International Journal of Engineering &Technology, *7 (3.12) (2018) 244-247*

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Real Time System for Human Identification and Tracking from Surveillance Videos

D.Vishaka Gayathri[1*], Shrutee Shree[2], Taru Jain[3], K. Sornalakshmi[4]

[1, 2, 3, 4]Department of Information Technology SRM University,
Kattankulathur, Kancheepuram District, Tamil Nadu
**Corresponding Author E-mail: [1]vishakagayathri.d@gmail.com, [2]shruteeshree26@gmail.com,*
[3]taru7jain@gmail.com, [4]sorna.krishnan84@gmail.com

Abstract

The need for intelligent surveillance systems has raised the concerns of security. A viable system with automated methods for person identification to detect, track and recognize persons in real time is required. The traditional detection techniques have not been able to analyze such a huge amount of live video generated in real-time. So, there is a necessity for live streaming video analytics which includes processing and analyzing large scale visual data such as images or videos to find content that are useful for interpretation. In this work, an automated surveillance system for real-time detection, recognition and tracking of persons in video streams from multiple video inputs is presented. In addition, the current location of an individual can be searched with the tool bar provided. A model is proposed, which uses a messaging queue to receive/transfer video feeds and the frames in the video are analyzed using image processing modules to identify and recognize the person with respect to the training data sets. The main aim of this project is to overcome the challenges faced in integrating the open source tools that build up the system for tagging and searching people.

Keywords: *Face detection, Face identification, Real time system, Distributed streaming analytics*

1. Introduction

The extensive usage of surveillance cameras at public places has elevated the role of technologies concerned with video content analysis essentially in many fields, mainly security. Research in face processing have expanded rapidly due to the fact that information about a person's identity, state, and intent can be extracted from images, and that it can yield useful results. There is a demand in computing power, high speed network infrastructure and large capacity storage devices. There are obstacles in detecting, recognizing and tracking people in videos due to gesture and lighting variations, image orientation, imaging conditions and occlusions. However, notable challenges are faced when trying to append such techniques as automated solutions involving real time implementation. This work aims at analysing surveillance videos and building a real time humanistic system that identifies and tracks people automatically in multiple video footages captured by multiple cameras established in a distributed framework.

Our underlying motive is to resolve the issues faced in order to meet the major goals which are to devise an automated, real time system on a distributed network using big data analytics. An implementation stack has been wisely constructed in order to meet the specific requirements using open source tools. The system we have designed captures a video through multiple sources at remote locations, split the video into frames using an image processing library, add metadata tags in each of the frames using a metadata library, sending them via a distributed messaging queue to the processing machine. Each of the frames is analysed using the image processing library to detect humans and recognize them based on the training data and track their movement.

The interface allows live tracking and searching in current video streams taken as input. All these techniques are processed continuously in a real time computation environment which distributes tasks over several nodes to achieve scalable, fault tolerant results. A database, suitable to work in such a framework, is used to store and allow indexing of human related information for supporting human identification. The main challenge is the coordination and communication between all the working components of our system in order to achieve the unified aim.

2. Literature Review

In this section, we review existing techniques relevant to the problem of detecting, identifying, and tracking people in video. Researches in the areas of moving object tracking, human detection, human recognition, human tracking have proposed many promising approaches. As mentioned in [1] [2], numerous algorithms for face recognition have been proposed. Solutions have been attempted using a wide variety of methods like Principal Component Analysis (PCA) [3], Independent Component Analysis (ICA) [4], Linear Discriminant Analysis (LDA) [5], Kernel Methods [6], Trace Transform [7], Support Vector Machine (SVM) [8], etc. Numerous face recognition studies and researches have pointed to conclusions wherein the usage of Eigenfaces and Fisherfaces were involved. The Eigenface method uses the PCA for maximizing scattering of all projected samples by dimensionally reducing linear projection. The Fisherface method is an enhancement of the Eigenface method that it uses Fisher's Linear Discriminant Analysis (FLDA or LDA) for the dimensionality reduction Further, comparison between the two algorithms under varied lighting conditions, different facial expressions and in linear subspace has been deeply analyzed and discussed in [9]. It has been concluded that, the Fisherface method appears to be the best at extrapolating/interpolating over variation in lighting and expression. Till date, many frameworks have been proposed for human identification using surveillance cameras. The problem of matching high-resolution gallery images with surveillance quality probe videos is addressed in [10]. A learning-based likelihood measurement model is used to handle large appearance and resolution differences between gallery images and probe videos which decreases the problems faced during matching procedures in face recognition.

