International Journal of Latest Research in Engineering and Technology (IJLRET)
ISSN: 2454-5031
www.ijlret.com || Volume 02 - Issue 07 || July 2016 || PP. 27-34

A Solution for Video Logging, Indexing, Data Analysis and Query Processing

Shubhankar R. Pandit[1], Vikram D. Narke[2], Prof. Dr. S.S. Bhatlawande[3]
[1]*(Department of Electronics, Vishwakarma Institute of Technology 666, Upper Indira Nagar, Bibvewadi, Pune-411 037 India)*

ABSTRACT: This paper présents an architecture and a methodology for high-speed digital video indexing and retrieval for physical security applications. State of the art computer vision algorithms have been used to for the purpose of face detection, face recognition and human detection. All the programs make use of open-source Open CV libraries and functions implemente dusing C++. The real-time video feed is also encoded and encrypted. This makes it possible to circulate this light-weight and protected stream through a network for further analysis and to detect security threats. A query processing scheme using Mongo DB is described to enable flexible and efficient analysis of the data generated from the video feeds.
KEYWORDS-Query Processing**,** Video Indexing, Video logging.

1. Introduction

Traditionally, video surveillance using camera includes monitoring of videos by trained security personnel. Capable of monitoring a modest number of incoming video streams, such personnel become progressively less effective as the volume of video data grows and overloads the ability of the human eye/brain to process visual details. Exacerbating the problem is sheer fatigue, with long shift hours degrading the monitoring abilities of security staff still further. Conventional video surveillance systems can record what they see, but they cannot make sense of what they are viewing. That duty is typically the responsibility of security staff members, who have watched their jobs become increasingly demanding as the average number of deployed surveillance cameras grows.

Our system product aims to perform video analysis of priority based events. For these aims an embedded system is designed and prototyped which is capable of capturing and storing live video for 10 hours. The system is capable of performing real-time data analysis using machine learning and image processing techniques. The salient feature of project is event based indexing and analysis based on priority of events mainly related to security

2. Block Diagram

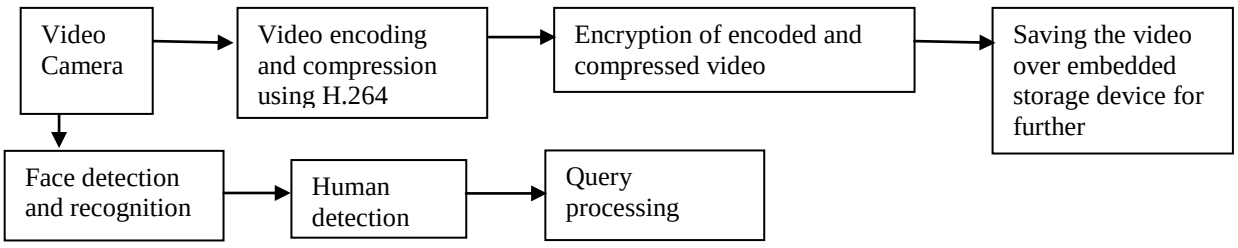

Fig1: Block diagram

Objectives:
1. To capture the video feed and store it on board by compressing and encoding it efficiently.
2. To encrypt the video source feed for further distribution.
3. Processing video for extracting features and region of interest from the input video feed.
4. Indexing the data extracted from the processing and logging it into on board memory.
5. Designing a proper query processing interface to retrieve the logged data.

3. Compression, Encoding, Encryption and Decryption of Video feeds

3.1 Video Capturing

Video surveillance systems traditionally are integrated together with a console which has abundant

data storage facility. Our system being an embedded stand-alone system it was essential that the video feed captured was stored on board with proper encoding with good compression. The system under consideration is being designed keeping industrial, official as well as domestic usage in preview.[1] The camera which acts as a video capturing device sends the video feed to the processor where it is further compressed and encoded using H.264 codec and further encrypted for security purpose.

