International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 06 | June 2020 www.irjet.net p-ISSN: 2395-0072

FACE RECOGNITION IN EMBEDDED SYSTEMS FOR SECURITY SURVEILLANCE

Shreyas Kalkar[1], Shubhankar Kelkar[2], Rutuja Kajale[3], Sunita M. Kulkarni[4]

[1,2,3]Student of MIT College of Engineering, Pune, India
[4]Faculty of Department of EnTC, MIT College of Engineering, Pune, India

---

Abstract - *The ability to automatically comprehend human faces supported by vibrant facial images is vital in defense, surveillance, and the health/independent living domains. Specific applications comprise access control to secure environments, identification of people at a specific locale, and intruder detection. This paper recommends a real-time solution for embedded system applications using IP Cameras. We break the process in the following steps - (1) face detection and (2) face recognition to identify a particular individual. For the first part, the embedded system tracks and crops the faces of the individuals within the frame of the camera. Features from this cropped faces are sent to a computational server for recognition. Now for the second part, an efficient recognition algorithm is used to label the detected faces using a known database. This project uses training and testing facial recognition algorithms and separate training and testing databases. We use the YOLO object detection model for detection and MobileFaceNets model for recognition which enhances our accuracy while preserving speed. We also look at what essential requirements are needed to run light neural networks on embedded systems This system can be implemented in numerous restricted areas, like the office, colleges, residential areas, or at the doorway of a sensitive installation. The system like any other works better under satisfactory lighting conditions and image depths.*

Key Words: **Computer Vision, Pattern Recognition, Face Tracking, Face Recognition, Face Detection, Smart attendance system, IP Camera, CNN Algorithm, YOLO Object Detection, MobileFaceNets.**

1. INTRODUCTION

Facial Recognition is one in all of the various wonders that AI Research has brought forward to the globe and has been a topic of curiosity for several folks. To understand how a machine can recognize faces, we can start by asking ourselves - "How do we recognize a face? Standard frontal images of human faces have two eyes, a nose, lips, forehead, chin, ears, hair which are constant as they rarely change. Yet, faces are different from one another. How do we distinguish between them? At the same time, the face of one person changes with emotion, expression, and age. A simple change in orientation creates a different image. How do we identify a person despite that?"

Generalizing, we can say that the rear components of a face are associated with age, emotion, and orientation while other components stick with the person no matter the age, emotion, etc. Further, we can say the components don't seem to be orthogonal or independent. We all have seen those that some people look alike sideways, but very different otherwise; or a child in a family whose face reminds us of his parents at that age, etc. But there still are some components which are not dependent on age or emotion. Any face recognition system aims to exploit these components. These are done through a long series of mathematical processes on the image. We need to simplify the procedure such that it runs on an embedded system that has modest resources.

Before we run any face recognition algorithms we need to check the required libraries that are required to run on the hardware present. Also, the hardware specifications for a certain system must be kept in mind before deploying any recognition model on it. Albeit the specifications can be reverse engineered to match the requirements of a specific model.

2. RELATED WORKS

Face Recognition has been a region of active research thanks to the interest in biometric security systems. Due to the rapid evolution in computer vision and computational abilities, there has been tremendous progress within this domain. Face Recognition System involves the identification and verification of an individual from an image or video frame. This process involves a group of methods applied to input test images to match those within the database.

Throughout history, there are multiple methods accustomed to carry out facial recognition. Image-based facial recognition techniques known to date are classified so that we can study them properly. Initially, when face recognition as an implementable subject came out, the face as a whole was considered the sole feature. This method which compares the similarity of the whole face is called the Holistic method. Holistic methods kept evolving from Facial Markers to linear algebraic operations on images like Eigenfaces, PCA, etc. In feature-based face recognition systems, the input image is processed to spot, extract, and measure facial features such as eyes, nose, mouth, skin color, etc. For this multiple cascaded filters are used. The rapid progress in Artificial Intelligence is the reason behind the development of Model-

based methods. These models revolve around neural networks. High accuracies can be achieved using this method. Although training a neural network on systems with lower computational power is not the best idea.

The real challenge in face detection and recognition techniques is that the ability to handle all those scenarios where subjects are non-cooperative and therefore the acquisition phase is unconstrained. There are various reasons which cause the appearance of the face to vary. We categorize these sources into two groups: Intrinsic factors and Extrinsic ones [17]. Intrinsic factors are pure to the physical nature of the face and are independent of the observer. These factors are further divided into two classes: intrapersonal and interpersonal. Intrapersonal factors are answerable for varying facial appearance and facial paraphernalia (facial hair, glasses, pose, cosmetics, etc.). Interpersonal factors, however, are answerable for differences in facial appearance of various people (ethnicity, similar faces, and gender). Extrinsic factors cause the looks of the face to alter via the interaction of sunlight with the face and the observer. These factors include illumination, pose scale, and imaging parameters (resolution, focus, imaging, noise, occlusion, shadows, camera not having the ability to locate and concentrate on the face, etc.) [17].