Till date, many frameworks have been proposed for human identification using surveillance cameras. [11] Provided a useful approach for proceeding with the implementation of this project in real time scenarios. [12] Has helped in outlining the essentials which build intelligent video systems and analytics. [13] proposes a system that integrates detection, identification and tagging of humans in a video but from a single static camera.

A successful framework would be one containing useful properties such as real time and automated performance, management of distributed framework in video input and cost effectiveness compiled as a single system. According to this summary, it can be inferred that interconnectivity between such properties within a single system does not persist. There is a gap between the effectiveness of automated systems and real time face tracking and recognition algorithms. In these areas, much work to be done.

3. Proposed Methodology

The method we propose consists of various different modules each pertaining to an aspect of solving the problem at hand. The implementation follows a pipeline which contains different modules.

The proposed methodology is divided into 4 main modules:

1. Video capture
2. Embedding metadata to frames
3. Queueing the frames
4. Image Processing

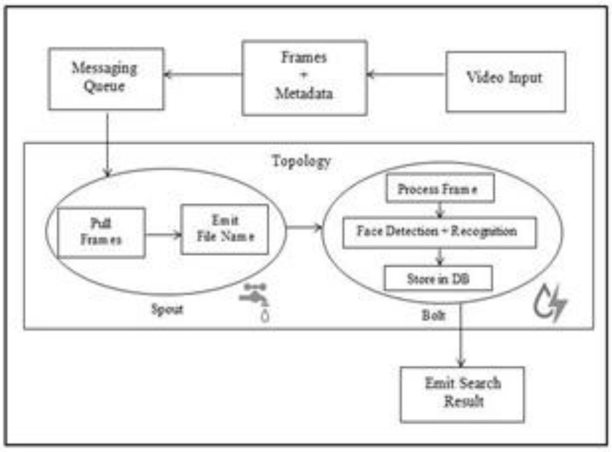

Fig.1 Working of the System

A. Video Capture Module

Installing CCTV camera and monitoring the footage is the first step in any surveillance system. We have seen surveillance systems at various public and private places including schools, banks, colleges, home etc. The video captured via surveillance system is monitored to notice any anomaly or lapse in security. This module is primarily concerned with capturing the video in the format convenient for processing. The resolution and format of video can be set based on the requirement. CCTV videos usually do not contain an audio feed hence they can be easily split into frames for processing. In our methodology we capture the video using OpenCV image processing library in Java programming language called JavaCV. The video is captured via the camera connected to the computing device. Each frame captured is written to the video and also stored as an image file. The video is used for future reference but the processing in our implementation is carried out using the frames stored as images. This way the videos and frames are captured from various cameras covering the whole area for surveillance.

B. Metadata Module

Metadata usually means data about data which means it is the information about the obtained data. This metadata is necessary because it may contain important information such as the time and place the frame was captured, the camera number and the area it covers, and other valuable information. When the frame is processed the metadata in the frame will help in tagging the results to the area or time or date of occurrence. In our module we have used metadata extractor library which extracts metadata from each frame. Other libraries also yield successful results. By this module the metadata will be embedded at the different computing machine where the data is captured.

C. Queueing Module

The raw data that is the frames contained in each of the computing machines need to be integrated and sent to a primary computing machine for processing. The data has to be sent from individual machines to the centralized system for processing. We use the ZeroMQ distributed messaging queue to send the data in a queue using sockets for transmission. The ZeroMQ sockets send the data from the individual computing machines to a queue which acts as a buffer. The data can be pulled from the processing machine whenever it is ready to process further data. This is the module which primarily focusses on transferring the data from different machines to the processing engine. The ZeroMQ server is set up in each of the individual computing machines. In this implementation the ZeroMQ library is integrated with the Java programming language.

D. Processing Module

This methodology uses Apache Storm processing engine for distributed processing of the frames. Programs in storm are implemented primarily in Java but other language features may also be used. This module uses Java to write the storm program. The storm processing module is split into spouts and bolt.