The development tool used for this coding is Open CV 2.4.10. Open CV is mainly a computer vision library, not a video stream, codec and write one. Therefore, the developers tried to keep this part as simple as possible. Due to this Open CV for video containers supports only the avi extension, its first version. A direct limitation of this is that you cannot save a video file larger than 2 GB. The incoming video stream had to be compressed efficiently without much loss and stored locally over the memory device.

1.H.264 is an open, licensed standard that supports the most efficient video compression techniques available today. Without compromising image quality, an H.264 encoder can reduce the size of a digital video file by more than 802. H.264 has already been introduced in new electronic gadgets such as mobile phones and digital video players, and has gained fast acceptance by end users.

2.Service providers such as online video storage and telecommunications companies are also beginning to adopt H.264.

3.In the video surveillance industry, H.264 will most likely find the quickest traction in applications where there are demands for high frame rates and high resolution, such as in the surveillance of highways, airports and casinos, where the use of 30/25 (NTSC/PAL) frames per second is the norm. This is where the economies of reduced bandwidth and storage needs will deliver the biggest savings.

4.H.264 is also expected to accelerate the adoption of megapixel cameras since the highly efficient compression technology can reduce the large file sizes and bit rates generated without compromising image quality.[2]

3.1.1Basic Methods for reducing data

Basic methods of reducing data A variety of methods can be used to reduce video data, both within an image frame and between a series of frames. Within an image frame, data can be reduced simply by removing unnecessary information, which will have an impact on the image resolution. In a series of frames, video data can be reduced by such methods as difference coding, which is used by most video compression standards including H.264. In difference coding, a frame is compared with a reference frame (i.e. earlier I- or P-frame) and only pixels that have changed with respect to the reference frame are coded. In this way, the number of pixelvalues that are coded and sent is reduced. With Motion JPEG format, the three images in the above sequence are coded and sent as separate unique images (I-frames) with no dependencies on each other. With difference coding (used in most video compression standards includingH.264), only the first image (I-frame) is coded in its entirety. In the two following images (P-frames), references are made to the first picture for the static elements, i.e. the house, and only the moving parts, i.e. the running man, is coded using motion vectors, thus reducing the amount of information that is sent and stored. The amount of encoding can be further reducedif detection and encoding of differences is based on blocks of pixels (macroblocks) rather than individual pixels; therefore, bigger areas are compared and only blocks that are significantly different are coded.[3]

The overhead associated with indicating the location of areas to be changed is also reduced. Difference coding, however, would not significantly reduce data if there was a lot of motion in a video. Here, techniques such as block-based motion compensation can be used. Block-based motion compensation takes into account that much of what makes up a new frame in a video sequence can be found in an earlier frame, but perhaps in a different location. This technique divides a frame into a series of macro blocks. Block by block, a new frame for instance, a P-frame can be composed or predicted by looking for a matching block in a reference frame. If a match is found, the encoder simply codes the position where the matching block is to be found in the reference frame.

Coding the motion vector, as it is called, takes up fewer bits than if the actual content of a block were to be coded.

3.2 Basics of Encryption and Decryption

1)Encryption: A process of encoding a message so that its meaning is not obvious.

2) Decryption: The reverse process.[4].We studied all video encryptions and found some drawbacks.

1.Completely Layered Encryption-In this method, the entire video is first compressed and is then encrypted using traditional algorithms like RSA, DES, and AES. This technique is not applicable in real time video applications due to heavy computation and very low speed.

2.Encryption Using Permutation-Here, the video content is scrambled using a permutation algorithm. The

entire video content may be scrambled or only particular bytes. A permutation list maybe used as a secret key for encryption.

3. Selective Encryption-To save computational complexity only particular video bytes may be encrypted.**4.Perceptual Encryption-** After encryption using this technique the video will still be perceptible. The audio/video quality can be controlled continuously.[5][6]

3.2.1 Our proposed algorithm

1.Encryption Process- We are using Open CV'srandu() function and bitwise xor() to do encryption.

i. Input a video.
ii. Separate the three frames.
iii. Perform Encryption by using bitwise xor() .
iv. Translate the video.

