**Design Document**

**Version 1.0**

## Video Management System API for

Face Detection and Face Recognition

DOCUMENT NO : 01

VERSION : 1.0

CONTACT : 8373048176

Mail : [shilpa.k@divsoftsolutions.com](mailto:shilpa.k@divsoftsolutions.com)

DATE : 09/12/19

1

**Disclaimers**

The information contained in this document is the proprietary and exclusive property of Divsoft Solutions India pvt. ltd except as otherwise indicated. No part of this document, in whole or in part, may be reproduced, stored, transmitted, or used for design purposes without the prior written permission of Divsoft Solutions India pvt. ltd .

The information contained in this document is subject to change without notice.

The information in this document is provided for informational purposes only. Divsoft Solutions India pvt. ltd specifically disclaims all warranties, express or limited, including, but not limited, to the implied warranties of merchant-ability and fitness for a particular purpose, except as provided for in a separate software license agreement.

**Privacy Information**

This document may contain information of a sensitive nature. This information should not be given to persons other than those who are involved in the Video Management Systemproject or who will become involved during the life cycle.

**Trademarks :**

VMS

**Version History**

|  |  |  |  |
| --- | --- | --- | --- |
| **REVISION CHART** | | | |
| **Version** | **Author(s)** | **Description of Version** | **Date** |
| 1.0 | Shilpa Kapas | Computer vision and machine learning based face detection and face recognition system. The project delivers an implemented attendence system for office environment. | 09/12/2019 |

2

**Document Owner**

The primary contact for questions regarding this document

Author : Shilpa Kapas

Project Name : Video Management System

Phone : 8373048176

Mail : shilpa.k@divsoftsolutions.com

**Document Approval**

Document Name:

Publication Date

Project Number:

Prepared by:

Approval:

###### Name and Organization

Concurrence:

###### 

**3**

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| [**1**](#_bookmark1) | [**Introduction**](#_bookmark1) | **5** |
|  | [1.1 Purpose of this document](#_bookmark1) | 5 |
|  | [1.2 Document Overview](#_bookmark1) | 5 |
|  | [1.3](#_bookmark1) [Relatinoship to Other](#_bookmark2) Plans | 5 |
|  | [1.4](#_bookmark1) [Methodology,](#_bookmark2) Tools, and Techniques | 5 |
| [**2**](#_bookmark3) | [**Design Overview**](#_bookmark3) | **6-7** |
|  | [2.1 Background Information](#_bookmark3) | 7 |
|  | [2.2 System Evolution Description](#_bookmark3) | 8 |
|  | [2.3 Technology Forecast](#_bookmark3) | 9-11 |
|  | [2.4 Proposed Process](#_bookmark3) | 11-12 |
|  | [2.5 Risks](#_bookmark4) | 12 |
|  | [2.6 Issues](#_bookmark4) | 12 |
|  | [2.7 Dependencies](#_bookmark5) | [1](#_bookmark5)2 |
| [**3**](#_bookmark6) | [**Scope of Work**](#_bookmark6) | [**1**](#_bookmark6)**3** |
|  | [3.](#_bookmark6)1 Software Interface | [1](#_bookmark6)3 |
|  | 3.2 Result | 13 |
|  | 3.3 References | 14 |

#### **1Introduction :**

**1.1 Purpose of the document :**

Authentication is a significant issue in system control in computer based communication.Human face recognition is an important branch of biometric verification and has been widely used in many applications, such as video management system, human-computer interaction, and door control system and network security.

This project was an attempt at developing an face detection and recognition system using modern computer vision technology. The project delivers an implemented attendence system for office environment.The system will record the attendance of the person in office environment automatically and it will provide the facilities to the faculty to access the information of the employee easily by maintaining a log for clock-in and clock-out time.

**1.2 Document Overview :**

Outline the main sections in this document, e.g.:

1. Chapter 1 – Introduction

2. Chapter 2 – Design Overview and detailing

3. Chapter 3 – Scope of work

**1.3 Relationship to Other Plans :**

Face detection and recognition APIs are directly relative to the project Video Management System(VMS) user interface.