1. Spouts

Spouts are data sources in Storm engine. In this implementation, Storm consists of a ZeroMQ socket from which data that is in the frames is pulled and stored in the processing machine. The spout then emits the name of the frames to the bolt for processing. The implementation of spout is done in Pure Java.

2. Bolts

Bolts are the processing modules in Storm. In this proposed method we implement the bolt in pure Java which would be the processing module. The data emitted from spout is sent to the bolt which uses the filename it receives to open the frame. The frame under consideration is studied based on Haar-cascade classifier to tag the faces in the frame. Then the faces tagged are compared with the existing datasets to identify the persons in the frame. Once identified the metadata of the image is read and stored in the database along with the name of the person identified. The name of the person identified is emitted from the bolt. MongoDB database is used and the MongoDB JDBC driver integrates it with the bolt. Other databases and scripting language can also be used for bolt.

Algorithm:

```
video capture (0)
time interval -t sec
't' sec --------------variable no. frames do {
face_detect ( frame ) ;
if ( yes )
{
face_recognize ( frame ) ;
store data;
}
} while ( frames available) ;
```

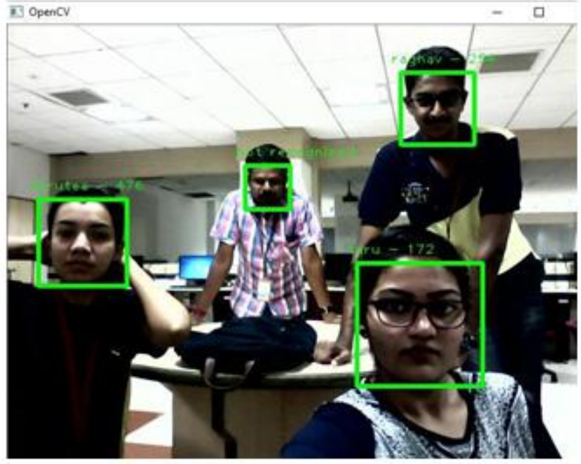

Fig. 2: Test case with recognized and unrecognized faces.

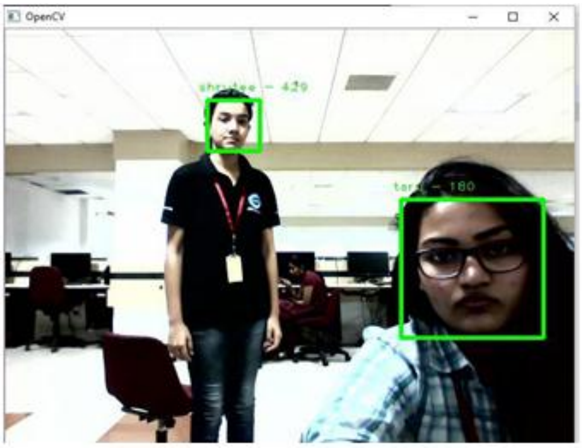

Fig. 3: Test case in which face detection is checked w.r.t proximity from the camera.

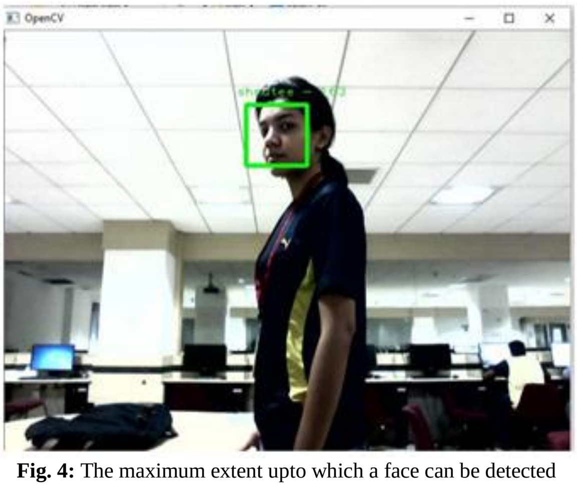

Fig. 4: The maximum extent upto which a face can be detected

4. Conclusion & Future Work

In this paper, a surveillance system which transfers video feeds and identifies a human and tracks the person from it has been proposed. A working demo written in Java was developed which uses ZeroMQ, OpenCV, Javacv libraries and allows instant detection, recognition and tracking of multiple people in video streams.The system makes use of only open-source tools for this project and functions well in real time with very little latency and works in distributed platform. This project has been extremely successful and this technical stack may be used to implement other similar systems for video analytics.