2.Decryption Process-Input a cipher video.

i. Break the video into frames.
ii. Decompress the frames.
iii. Combine entire channel.
iv. Convert into frames and output will be a video.

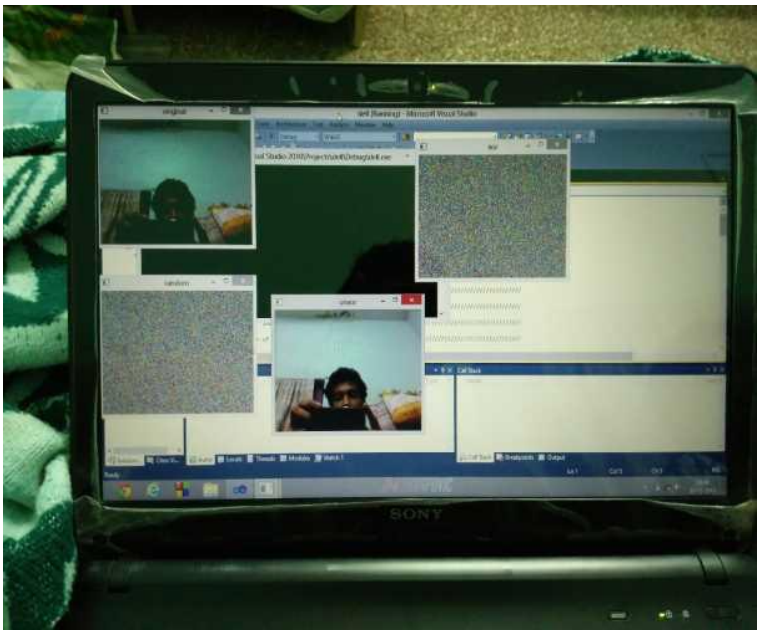

Fig 2:Figure shows that four windows.In first window original video,in second window video is compressed and encrypted ,in third window video is decrypted ,in fourth window random video has shown.

4. Real-time Video Processing Materials and Methods

4.1 Real-time Face Detection

Object Detection which uses Haar feature-based cascade classifiers is an effective object detection method proposed by Michael Jones and Paul Viola in 2001 in their paper "Rapid Object Detection using a Boosted Cascade of Simple Features in 2001". In this a cascade function is trained from a number of positive and negative images. It is a machine learning based approach. Then it is to detect the objects. It requires positive and negative images to train the classifier initially. Positive images contain images of faces and negative images do not contain images of faces. Then, we need to extract features from it. Haar features are used for this purpose. However, it is estimated that even a 24x24 window would result in 160000 features. And among all these features , most of them are irrelevant. So, we use Adaboost to select the best features out of them. In this, we apply each and every feature on all the training images. For every feature, it will find the optimum threshold which will classify the faces as negative or positive. Out of these, we select the features that best classifies the face and non-face images. Final classifier is a weighted sum of all the weak classifiers. They are called weak because alone they cannot classify the image, but together they form a strong classifier. The final set-up of Viola-Jones method had around 6000 features. Thus, we take an image , take a 24x24 window and apply 6000

features to it to check if it is a face or not. However, this proves to be inefficient and time consuming. For this, they introduced the concept of cascade of classifiers. In this, they group the features into different stages of classifiers and apply one by one instead of applying all the 6000 features at once. If a window fails the first stage, it is discarded and the remaining features are not considered. The one which passes all the stages is a face region.[14]

