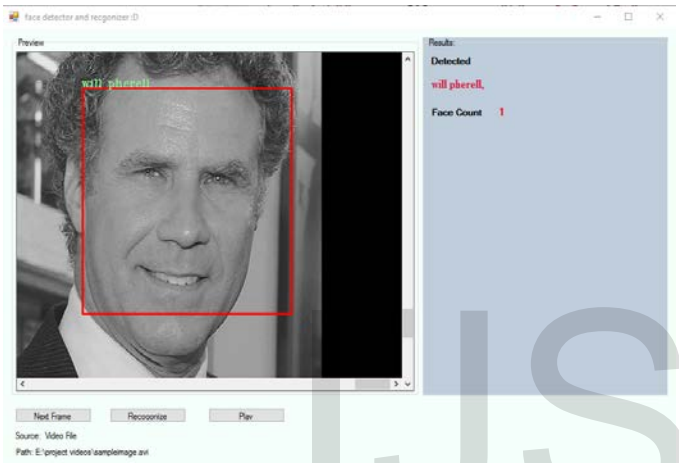

Fig.19 Sample output for recognition

6 Conclusion

The proposed method makes use of haar classification and CNN for an improved and more accurate faster face detection in real-time. The faces are detected from multiple angles with the use of haar cascade classifier. The method can be used in any complex scenario. The use of eigen faces and self-leaning increases the efficiency of the system over time and the system is very easy to implement.

7 Future Work

The speed of detection and recognition can be increased by diving the picture into regions and assigning them to multiple processors. Instead of eigen faces any other ANN based recognition can be used.

.

References

[1] . Arel, I., Rose, D.C., Karnowski, T.P., "Deep machine learning – a new frontier in artificial intelligence research," IEEE Computational Intelligence Magazine, 14 – 18 (November 2010).

[2] . Improving Face Image Extraction by Using Deep Learning Technique Zhiyun Xue, Sameer Antani, L. Rodney Long, Dina Demner-Fushman, George R. Thoma National Library of Medicine, NIH, Bethesda, MD

[3] . Afshin Dehghan, HaroonIdrees, Amir Roshan Zamir, and Mubarak Shah, "Automatic Detection and Tracking of Pedestrians in Videos with Various Crowd Densities", Computer Vision Lab, University of Central Florida, Orlando, USA e-mail: adehghan@cs.ucf.edu

[4] . U. Weidmann et al. (eds.), Pedestrian and Evacuation Dynamics 2012, DOI 10.1007/978-3-319-02447-9__1,

[5] . OgnjenArandjelovi´c, Department of Engineering, University of Cambridge, CB21PZ, UK, Crowd Detection from Still Images, InProc. IEEE Conference on Computer Vision and Pattern Recognition(CVPR),2008.

[6] . Mikel Rodriguez1,4 Ivan Laptev2,4 Josef Sivic2,4 Jean-Yves Audibert3, Universite Paris-Est, Density-aware person detection and tracking in crowds, MPEG video compression standard, International Thompson publishing, 2010

[7] . Shengcai Liao, Anil K. Jain, Fellow, IEEE and Stan Z. Li, Fellow, IEEE, Unconstrained Face Detection 2013

[8] . LeCun, Y., Bengio, Y., Hinton, G., "Deep learning", Nature, 521, 436-444 (2015).

[9] . Lienhart, R., Kuranov, A., Pisarevsky, V., "Empirical analysis of detection cascades of boosted classifiers for rapid object detection," Proceedings of the 25th DAGM Symposium on Pattern Recognition. Magdeburg, Germany, (2003).

[10] . Garcia, C., Delakis, M., "Convolutional face finder: A neural network architecture for fast and robust face detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(11), 1408-1423 (2004).

[11] . Xue, Z., You, D., Chachra, S., Antani, S.K., Long, L.R., Demner-Fushman, D., and Thoma, G. R., "Extraction of endoscopic images for biomedical figure classification," Proc. SPIE. 9418, Medical Imaging 2015: PACS and Imaging Informatics: Next Generation and Innovations, 94180P (March 2015)

[12] . LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., "Gradient-based learning applied to document recognition," Proceedings of the IEEE, 86 (11), 2278–2324 (1998).

[13] . Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L., "ImageNet Large Scale Visual Recognition Challenge," International Journal of Computer Vision, (2015).

[14] . Zhang, C., Zhang, Z., "A survey of recent advances in face detection," Technical Report, MSR-TR-2010-66, Microsoft Research, (2010). [4] Viola, P., Jones, M.J., "Robust real-time face detection," International Journal of Computer Vision, 57(2), 137-154 (2004).

[15] . Krizhevsky, A., Sutskever, I., Hinton, G., "ImageNet classification with deep convolutional neural networks," Neural Information Processing Systems (NIPS), 1097-1105 (2012).

[16] . Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. "Dropout: a simple way to prevent neural networks from overfitting," The Journal of Machine Learning Research, 15(1), 1929-1958 (2014).

[17] . Rowley, H., Baluja, S., Kanade, T., "Neural network-based face detection", IEEE pattern Analysis and Machine Intelligence, 20, 22-38 (1998).

[18] . Milborrow, S., Morkel, J., Nicolls, F., "The MUCT landmarked face database," Pattern Recognition Association of South Africa, (2010).

[19] . Krizhevsky, A., Sutskever, I., Hinton, G., "ImageNet classification with deep convolutional neural networks," Neural Information Processing Systems (NIPS), 1097-1105 (2012).

[20] . Preeja Priji, Rashmi S Nair, (Dec 2016) Survey on multiple face detection and tracking in crowd. International Journal of Innovations in Engineering and Technology. (IJIET). ISSN 2319-1058, Volume 7, issue 4,