We try to tackle some, if not all, of the above-mentioned problems so we can maximize the efficiency of our face recognition system.

3. MOTIVATION

Face recognition systems proposed till now make use of large databases to achieve accuracy which in turn increases the computational burden on the processor. This also reduces the speed of processing. Also, all these systems oblige a permanent workforce to monitor the operation. Human interaction can occasionally lead to involuntary errors which may cause unaffordable loss. Hence we are trying to overcome all these loopholes in our model. To achieve accuracy, we will provide the required set of databases. But at the same time, we won't compromise with the speed and burden on the processor. This can be achieved by separating the real-time capturing unit and the computational unit of the entire system. This system is capable of real-time data transfer and hence more efficient.

3.1 Previous Models

Initially, analog CCTV cameras were used but later VCR (Video Cassette Recorder) was launched. It could store the footage for nearly 8 hours and to be replaced manually. Later hybrid systems were introduced. It has a partial digital application for the storage of footage. In the time ahead Network based DVR (Digital Video Recorder) came into existence. They were equipped with Ethernet for real-time remote monitoring. With further improvement in technology, Video Encoders have become popular. They can be operated using software installed in computers [1].

3.2 Proposed Model

In the model proposed Analog CCTVs are replaced by fully digital IP Cameras. IP cameras simplified the task of maintenance and installation. IP Cameras are universal. Videos are encrypted and hence secured for all types of transmission. Also, the resolution of IP Cameras keeps on becoming better and better. Even they are capable of detecting smoke, transition, people, etc. And therefore perfect for this application [2].

4. RELATED WORKS

We have implemented facial recognition on a proposed system. But in reality, a user can select any system with similar capabilities in terms of hardware capabilities. The power specification will also change depending on the country, application, and architecture. Although our hardware specification gives an idea about selection for this application but can be extended further for other applications as well.

The Architecture of our FR & FD System is as shown in Fig No. 1. We have designed this system to operate on 24V DC. We have used voltage and current regulators for distributing the power to the different circuit components. A suitable digital IP camera namely OmniVision5640 is selected. We have used the Toradex Colibri iMX6 256MB microprocessor along with the carrier board for enabling interfacing. The data between the microprocessor and server can be exchanged via either the WiFi Module or the GSM Module. We have selected the ESP8266 as our WiFi Module. Along with that we have used the Quectel EC25 as our GSM Module. A high-end microprocessor is not needed as the fps and image quality is within nominal requirements [8].