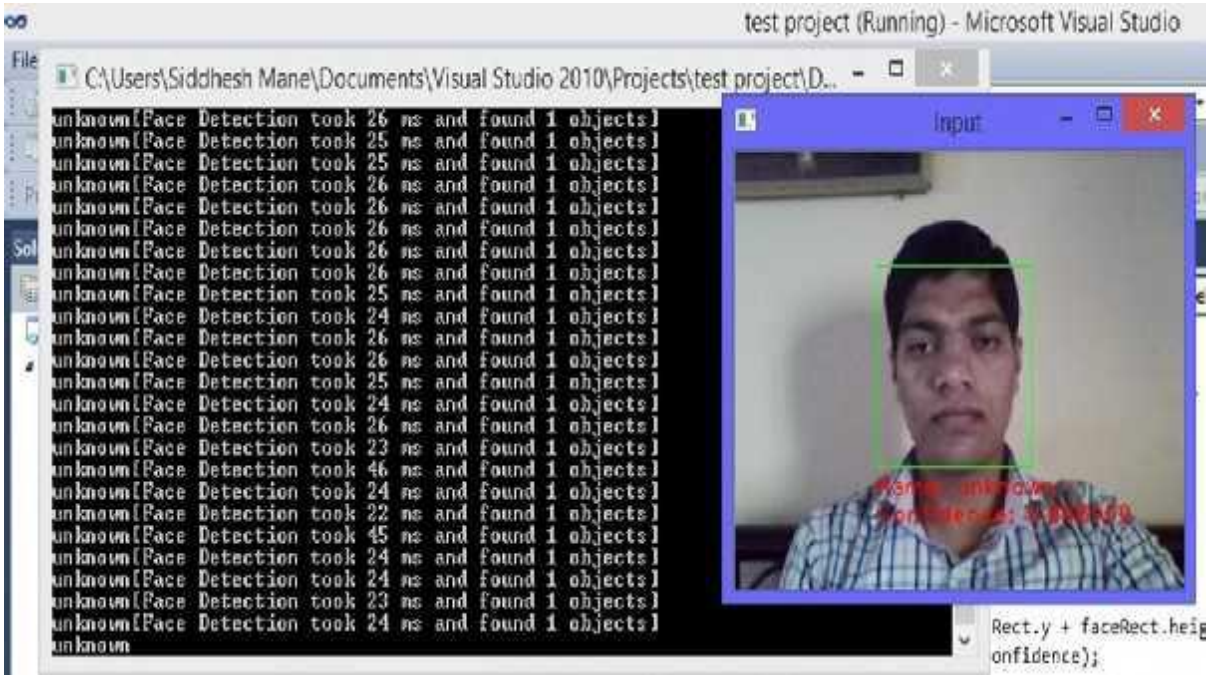

Fig. 3 :The above images is the output of the face detection code implemented in Visual Studio 2010 IDE using OpenCV libraries in C++.

4.2 Real-time Face Recognition

It is definitely one of the most popular and intriguing problems in Computer Vision. It is also one of the most challenging in the group of machine learning problems. Following are the major obstacles faced in face recognition:

1)Even a slight change in the illumination of the image can widely affect the results.
2)Pose changes any rotation of the head of a person will affect the performance.[12]

4.2.1 EigenFaces

After carefully studying several methods or approaches to face recognition like Gabor filters , Neural networks and Correlation and EigenFaces, because of the low computational complexity and higher accuracy, we have used proposed to use EigenFaces algorithm for Face Recognition in our system. It is a dimensionality reduction scheme. Principal component analysis is the most commonly used for dimensionality reduction in computer vision. Consider N sample images,$x_1, x_2, \ldots, x_N$taking values in an n-dimensional image space, and assume that each image belongs to one of c classes.$X_1, X_2, .., X_c$. Let us also consider a linear transformation mapping the original n-dimensional image space into an m-dimensional feature space, where m < n. The new feature vectors are$y_k \Leftarrow R_m$defined by the following linear transformation$y_k = W_T x(k)$; k=1,2,…,N where$W \Leftarrow R_m$is a matrix with ortho-normal columns. If the total scatter matrix $S_T$is defined as $ST = \sum_{k=1}^{N} (xk - \mu)(xk - \mu)T$where N is the number of sample images, and$\mu \Leftarrow R_n$is the mean image of all samples, then after applying the linear transformation W_T , the scatter of the transformed feature vectors$\{ y1, y2, .., yN \}$is $W_T S_T W$ .In Principal Component Analysis, the projection$W_{opt}$is chosen to maximizethe determinant of the total scatter matrix of the projected samples i.e.$W_{opt} = argmax| W_T S_T W| = [w_1 w_2 \ldots w_m]$ where$\{w_i | i = 1,2, \ldots, m.\}$ is a set of n-dimensional eigenvectors of $S_T$corresponding to m largest eigenvalues. These eigenvectors have the same dimension as the original images[13]. In this method, each training face image is decomposed and expressed in terms of the same