The advantage of automating the surveillance system is immense as the lapse in security systems because of human errors are greatly eliminated, the number of people employed for security systems can be reduced and the manpower could be used for some other useful purposes. The human force could be deployed in securing the glitches in security. The system has been executed with minimal resources and maximum result. If the resources are improved, the chance of improving the system is great. Though the installation and setup of this system is difficult and time consuming, the system once established is easy to run and maintain. The data is replicated in different places based on the replication factor of Apache Storm, avoiding any chances of data loss and corruption.

The system leads to successful automation of surveillance system by tagging, identifying and searching individuals from the surveillance videos by distributed processing using open source tools and the complex technical stack of various components like ZMQ, OpenCV, Apache Storm, MongoDB etc. Hence, it can be said that the output from our study helps in understanding more about the extent to which video surveillance can play a role in influencing the security of people.

Further research can be done in a number of aspects to improve this work. A searching mechanism can be added to this project which will find the current location of a person in live video stream.Improved algorithms used for Image Processing can be imbibed. Libraries of OpenFace can be used instead of OpenCV for improved face detection, tracking , comparison and results. Also the technical stack can be tested for alternatives and the more efficient one can be chosen like for instance, using Apache Kafka instead of ZMQ. Even a way can be found to achieve the same technical stack, extending it to Android and other platforms as well.

References

[1] Dinesh Singh, C. Vishnu and C. Krishna Mohan, “Visual Big Data Analytics For Traffic Monitoring In Smart City”, 15th IEEE International Conference on Machine Learning and Applications, 2016

[2] Eigenfaces Vs. Fisherfaces: Recognition Using Class Specific Linear Projection

[3] H. Bhaskar, “Integrated Human Target Detection, Identification And Tracking For Surveillance Applications,” IEEE Intl’ Conf. On Intelligent Systems, P.467–475, 2012.

[4] H. Moon, P.J. Phillips, Computational And Performance Aspects Of PCA-Based Face Recognition Algorithms, Perception, Vol. 30, 2001, Pp. 303-321

[5] Honghai Liu, Shengyong Chen, Naoyuki Kubota, “Intelligent Video Systems And Analytics: A Survey”, IEEE Transactions on Industrial Informatics, Volume: 9, Issue: 3, Aug. 2013,

[6] J. Lu, K.N. Plataniotis, A.N. Venetsanopoulos, Face Recognition Using LDA-Based Algorithms, IEEE Trans. On Neural Networks, Vol. 14, No. 1, January 2003, Pp. 195-200

[7] J. Lu, K.N. Plataniotis, A.N. Venetsanopoulos, Face Recognition Using Kernel Direct Discriminant Analysis Algorithms, IEEE Trans. On Neural Networks, Vol. 14, No. 1, January 2003, Pp. 117-126

[8] K. Jonsson, J. Matas, J. Kittler, Y.P. Li, Learning Support Vectors For Face Verification And Recognition, Proc. Of The IEEE

International Conference On Automatic Face And Gesture Recognition, 26-30 March 2000, Grenoble, France, Pp. 208-213

[9] M.S. Bartlett, J.R. Movellan, T.J. Sejnowski, Face Recognition By Independent Component Analysis, IEEE Trans. On Neural Networks, Vol. 13, No. 6, November 2002, Pp. 1450-1464

[10] R. Chellappa, C.L. Wilson, S.Sirohey, "Human And Machine Recognition Of Faces: A Survey", Proceedings of the IEEE, Volume: 83, Issue: 5, May 1995

[11] S. Biswas, G. Aggarwal, P. Flynn, "Face Recognition In Low-Resolution Videos Using Learning-Based Likelihood Measurement Model," Intl' Joint Conf. On Biometrics, P.1–7, 2011

[12] S. Srisuk, M. Petrou, W. Kurutach And A. Kadyrov, Face Authentication Using The Trace Transform, Proceedings Of The IEEE Computer Society Conference On Computer Vision And Pattern Recognition (CVPR'03), 16-22 June 2003, Madison, Wisconsin, USA, Pp. 305-312

[13] Samal, A., & Iyengar, P. A. (1992). Automatic Recognition And Analysis Of Human Faces And Facial Expressions: A Survey. Pattern Recognition, 25(1), 65-77.