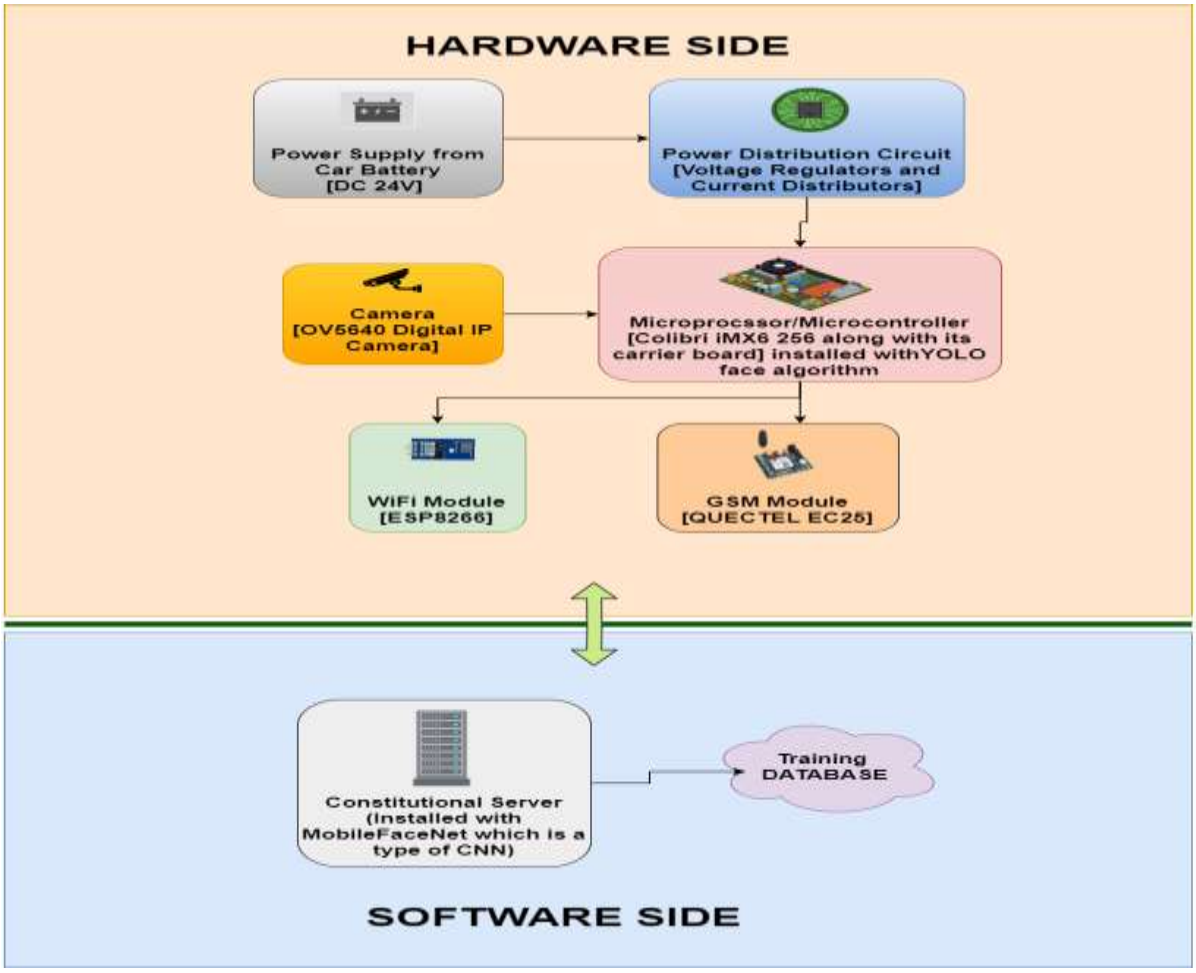

Fig -1: Architecture of our System

4.1 Specifications

Camera: MIPI-CSI [7], Auto-focus OV5640 camera sensor module, 5 Megapixels, Resolution - 2592x1944, Operating Voltage - 3.3V, ¼" CCD Lens, Aperture 2.8mm [3].

Microprocessor: Colibri iMX6 256MB DDR3L RAM - 256MB, Multimedia SoC 1GHz ARM Cortex 9core, Embedded Linux, Micro SD/eMMC up to 64GB [4]

WiFi Module: ESP8266, IEEE 802.11 standard b/g/n support, 2.5GHz, 72.7Mbps Data Rate, Operating Voltage 2.5-3.6 V, Supports Defragmentation, Fast Switching between sleep and wake up modes [5]

GSM Module: Quectel EC25 4G Module, 150 mbps downlink data rate, 50 Mbps uplink data rate, SPI/SDIO Interface, Operating Voltage 3.8V [6]

5. METHODOLOGY

The camera sensor OV5640 used in the hardware having MIPI CSI-2 acts as an input which is interfaced with our system. The Camera first sends the video frames to the microprocessor. Video frames sent by the camera to the processor are of resolution 640x480 pixels. Subsampling is done by binning and the pixel clock is 48/96MHz [4]. The given frames are processed at 5fps by the Colibri iMX6.

Usually, Face Detection Algorithms make use of pixel-wise classifiers or the sliding window approach to perform detection. A slightly better way would be to first predict which part of the image contains more relevant information and then run the classifiers on these regions. The detailed Face Detection Process using the YOLO algorithm [12] is explained below. YOLO divides the input image into an S×S grid. If the center of the face lies in that grid, it is responsible for detecting the face [13]. YOLO predicts bounding boxes containing the possible objects (or in our case face) and also the confidence scores (class probabilities) for those boxes [11].

The faces are thus cropped out of the video frames. Then the facial features are extracted from the faces to form facial vectors. These feature vectors are then sent to the Computational Server. The Computational Server then searches the database and uses the CNN (Computational Neural Network) "YOLO-Face" Algorithm. For detecting a face from frames we use and YOLO algorithms newest version YOLOv3. YOLO, after all, is a convolutional neural network written in Darknet. YOLO has 24 convolutional layers

followed by 2 fully connected layers. YOLOv3 removes the connected layers and has almost 3 times as many convolutional layers [14]. The reason behind using YOLOv3 is one of the faster face detection/object detection algorithms out there with good accuracy [14].

The computational server where trained model is built on a database, uses the CNN "MobileFaceNets" algorithm. It compares the model based which is based on training set to the image vector received from security system and finds the best possible match for the feature vector. It then labels the feature vectors and sends them back to the microprocessor where the identification labels of individuals are displayed.

Suppose our camera provides us with image samples at n-rate. We need an algorithm that detects and processes images less than rate n. In this case, one frame will take 20ms to arrive (5 frames per second => 1s/5=200ms). We need the whole detection and basic image processing part to be completed under 20ms.