eigenvectors of the scattering matrix. And then , when the input probe image which is to be classified is expressed in terms of the same eigenvectors and then the Euclidean distance is calculated with the help of the coefficients of the vectors. The class with which it has the least distance is the one to which the probe face image belongs. This is used to calculate the confidence or the closeness of the incoming face image to the class to which it has been mapped to. We have added the additional feature of detecting an unknown person. That is, whenever the confidence falls below a particular threshold level, we classify the person as unknown. The control of this threshold lies in the hands of the user or the administrator.

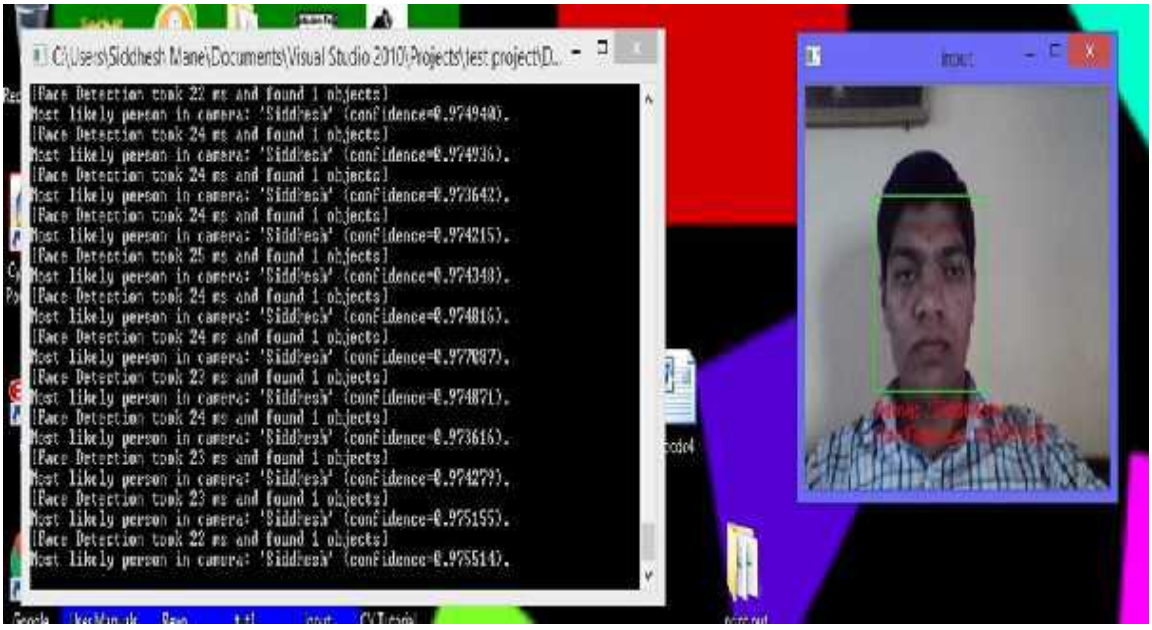

Fig. 4 :This is the output of our face recognition program implemented using EignFaces function using OpenCV libraries in C++. The IDE used was Microsoft Visual Studio 10.