**1.4 Methodology and Tools :**

* This face recognition API uses of the deep learning tools from the dlib C++

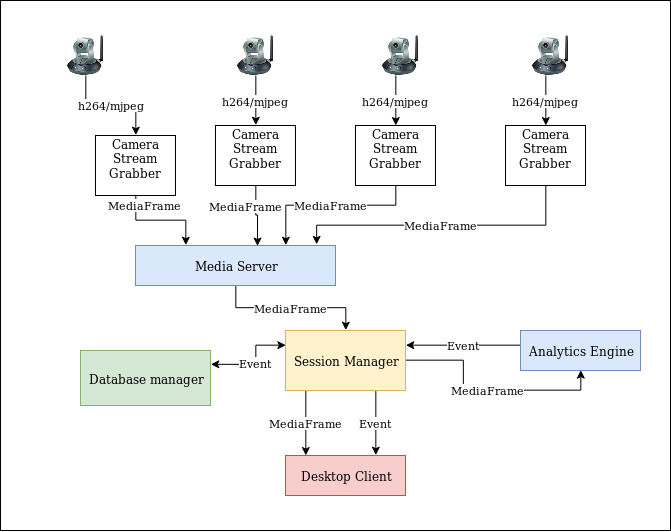
Library and opencv library.

* This API uses dlib\_face\_recognition\_resnet\_model\_v1(responsible for recognition) ,shape\_predictor\_5\_face\_landmarks model(face landmarking model to align faces to a standard pose) which are freely available in the dlib web site.

**5**

**2.0 Design Overview**

This design describes how the components are relative and interacting to each other and how the whole process is working

 Architecture of VMS with analytics

Components are :

1. Media Server

2. Session Manager

3. Analytics Engine

4. Desktop Client

5. Database Manager

**6**

**Process Workflow :**

**1. Media Server :**

i. Grabbing the camera streams from network cameras,

ii. managing the recording,

**2. Session Manager :**

i. Maintaining user sessions

ii. Maintaining media servers

iii. Authentication and authorization of application users.

iv. Relay user controls to the respective media servers.

v. Storing and loading application state during application launch and termination.

(Maintaining application state)

**3. Analytics Engine :**

i.Session Manager serving motion jpeg frames to the analytics engine.

ii.Analytics engine serves frames after doing specific task(recognize faces)

return events(as jpeg image) to the session manager

**4. Desktop Client :**

i.Desktop clients basically user interface.It handles **add camera** to a particular

media server,**delete camera**,**start analytics** etc.

ii.**Pause/Resume** streaming of a particular camera

**5.Database Manager :**

i.It is responsible for keeping the recorded clips mapped to the database table as well

as deleting old records upon local storage overflow.

ii. It also keeping records of generated event details(recognized peson’s name,gender,id,

date of birth,time of event generation)

**2.1 Background Information :**

Now the background information of face recognition system

The first attempts to use face recognition began in the 1960’s with a semi-automated system. Marks were made on photographs to locate the major features; it used features such as eyes, ears, noses, and mouths. Then distances and ratios were computed from these marks to a common reference point and compared to reference data. In the early 1970’s Goldstein, Harmon and Lesk created a system of 21 subjective markers such as hair colour and lip thickness. This proved even harder to automate due to the subjective nature of many of the measurements still made completely by hand.

Fisherand Elschlagerb approaches to measure different pieces of the face and mapped them all onto a global template, which was found that these features do not contain enough unique data to represent an adult face.

**7**

**2.2 System Evolution Description :**

1.Firstly analytics engine getting frames from session manager.

2.After getting frame resize it.