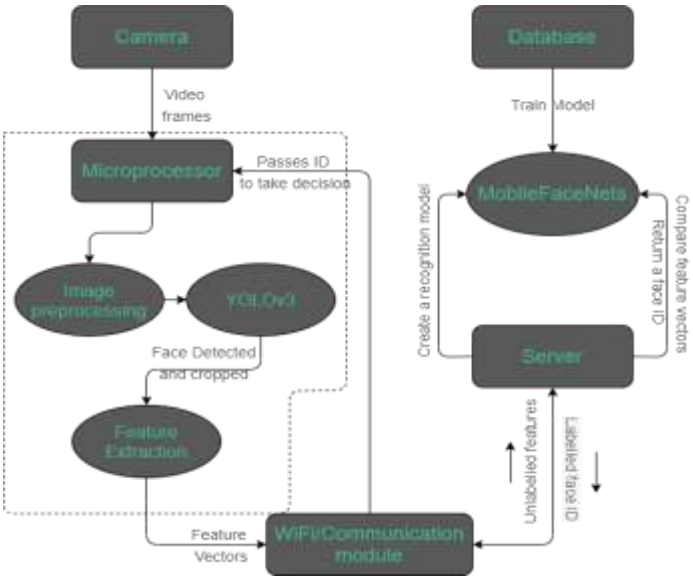

Fig -2: Flowchart of the process

Model	Train	Test	mAP	FLOPS	FPS
SSD300	COCO trainval	test-dev	41.2	-	46
SSD500	COCO trainval	test-dev	46.5	-	19
YOLOv2 608x608	COCO trainval	test-dev	48.1	62.94 Bn	40
Tiny YOLO	COCO trainval	-	-	7.07 Bn	200
SSD321	COCO trainval	test-dev	45.4	-	16
DSSD321	COCO trainval	test-dev	46.1	-	12
R-FCN	COCO trainval	test-dev	51.9	-	12
SSD513	COCO trainval	test-dev	50.4	-	8
DSSD513	COCO trainval	test-dev	53.3	-	6
FPN FRCN	COCO trainval	test-dev	59.1	-	6
Retinanet-50-500	COCO trainval	test-dev	50.9	-	14
Retinanet-101-500	COCO trainval	test-dev	53.1	-	11
Retinanet-101-800	COCO trainval	test-dev	57.5	-	5
YOLOv3-320	COCO trainval	test-dev	51.5	38.97 Bn	45
YOLOv3-416	COCO trainval	test-dev	55.3	65.86 Bn	35
YOLOv3-416	COCO trainval	test-dev	57.9	140.69 Bn	20

Fig -3: Performance comparison in terms of mAP- mean average precision, FPS- frames per second, FLOPS- Floating Point Operations per second [11]

The output bounding box and the labels corresponding to it from our training data can be further processed and prepared for training. We can create an inference session to prepare the model. Moving on from the detection part, we need to implement this trained model on faces in real-time. Modern high accuracy face verification models, similar to detection, tend to be built on deep and bulky convolutional neural networks which are not suitable for embedded applications. MobileFaceNets is an efficient CNN model. We can use any facial landmarks detector and load it for our facial recognition model to work on. How exactly is a Feature Vector obtained from the cropped facial image is explained below in Figure 4.

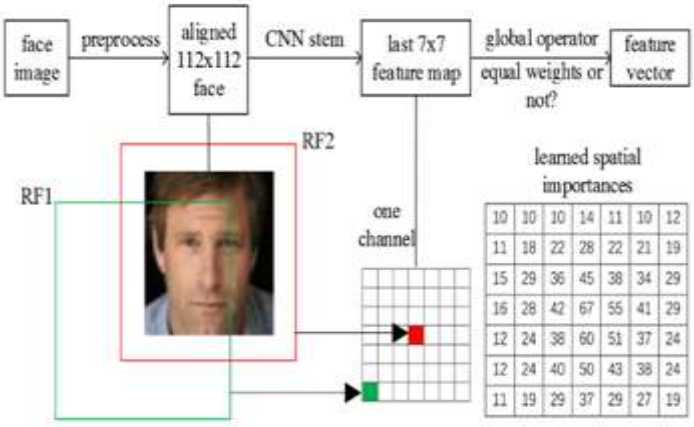

Fig -4: Extracting feature vector from face images

We align the faces by similarity transformation according to the landmarks. Every pixel is normalized in RGB images by subtracting 127.5 and dividing by 128 [12]. Finally, a feature vector is obtained from face feature embedding. The activation function used is the parametric ReLu. It compares the saved embedding from the training model with embedding from the image of the camera and labels the one with the closest mean square difference. We can set a threshold for unnecessary and erroneous detections.

6. EXPERIMENTAL RESULTS AND ANALYSIS

Operating Face Recognition or simply Face Detection on any embedded system with restricted power can be a tricky job. The runtime on GPU-accelerated devices will be much less than on embedded systems with less powerful specifications [10]. We need our embedded systems to do the work of PCs and CoMs (Computer on Modules) are one such example. To integrate OpenCV on embedded systems it has to support multicore processing by additional libraries. Some of the important libraries used by Toradex are OpenMP(Multi-Processing) and TBB (Threading Building Blocks). We use OpenCV to read an image from a camera. The following table is for reading n-image frames per second on Colibri iMX6 using OV5640 with a resolution of 640x480 for various libraries.