4.3 Human Detection using HOG descriptors in real-time

HOG stands for Histograms of Oriented Gradients. HOG is a type of feature descriptor. The intent of a feature descriptor is to generalize the object in such a way that the same object (in this case a person) produces as close as possible to the same feature descriptor when viewed under different conditions[15]. This makes the classification task easier. The creators of this approach trained a Support Vector Machine (a type of machine learning algorithm for classification), or SVM, to recognize HOG descriptors of people [7]. HOG uses global feature to describe a person rather than a collection of a number of local features. That means the entire person is represented by a single feature vector. It uses a sliding window detection technique. At each position of the detector window, a HOG descriptor is calculated. This is then shown to a trained SVM classifier which then classifies it as a human or not. To detect a person at different scales, the image is sub-sampled into multiple sizes [8]. Using this, we can tell whether the detected entity is an animal or not.

4.3.1 Gradient Histograms

The person detector uses 64x128 pixel detection window. HOG descriptor is calculated on 8x8 pixel cells within the detection window. The gradient vector is calculated at each and every pixel. Like this 64 gradient vectors are calculated and put into 9-bin histogram. The histogram ranges from 0 to 180 degrees, so there are 20 degrees per bin. For each gradient vector, its magnitude gives the contribution to the histogram. To make the gradient values invariant to multiplication of pixel values, the gradient vectors are normalized [9].

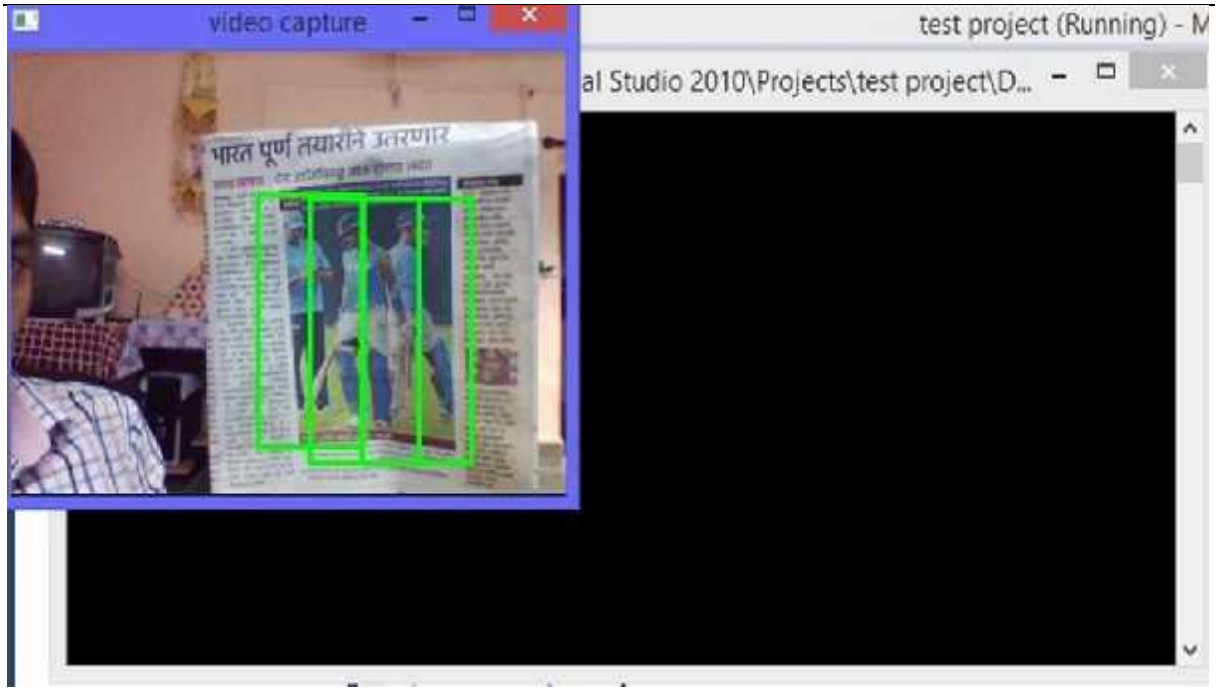

Fig. 5: This is the output of the human detection code using HOGDescriptor function of OpenCV libraries.