3.Transfer image into grayscale image

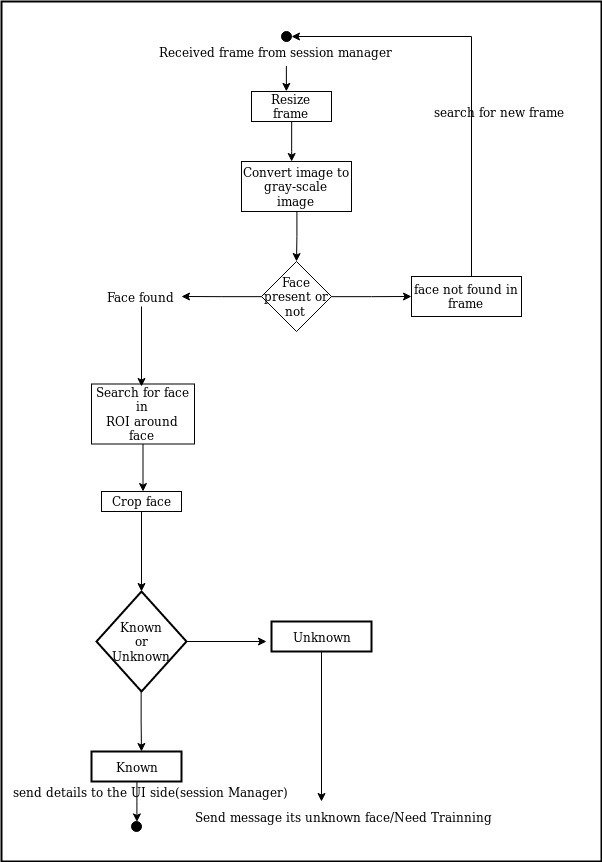
4.Find face is present or not

5.If face present crop it and try to recognize it.

6.It face not present in frame then take next frame

7.If face present and recognition is done then send the session manager the details of person(name,id,gender etc)

8.If fails recognition then send Unknown person need training

****

**8**

**2.3 Technology Forecast :**

Now comes the point, detailing about technology used for facial recognition.

Two important external libraries:

1. Opencv

2.dlib

**Load two model**

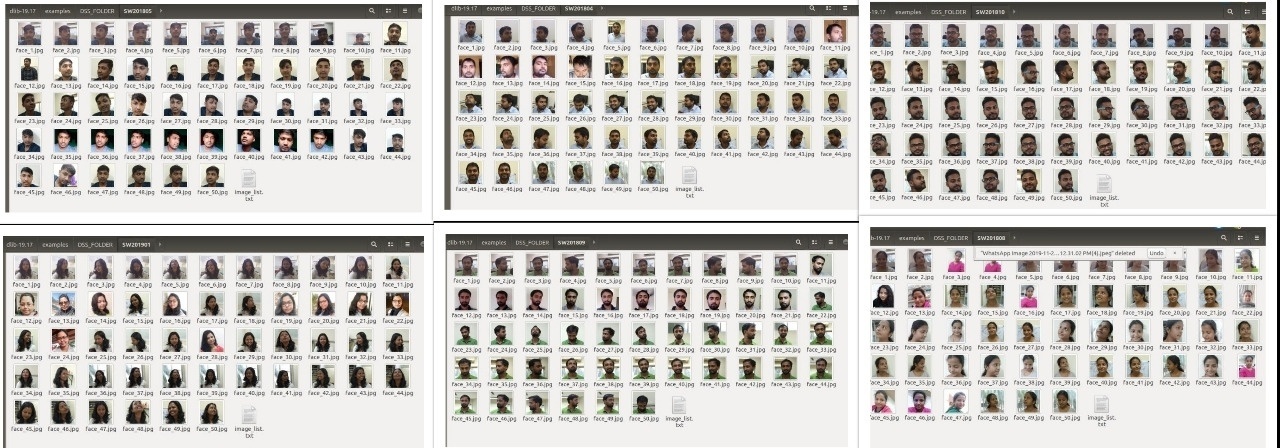
1. shape\_predictor\_5\_face\_landmarks.dat (For facial landmarking)

2. dlib\_face\_recognition\_resnet\_model\_v1.dat(load the DNN responsible for face recognition.)

3 . Required template required for Deep neural network

**Trainning Phase :**

In training phase first reads the images from dataset for every person.



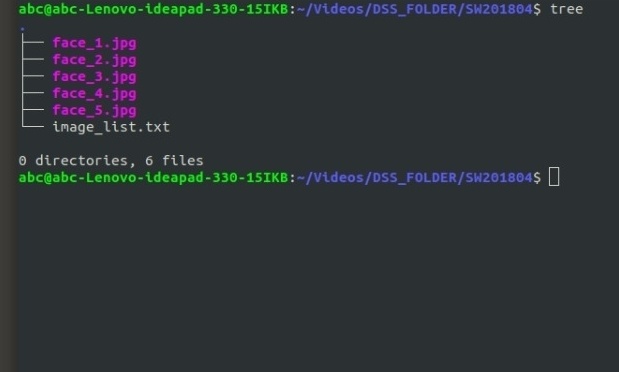
Our dataset:

**Step 1**. Store few photos of a each person with different angle(Left turn,right turn,up ,down etc) in a folder.Folder name should be same as the ID of employee.Store name of every images in a text file.It is for if any image missing in the directory it will show message as example :face\_41.jpg missing in the folder.