Table -1: FPS obtained with different libraries

Library used on Colibri iMX6	Average Frames per Second (1000 samples)
OpenCV 3.1 (TBB)	8.84
OpenCV 3.1 (OpenMP)	8.67
OpenCV 2.4 (TBB)	7.35
OpenCV 2.4 (OpenMP)	7.42

Allotting the threshold for our application is important. High-security applications that urge high accuracy and cannot let any false positives occur will have a threshold on the lower end while those who can tolerate some trade-off for increased accurate face predictions can choose for a threshold on the higher end. The threshold also depends on the number of images used while training. The increase in training images enables us to select a lesser threshold for a particular percent of unknown faces. This can be seen from Fig5-14. For midrange embedded system applications, we select a higher threshold for a fairly limited number of training images over the other way around. Whereas in favorable circumstances, an ideal closing threshold is to be chosen at the elbow point in the graph. The relation of the threshold for various no of unknown faces is demonstrated below.

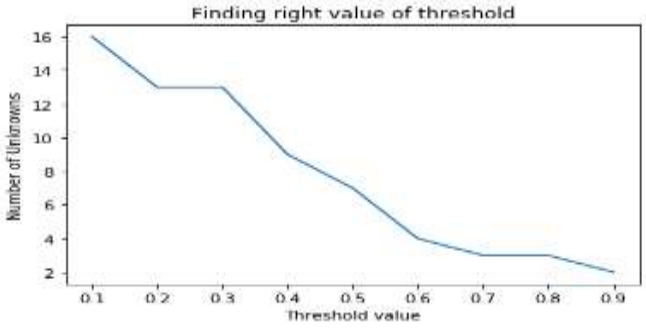

Fig -5: Unknowns in 16 test images

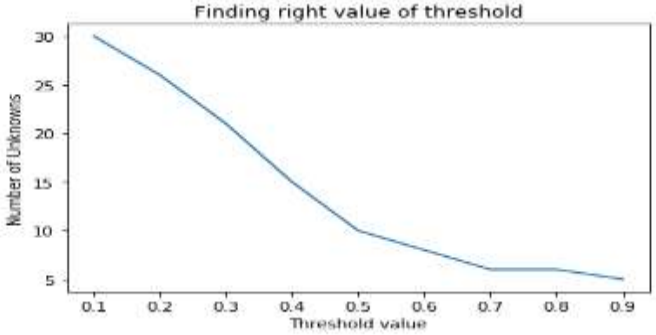

Fig -6: Unknowns for 32 test images

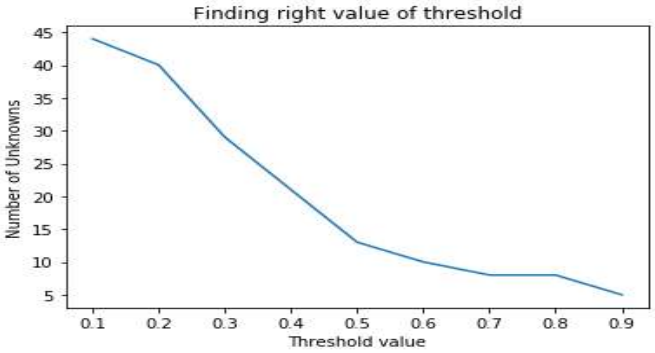

Fig -7: Unknowns in 48 test images

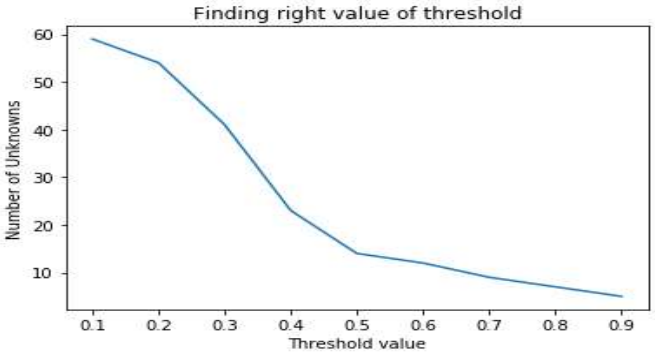

Fig -8: Unknowns for 64 test images

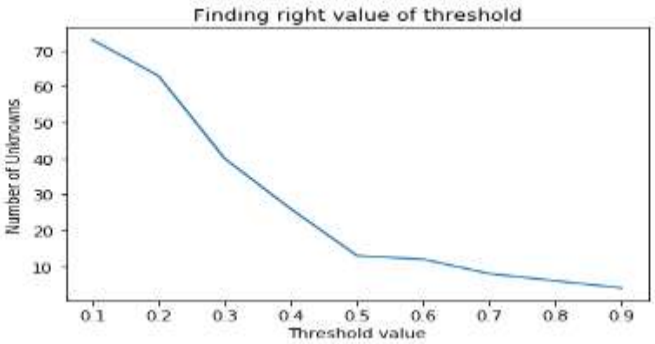

Fig -9: Unknowns in 80 test images

