Abstract—Particulate Face/Movements Detection in a Video will assist security personals in keeping track of all the people entering and leaving a secured zone. The core functionality of Particulate Face/Movements Detection in a Video will allow real-time tracking and recognition of human faces in a video stream. It will have additional features like motion detection, appearance history recording, and tracking of specified person. The main application is securing an area by tracking the human entry/exit in that area is the most generic need of most business enterprise thinking about security.

General Terms: Matlab and OpenCV are used for the simulation of Particulate Face/Movements Detection in a Video.

Keywords: Face Recognition, Hidden Markov Model, Singular Value Decompositions, GWT.

I. INTRODUCTION

Particulate Face/Movements Detection in a Video will assist security personnel in keeping track of all the people entering and leaving a secured zone. The core functionality of Particulate Face/Movements Detection in a Video will allow real-time tracking and recognition of human faces in a video stream. It will have additional features like motion detection, appearance history recording, and tracking of specified person. The main application is securing an area by tracking the human entry/exit in that area is the most generic need of most business enterprise thinking about security.

Our aim is that particulate one face detect from a set of different faces and recognize it and also tracking in real time in image and video source. The target area, where particulate face/movements detection will track people, has sufficient illumination, and our assumption is that People in the scene are moving at normal pace and not running. At least one frame of video stream must capture the complete face in frontal profile, or 15 degree left/right profile.

II. OUR SOLUTION

From the problem definition I have decided to make system with the help of MATLAB track real-time the Particulate face/movements detection in a video. For that first I will detect the face and then real-time track the movement of particulate one faces in a video from video source in MATLAB. The system flow is as shown in figure 1.

A. Stage 1: Source Stage

This stage refers to the source of visual data for the system. The possible components of this stage are CCD camera, MPG video file, sequence of images, etc. Most face recognition systems are tightly coupled with the source, like a camera, which limits the capability of the software. This module will allow the rest of the system to fetch visual data from a variety of sources.

B. Stage 2: Source Transformation Stage

Face extraction algorithms usually process raw visual data. However, most of them have good performance when supplied with visual data that has been preprocessed or modified in some desirable way. This stage handles the task of transforming the visual data as required by the next stage algorithm. Identity Source transformation, which makes this stage transparent between STAGE 1 and STAGE 3, will be applied to those algorithms that do not require any kind of source transformation.

To illustrate the functioning of this stage, let us consider a hypothetical scenario in which the visual source, a CCD camera, is operating at 30 fps. The algorithm in STAGE 3 is not capable of processing this huge amount of data and hence only desires to receive the frame numbers 1, 5, 10, 15, 20, 25, 30. So a module in Source Transformation Stage will filter out the frames that are not required and only pass the desired frames to the next stage.

C. Stage 3: Face Extraction Stage

Fig. 1: System Flow

Fig. 2: face detection using HMM
This stage implements the face extraction algorithms chosen for this project. It will extract human face area from a cluttered background and pass the details of this extracted region to next stage. Two algorithms will be implemented for this stage.

- Neural Network Based face Detection
- AdaBoost Algorithm and Cascaded Detector

D. Stage 4: Pre-recognition Transformation

This stage is similar to Source Transformation Stage except that it performs transformation as required by the face recognition algorithms in STAGE 5.

To illustrate the functioning of this stage, let us consider a hypothetical scenario in which Eigen faces[6] based recognition scheme is implemented in STAGE 5. Eigen faces scheme cannot recognize faces having size not same as the training sample image size i.e: Eigen faces algorithm is not scale invariant. Hence, the face extracted by STAGE 3 algorithm must be scaled to match the size of training samples used to train the Eigen faces algorithm. Hence a PRE-RECOG Transformation module, that performs scaling, can be used in this stage to increase the robustness of algorithm implemented in STAGE 5.

E. Stage 5: Face Recognition Stage

This stage implements the face recognition algorithms chosen for this project. It will identify the faces extracted by STAGE 3 by performing a match against know face database. Two algorithms will be implemented for this stage.

- Subspace Linear Discriminate Analysis (LDA)
- Face Recognition using Gabor Wavelet Transform

F. Stage 6: Presentation Stage

The result of face tracking/recognition will be made available by this stage. The output format can vary according to the requirement of user. The most general form of STAGE 6 output is depicted in.

![Video Tracking](image)

**Fig. 2:** Output format for result of face tracking/recognition

1) Unified Training Interface

The algorithms implemented in STAGE 3 and STAGE 5 requires some form of training before they can become capable of performing the face tracking/recognition task. The training method of these algorithms varies considerable. But most of these algorithms take images containing faces as training data. Hence the Unified Training Interface, not the part of any stage, handles the task of training the algorithms with appropriate data.

2) Software Development Methodology

Groups were tools will be extensively used for collaboration of the project design and development. Visual studio and OpenCV will be used for collaboration on code development; Matlab will be used for initial stage coding. C++ (compiler) will be the main programming language for development. Several libraries OpenCV, Emgu C++ libraries, etc

III. EXPECTED RESULTS

In order to we operate an interface that shows you the face detection state in real time by marking the face detection. Many of the faces detected in real time but particulate one face in specified person cannot detect. So we implement HMM based system for particulate one face detection in a real time and recognize it. We used a small number of quantized Singular Values Decomposition (SVD) coefficients as features describing blocks of face images. This makes the system very fast. LDA and GWT algorithms will be implemented for face recognition. And also track it in real time. When person moving in a real time video steam the square boxes also track it in real time

IV. CONCLUSIONS AND FUTURE IDEAS

Particulate face/movements detection in a video cannot be used as complete software based solution for security in this proposed form. However, it will mark a beginning of an open source tool that will help business enterprise (or anyone who requires to secure an area) to implement a low cost solution. Anyone who wants to track the entry/exit of humans in a secured area (a generic problem in security field) can use Particulate face/movements detection in a video .this is an open source project and hence it will encourage community participation for research and development in creating complete software based solution for securing an area.

This is the Hidden Markov Model (HMM)-based face recognition system. And using this system will allow real time tracking and recognition of human faces in a video stream.

In the future, we will focus on the use of larger and more complicated databases to test the system. For these complicated databases it is simply expected that all the previous methods will not repeat such efficiency reported in the papers. We will try to improve the feature extraction and the modeling of the faces. The use of 2-D HMM or more complicated models may improve the system performance.

ACKNOWLEDGMENT

Thanks to my Family members my father, mother, grandfather, grandmother and many more for their support to carry out this Research. Thanks, to my brother for always supporting during my work.

Finally, thanks to God, for helping me out at difficult times.

V. REFERENCES

A Review on Particulate Face/Movements Detection in a Video
(IJSRD/Vol. 1/Issue 12/2014/002)


[8] Aamer S.S.Mohamed, Ying Weng, Stan S Ipson, Jianmin Jiang “Face Detection based on Skin Color in Image by Neural Networks”.

[9] Christophe Garcia and Manolis Delakis “A Neural Architecture for Fast and Robust Face Detection”, Department of Computer Science, University of Crete, P.O. Box 2208, 71409 Heraklion, Greece,IEEE