**Directory should like this**:Parent\_Folder\_name/Employee\_Id/All\_faces.jpg

**As example**:DSS\_FOLDER/SW201808/image\_list.text(which contains all image name)

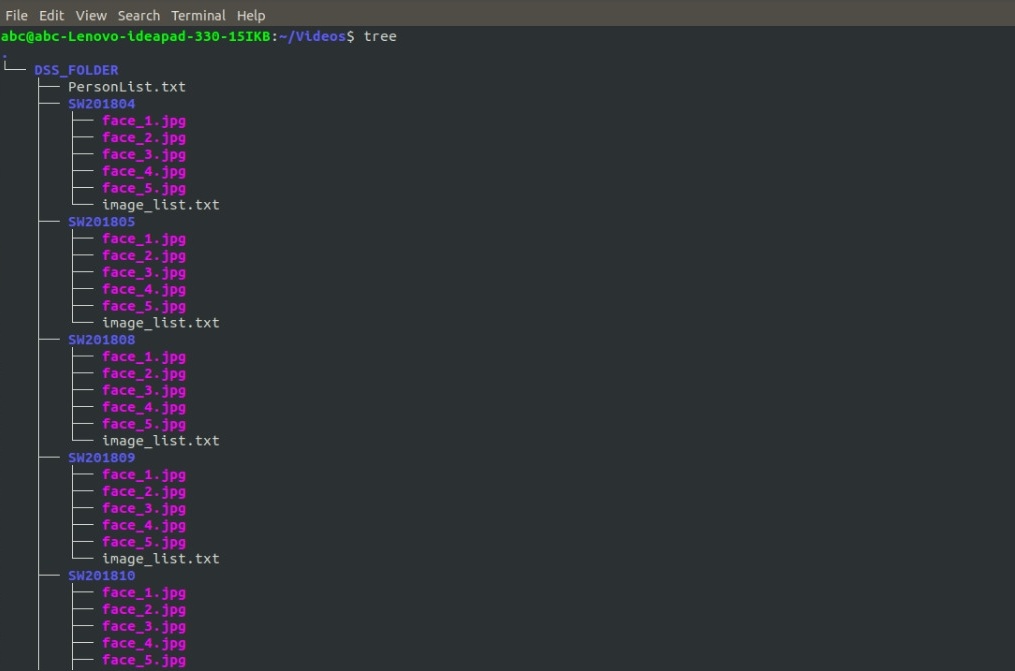
**9**



Parent folder contains all the 6 child folders as discussed above and one text file which containing every person’s name,id,gender,date of birth.

As example :DSS\_FOLDER/SW201808

DSS\_FOLDER/persons\_list.txt



**Step 2**. Read every images and try to find faces from the images.

**Step 3**. Detect faces in image.

**10**

**Step 4.** Create the 128-d embeddings for each face in the dataset.Means store the 128

facial feature(feature vectors) for every person’s every image with person’s details (person’s name,id,date of birth,gender) in a .dat file.



Stucture of storing face data

**Recognition** **Phase :**

Recognition phase consists of four sub phases

1. Detect faces

2.Compute 128-d face embeddings to quantify a face

3.Train a Support Vector Machine (SVM) on top of the embeddings

4.Recognize faces in images and video streams

After getting a new frame from session manager side try to find out if frame contains

face or not.If contains face the detect faces using face detector of dlib.Detects the presence and location of a face in an image, but does not identify it.After detection push the details(128d facial features vector) of facial feature in a datastructure(matrix) of input face and crop the detected face.

Compare it to the present face data means with trained data.Among pretrained data which is closest(Euclidean distance) to the input image taking the name and other detail as recognized person.Cropped face image is send to the session manager as a event and session manager stores it’s time and other details in database.

**2.4 Proposed Process :**

When image quality is taken into consideration, there is a factor that influence the system’s accuracy.