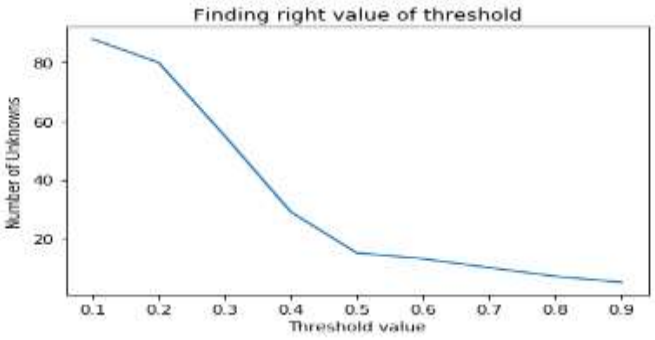

Fig -10: Unknowns in 96 test images

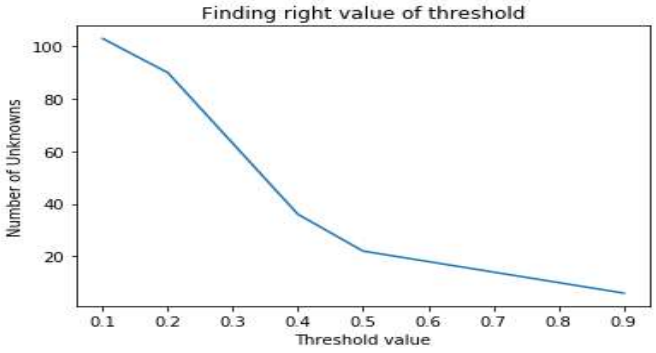

Fig -11: Unknowns in 112 test images

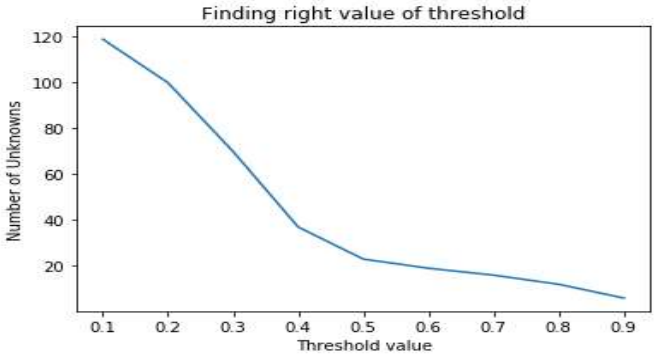

Fig -12: Unknowns in 128 test images

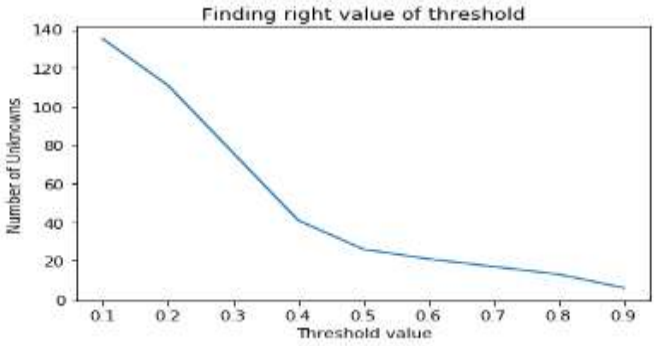

Fig -13: Unknowns in 144 test images

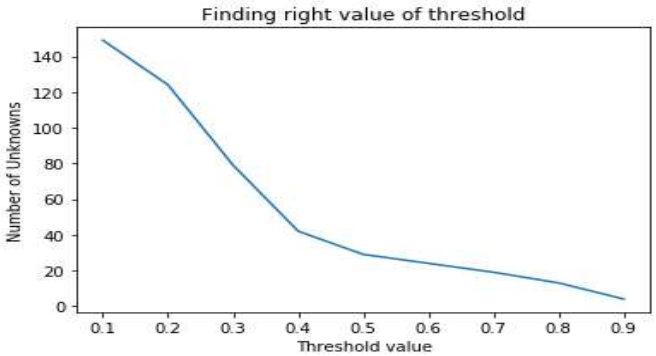

Fig -14: Unknowns in 160 test images

A threshold is the minimum confidence value with which we can accurately detect a face. As seen from the figures above, for lesser threshold values the number of unknown faces increases, and as the threshold increases the number of unknown faces decreases. From the above graphs, it is observed that there is a steep decrease in the number of unknowns up to 0.6 and no significant decrease after 0.6 value of the threshold. Based on these observations we have chosen a threshold of 0.6. We cannot keep the threshold value very high like 0.8 or 0.9 because then the number of misclassified faces increases significantly and this hampers the performance. This is shown in Fig 16 where the misclassification rate increments over its usual rate at the threshold of 0.8.

YOLOv3 is one of the faster face detection/object detection algorithms with good accuracy as compared to other algorithms such as RetinaNet-50 and RetinaNet-101. This can be seen from Fig No. 15. The inference time for the YOLOv3 is the minimum as compared to other algorithms such as RetinaNet-50 and RetinaNet-101.