5. Query Processing

The output of Face Recognition program consists of a text indicating the name of the person being detected in the current video frame, the confidence or the accuracy with which it was detected with respect to the training images and the time at which it was recognized as well the time in milliseconds it took for the algorithm to recognize the person. After this we wish give to give the user a platform with which it can easily navigate through this data and extract the relevant instances according to its need. For this query processing is used. The output of the face recognition algorithm is given to the MongoDB console and stored in the database named persons. As a document-oriented database, MongoDB is a more generalized No SQL solution. It should be viewed as an alternative to relational databases. Like relational databases, it too can benefit from being paired with some of the more specialized NoSQL solutions. MongoDB is a free and open-source cross-platform document-oriented database. Classified as a NoSQL database, MongoDB avoids the traditional table-based relational database structure in favor of JSON-like documents with dynamic schemas making the integration of data in certain types of applications easier and faster. Since our input data in the database does not change its form it is beneficial to use MongoDB [10].

For the purpose of simplicity of operation, the way in which the output of the face recognition program is printed in the notepad file is changed. It format is changed to a format of a MongoDB command to insert the data entry into the database, which in this case is ‘persons’.

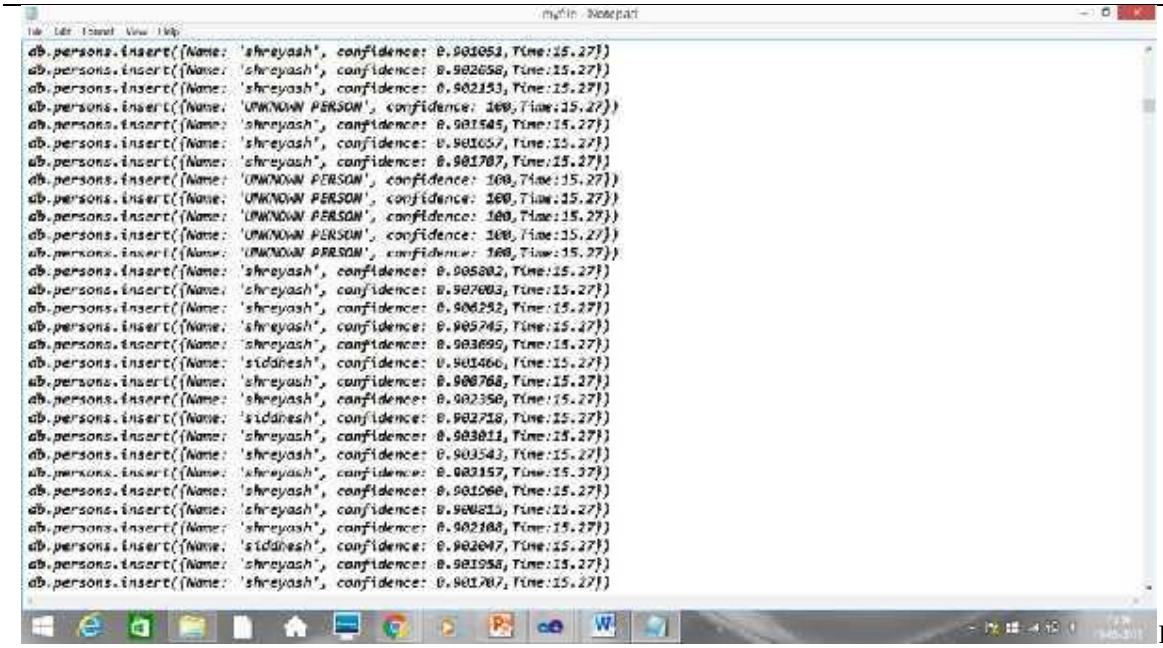

Fig. 6: Output text file of face recognition code

Once the data is copied to the MongoDB console, the user can enter commands in standard MongoDB command format to extract the necessary instances, like the below image for example, all the data entries where the confidence of the recognized person is greater than or equal to 0.92 is shown.