It is extremely important to apply various image pre-processing techniques to standardize the image

to a face recognition system.Most face recognition algorithms are extremely sensitive to lighting

conditions, so that if it was trained to recognize a person when they are in a dark room, it probably

**11**

won’t recognize them in a bright room. This problem is referred to as "luminatio dependency

and there are also many other issues, such as the face should also be in a very consistent position

(such as the eyes being in the same pixel coordinates), consistent size,rotation angle, hair and makeup,

emotion (smiling, angry, etc), position of lights (to the left or above, etc).

**2.5 Risks :**

1. Environment
2. Light conditions

**2.6 Issues :**

|  |  |  |
| --- | --- | --- |
| Ref | Issue | Action |
| 1. | Time requirement | 1.Resize frame actually downsampling causes image size lesser.So it needs less time to read than original size  2.Convert RGB image to gray-scale. |

##### Table 2 — Issue

**2.7 Dependencies :**

|  |  |  |
| --- | --- | --- |
| Ref | Dependency | Action |
| 1. | Library dlib | Install dlib-19.17  Load two model 1.**shape\_predictor\_5\_face\_landmarks.dat** (For facial landmarking)  2.**dlib\_face\_recognition\_resnet\_model\_v1.dat(**load the DNN responsible for face recognition 3 . Required template required for Deep neural network |
| 2. | Library Opencv | Opencv 3.4.1  Include required header of opencv |

##### Table 4 — Dependencies

##### **12**

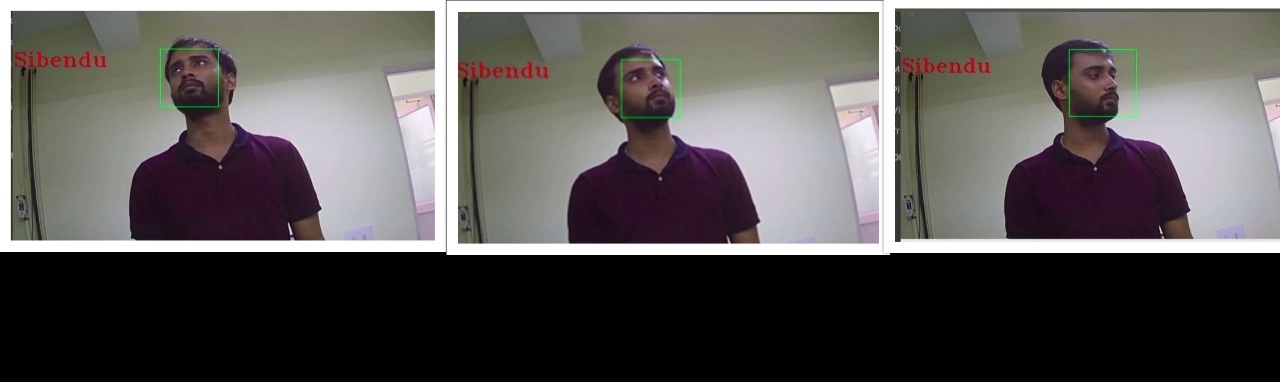
##### **3.0 SCOPE OF WORK** : There are numerous scope of facial recognition. They can be segmented into blacklist and whitelist applications. Blacklist applications include the ones related to security & surveillance and identification of criminals. The all other applications such asattendance tracking, access control and others fall under the category of whitelist applications.

|  |  |
| --- | --- |
| End Use | Application |
| **Offices** | Allow attendance tracking of the employees in time and out time |
| **Education** | Allow attendance tracking of the students and entry to labs |
| **Construction** | Control access to specific point at a site |
| **Aviation** | Paperless travel at airports |
| **Warehouse** | Control process to provision entry and exit of vehicles |
| **Government** | Helps to Identify missing children |

**Result :**

* This system has real-time face detection and recognition running at about 12FPS
* The System is able to handle angle up to about 15 to 20 degree in both negative and positive direction with partial changes in appearance (with glasses) closed eye etc.
* The recognition performs reasonably well in different pose and lighting variation

but lacks robustness for extreme condition

****

**3.1 Software Interfaces :**

For communication with User interface/Session manager analytics engine required native interfacing.Needs to load some native library(JNI) for interfacing two different language

APIs.

**13**

**References :**

<http://dlib.net/face_recognition.py.html>

<https://en.wikipedia.org/wiki/Facial_recognition_system>

<https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html>

<https://towardsdatascience.com/facial-recognition-using-deep-learning-a74e9059a150>

**14**