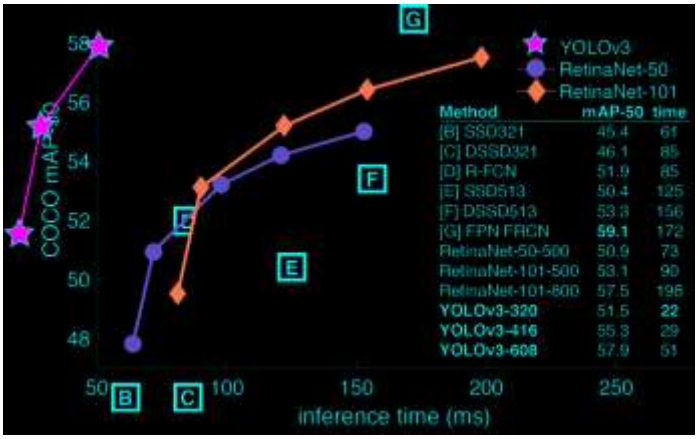

Fig -15: Comparison of mAP and Interference time [14]

Although higher thresholds are desirable as they help us classify more faces, they also lead to increased misclassification rate which is evident from the figure 16. This is because of the increased false positives and false negatives for all classes. For the required applications, the tradeoff between accuracy and misclassification can be set. Although with increasing numbers of training images, we can see that higher accuracies can be obtained at lower threshold levels. Our model attains a maximum accuracy of 93.61% on the training of 200 images. Although a real-life scenario will rarely ask an embedded system to apply a model of 200 trained images.

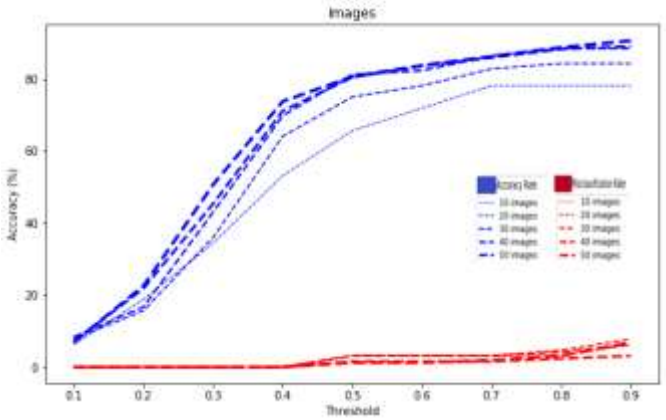

Fig -16: Variation of accuracy and error with a threshold for a varying number of input images

Our database for the obtained accuracies in Fig 18 consists of four faces. Each face has N images at a 60-40 train-test split. The number N is varied to select the right amount of database for training which would result in efficient recognition in embedded systems.

Another measure to determine the performance of a face recognition system is by calculating Precision and Recall. Precision quantifies the number of positive class predictions that belong to the positive class. Recall quantifies the number of positive class predictions made out of all positive examples in the dataset. To calculate these measures, we need to obtain a confusion matrix for all the classes. Precision is given by (TP/TP+FP) whereas Recall is given by (TP/TP+FN) [15]. The average precision for all the classes for our system is 0.8905 whereas average recall is 0.8745. From this, we can say that the average number of false negatives and false positives is almost the same. Although it can vary in between classes. This variance will tell us whether a model is overfitted or under fitted for a particular face ID in a dataset. We can use a precision-recall curve to analyze it further. The F-Score given as (2*(precision* recall)/(precision + recall)) comes out to be 0.8824 for our system [16].

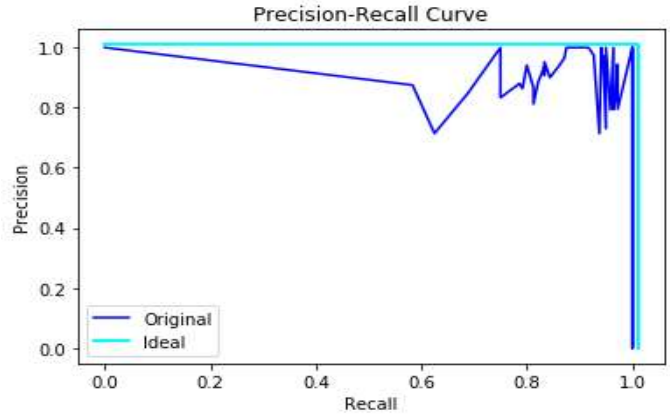

Fig -17: Precision-Recall curve

Accuracy doesn't increase linearly with the increase of training images after a point. As seen from the graph shown in Fig 18 and Fig 16 the accuracy significantly rises until it reaches an elbow point and then further increases slowly. As seen this elbow point corresponds to 60 training images. Hence, we applied a total of 60 images and plotted the Accuracy with the decrease in the number of images and it was exactly 89.56%. The initial high value of accuracy can be explained by the lack of testing images.