Fig. 7: Output in the console after a query is input.

6. Conclusion

Our system captures the video feed and stores it by compressing and encoding with good efficiency. It also encrypts video source feed for further distribution. The encoded video is compressed optimally to save memory footprint and make the video feed available remotely over the network which is done with help of H.264 codec. Video processing techniques like face detection, face recognition and human detection were successfully implemented. We have used voila jones algorithm for face detection. For face recognition we have used Eigen Faces algorithm.

Special feature of our system is query processing. With the help of this feature we have successfully overcome the drawback of conventional video surveillance system. We are currently working on system integration of all the above mentioned sub-systems. After thorough analysis we have chosen Beagle board XM as the hardware platform for implementation.

Further, this existing framework can be extended to track humans and perform motion estimation. The different gestures and actions of humans can be recognized by using training from a data set or feature points. This would find use in autonomous driving vehicles. This system can be used to recognize different actions made the traffic police and then take there quired course of action as far as the motion of the car is concerned. Different human postures or poses can be recognized and can be categorized as to which positions are dangerous or which actions are suspicious. This can be further advanced to track and understand micro expressions are very brief facial expressions, lasting only a fraction of a second. They occur when a person either deliberately or unconsciously conceals a feeling. With the help for micro-expressions, we can spot concealed emotions. Face Suite and PEG Interactive [11] research on micro expression has shown that we often miss facial expressions when they contradict words being spoken. This can be used for the purpose of lie detection.

REFERENCES

[1]. Ke Li, Ke-Bin Jia, Jing Xie, and Yan Wang, ”Design and Optimization of H.264 Video Encoder on DSP Platform [C]”, Second International Conference on Innovative Computing, Kumamoto, Japan, 2007, pp. 541-541.
[2]. Hou-Jie Bi, ”The new generation video compression coding standard - H.264/AVC [M]”,Posts and Telecom Press, China, 2005.
[3]. N. Kamaci and Y. Altunbasak, “Performance comparison of the emerging H.264 videocoding standard with the existing standards,” Multimedia and Expo, 2003.ICME ’03.Proceedings.2003 International Conference on, 2003, pp. I-345-8 vol.1.
[4]. Basic Encryption and Decryption by H. Lee KwangDepartment of Electrical Engineeringand Computer Science,KAIST and An Overview of Video Encryption Techniques by M.Abomhara, Omar Zakaria, Othman O. Khalifa.
[5]. Shonharris, (2007). “SICCP Exam Guide, fourth edition, McGraw-Hall”.
[6]. Stallings, William, (2007). “Network Security Essentials, applications and Standards,Pearson Education, Inch.”
[7]. Isaac Cohen and Gerard Medioni.“Detecting and Tracking Moving Objects for VideoSurveillance” prsented at IEEE Proc. Computer Vision and Pattern Recognition Jun. 23-25,1999. Fort Collins CO.
[8]. ShervinEmami ”Introduction to Face Detection and Face Recognition”
[9]. http://www.shervinemami.info/faceRecognition.html, 2nd June, 2010
[10]. OpenCVdevteam ”Face Recognition with OpenCV”OpenCVdevteam ”Hough Circle Transform”
[11]. Face Suite and PEG Interactive to offer you the most training in emotional awareness andenhanced communication(2016) http://www.paulekman.com/micro-expressions/
[12]. Jan Fajfr“The basics of face recognition” Posted on 20/10/2011.
[13]. Peter N. Belhumeur, Jo ao P. Hespanha, and David J. Kriegman. ”Eigenfaces vs. Fisherfaces:Recognition Using Class Specific Linear Projection” presented at IEEE TRANSACTIONSON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 19, NO. 7,JULY
[14]. Paul Viola, Michael J Jones “Robust Real-Time Face Detection” International Journal of Computer Vision 57, pp. 137-154, Netherlands, 2004.
[15]. NavneetDalal and Bill Triggs, ”Histograms of Oriented Gradients for Human Detection”presented at 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05).