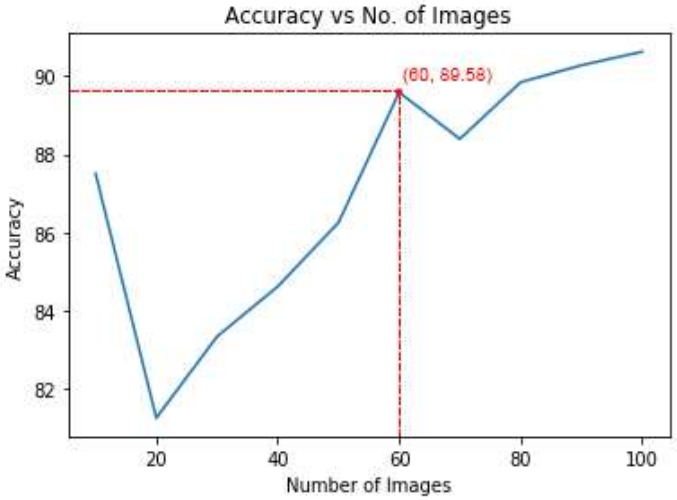

Fig -18: Accuracy

7. CONCLUSIONS

We have acquired a maximum accuracy of 93.61% on the training of 200 images. Even for a realistic approach to train only 60 images, we get a pretty good accuracy of 89.56%, precision of 89.05%, recall of 87.45%, and an f-score of 88.24%. The accuracy can further be boosted with the help of a neat and uniform database. We have highlighted problems in systems that use low training images and exhibited a way to tackle them. We also showed how the threshold value can be tweaked to use in different applications.

Usually difficult to replicate the speed and accuracy of CNN, we have made a ready to deploy a system that could act as a substitute surveillance-based system. This system is a fully Automated System that is eligible to generate an indication just for unknown faces. This system is rapid and cheaper than the traditional ways of executing CCTV systems on FPGAs, CPLDS, etc. The speed and accuracy are set and customized by the user by controlling the database which gives the flexibility to the user and his intended applications. As the outcome process using the Training Algorithm is imparted on a distinct server, we save a lot of computational efforts of our hardware system. Furthermore, this assists the computational server to adjudge the best match much faster. Hence it's an authentic real-time system. There is a lot of

scope of improvement in this system. The system can reimburse for its shortcomings by avoiding overfitting or underfitting of a model and cleaning training data.

REFERENCES

[1] The Crea Vision Website [Online]. Available: https://www.crea.vision/the-blog/2019/6/21/the-evolution-of-cctv-systems.
[2] The Security Magazine Website [Online]. Available: https://www.securitymagazine.com/articles/88854-pros-and-cons-for-ip-vs-analog-video-surveillance.
[3] Toradex, "CSI Camera Module 5MP OV5640 Datasheet", OC5640 Datasheet, Nov 2019.
[4] Toradex, "Colibri iMX6 Datasheet", Colibri iMX6 256MB microprocessor.
[5] [5] Espressif Systems, "ESP8266EX Datasheet", ESP8266 WiFi Module.
[6] Quectel, "EC25 Datasheet", EC25 GSM Module.
[7] The Edge-AI Vision Alliance Website [Online]. Available: https://www.edge-ai-vision.com/2019/03/camera-selection.
[8] Yi Zheng, Joe Manns, Michael Surma, Rick Monroe, Jeffrey Newman, Jia Wan, and Kimm Mueller, "A HIGH SPEED AND LOW-COST DATA AND IMAGE PROCESSING SYSTEM USING DSP TMS320C25 AND AN IBM-PC", Feb 1991 Iowa State.
[9] Laurentiu Acasandrei, Angel Barriga, Manuel Quintero, Alejandro Ruiz, "FACE IDENTIFICATION IMPLEMENTATION IN A STANDALONE EMBEDDED SYSTEM", June 2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE).
[10] Fahad Parvez Mahdia, Md. Mahmudul Habibb, Md. Atiqur Rahman Ahadb, Susan Mckeeverc, A.S.M. Moslehuddinband Pandian Vasanta, "FACE RECOGNITION BASED REAL-TIME SYSTEM FOR SURVEILLANCE", September 2016 Intelligent Decision Technologies.
[11] The YOLO Website [Online]. Available: https://pjreddie.com/darknet/yolo/
[12] Sheng Chen, Yang Liu, Xiang Gao, Zhen Han, "MOBILEFACENETS: EFFICIENT CNNS FOR ACCURATE REAL-TIME FACE VERIFICATION ON MOBILE DEVICES", June 2014 Cornell University.
[13] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, "YOU ONLY LOOK ONCE: UNIFIED, REAL-TIME OBJECT DETECTION", May 2016 Cornell University.
[14] Joseph Redmon, Ali Farhadi, "YOLOV3: AN INCREMENTAL IMPROVEMENT", April 2018 Cornell University.
[15] Towards Data Science Website [Online] Available: https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c
[16] Mala Sundaram, Ambika Mani, "FACE RECOGNITION: DEMYSTIFICATION OF MULTIFARIOUS ASPECT IN EVALUATION METRICS", July 2016 IntechOpen.
[17] Sonia Ohlyan, Ms. Sunita Sangwan, Mr. Tarun Ahuja, "A SURVEY ON VARIOUS PROBLEMS AND CHALLENGES IN FACE RECOGNITION", June 2013 IJERT.