1. INTRODUCTION

1.1 Problem definition

To Control Media player using Image Processing which plays and pause the video by detecting face.

1.2 Aim and objective of the project

1. To develop advanced media player, which is Free from the hustle of keyboard and mouse.
2. To pause the video as soon as the user's face is not detected without much latency.
3. To resume the video at the point from where they missed.

1.3 Scope and limitation of the project

Scope:

We can use this software in home, college, offices etc.

Limitations:

This system is detected only a single face.

1.4 TimeLine of the project

We started to gather the necessary information for the project from mid of July 2018. By the end of August, we had finalized the topic for the project. We had completed software requirement analysis by the mid of September 2018 which encompasses both system and software requirement gathering. By the end of December 2018, we had completed planning and design of the project. Based on design prepared in the previous stage, we started coding by the mid of January 2019. The coding part was completed by the mid of February 2019. After completion of the coding part, an important par-t of software development life cycle is software testing. We started testing the modules. The testing phase was carried out in the last week of February 2019. By second week of March 2019. After completion of the project, we tested it again and cross checked it with the design.

1.5 Project Management Plan

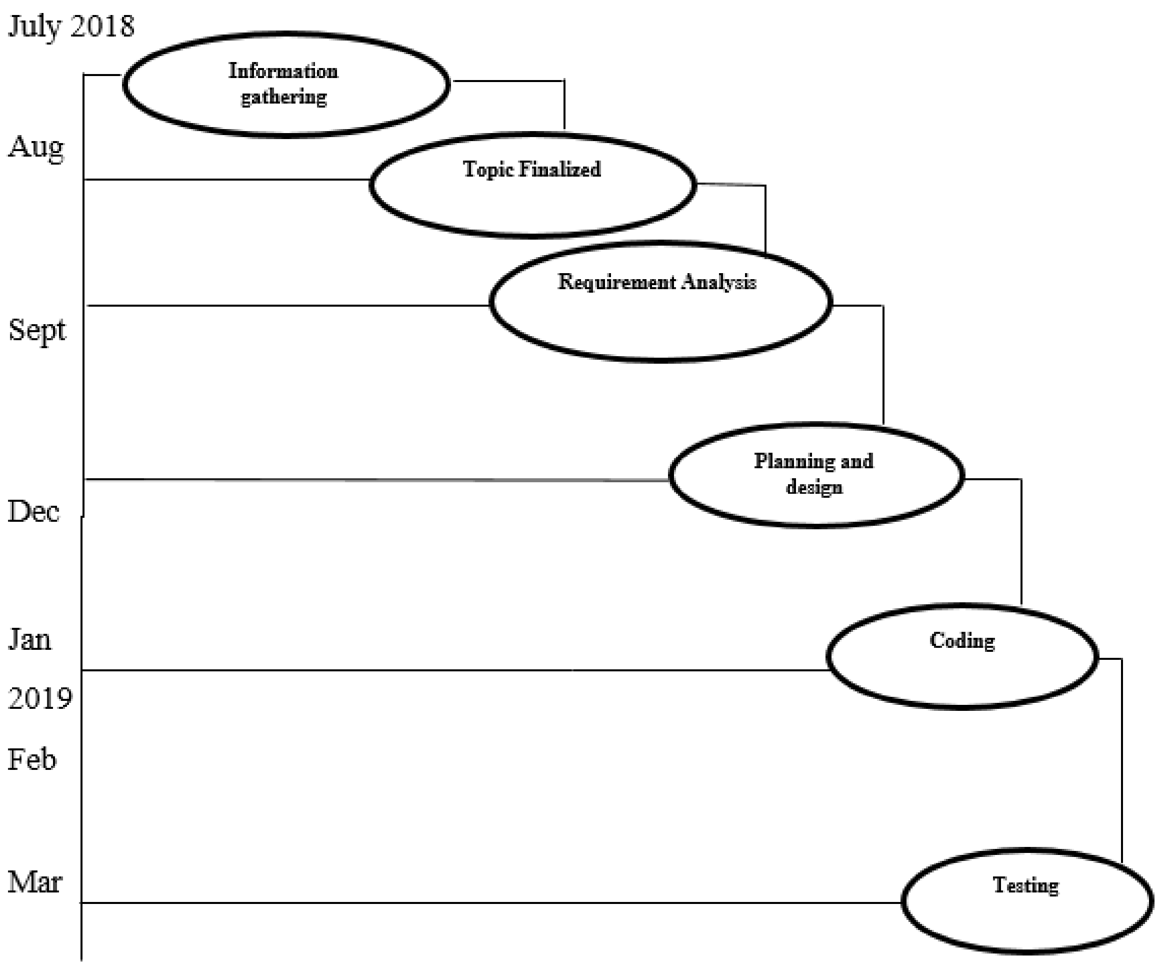


Fig. Timeline of project

1.6 Project Cost

In this project the Cost Estimation based on COCOMO (Constructive Cost Model) the formula for this Model is follows: -

Effort = Constant × (Size) scale factor× Effort Multiplier

Effort in terms of person-months

Constant: 2.45 in 1998 based on Organic Mode

Size: Estimated Size in KLOC

Scale Factor: combined process factors

Effort Multiplier (EM): combined effort factors

The basic COCOMO equation take the form

Effort Applied (E)=a^b (KLOC)b^b[man-month]

Development Time (D)=c^b(Efforts Applied) d^b[months]

People required (P)=Effort Applied/Development Time [Count]

Where,KLOC is estimated number of delivered Lines (expressed in thousands) of code for project.The coefficient a^b,b^b,c^b and d^b are given in the following table:

Software project	a^b	b^b	c^b	d^b
Organic	2.4	1.05	2.5	0.38
Semidetached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Organic Mode

Effort = 2.4*(0.22)^1.05 = 0.489

Development Time = 2.5 * (0.489)^0.38 = 1.904

People Required = 0.489 / 1.904 = 0.256

2. BACKGROUND STUDY AND LITERATURE OVERVIEW

2.1 Literature Overview

PetcharatPattanasethanon and CharuaySavithi [1] proposed a Human Face Detection and Recognition using Web-Cam. In this paper the illuminance insensitivity that reflects the angle of human facial aspects occurs once the distance between the object and the camera is too different such as animated images. This has been a problem for facial recognition system for decades for this reason, our study represents a novel technique for facial recognition through the implementation of Successes Mean Quantization Transform and Spare Network of Winnow with the assistance of Eigen face computation. After having limited the frame of the input image or images from Web-Cam, the image is cropped into an oval or eclipse shape. Then the image is transformed into greyscale color and is normalized in order to reduce color complexities. We also focus on the special characteristics of human facial aspects such as nostril areas and oral areas. After every essential aspect sarescrutinized, the input image goes through the recognition system for facial identification. In some cases where the input image from the Web-Cam does not exist in the database, the user will be notified for the error handled. However, in cases where the image exists in the database, that image will be computed for similarity measurement using Euclidean Distance measure from the input image. But in this paper checking the current face into stored database so it taking so much time for recognizing the image.

S.V. Viraktamath, MukundKatti, Aditya Khatawkar&Pavan Kulkarni [2] proposed a Face Detection and Tracking using OpenCV. In this paper used ADABOOST Algorithm is use to Detecting a faces. An application for automatic face detection and tracking on video streams from surveillance cameras in public or commercial places is discussed in this paper. Prototype is designed to work with web cameras for the face detection and tracking system based on open source platforms OpenCV. The system is based on AdaBoost algorithm and abstracts faces Haar-Like features. This system can be used for security purpose detect and track the face. A program is developed using OpenCV that can detect people's face and also track from the web camera. Face detection is a process, which is to analysis the input image and to determine the number, location, size, position and the orientation of face. Face detection is the base for face tracking and face recognition, whose results directly affect the process and accuracy of face recognition. The common face detection methods are: knowledge-based approach, Statistics-based approach and integration approach with different features or methods. The knowledge-based approach can achieve face detection for complex background images to some extent and also obtain high detection speed, but it needs more integration features to further enhance the adaptability. Statistics-based approach detects face by judging all possible areas of images by classifier, which is to look the face region as a class of models, and use a large number of “Face” and “non-face” training samples to construct the classifier. The method has strong adaptability and robustness, however, the detection speed needs to be improved.

Swapna Agarwal and SaiyedUmer [3] proposed a Media Player controlled by Facial Expressions and Gestures. In this paper we represent a new technique to interact with the computer in a non-tangible way. Specifically we have designed a Media Player system controller by Facial Expressions and Gestures (MP-FEG). We detect and track one landmark point on the finger and 18 landmark points on the lips to capture the movement of the finger and the lips of the user. The movement patterns are classified into hand gestures and facial expressions using support vector machine (SVM).Our main purpose is to find a non-tangible way to interact with the computer and for this we have performed experiment to verify whether the facial expressions and the hand-gestures can be used to give command to the computer, specifically for controlling a media player system in the real time situation. The proposed MP-FEG system has two parts, front-end and back-end. For this system, we utilize Mat-lab graphical user Interface (GUI) to design the front-end. The front-end contains a panel where the video gets displayed and other utilities such as a ‘list box’ that contains a set of videos to select from, pushdown buttons for giving such instructions as ‘Play’, ‘Pause’, ‘Exit’. The system in the back-end captures the video of the frontal face and/or the finger of the user using a low-resolution webcam. The back-end system also contains module to recognize the hand-gestures and the facial expressions. To control the different functions of MP-FEG we have selected seven types of hand-gestures and three types of facial expressions. In this project Mat-lab is used but we are developing a project without using Mat-lab and supporting for new advanced media player system.

2.2 Critical appraisal of other people’s work

S.V.Viraktamath, MukundKatti, Aditya Khatawkar & Pavan Kulkarni proposed a “Face Detection and Tracking using OpenCV. The system which have less robust in face detection, they should use different images to identify face. We need to improve our system and try to been more robust algorithm for face detection even in we create a live camera video and detect face continuously. In this system we need to create a one own media player which plays the any video and support the face detection window.

2.3 Investigation of current project and related work

This project concentrate on problem of previous media player such as you need to be always well equipped with the devices like keyboard and mouse without which you will be not able to control the media player. And the most important point was wastage of time for dragging back to the point where you left the video in case you forgot to pause the video in case of emergency call or any other work.

Controlling media player using image processing overcomes all of the above mentioned problem like you need to be equipped with keyboard and mouse especially for controlling media player like pause and play of video. It stops the time wastage of dragging back the video where you missed in case user forgot to pause. There would be the less wastage of time and lesser the effort of user like personally pausing or playing of video while you are not watching it.

3. REQUIREMENT ANALYSIS

A. Functional Requirements

1. External Interface Requirements

a. User interface Requirement:

- This system requires Python Editor.
- This system requires Web Camera.

b. Hardware interface requirement:

- RAM: 4GB
- Hard disk: 1TB
- Processor: C2D 2.0 GHz or above

B. System Requirements

1. Operating System Requirements

- 64 bit Windows OS.

2. Application or web server Requirements

- This system requires Python 3.5.

3. Technology requirements

- OpenCV, Tkinter, dlib

4. SYSTEM DESIGN

a. Architectural Design:

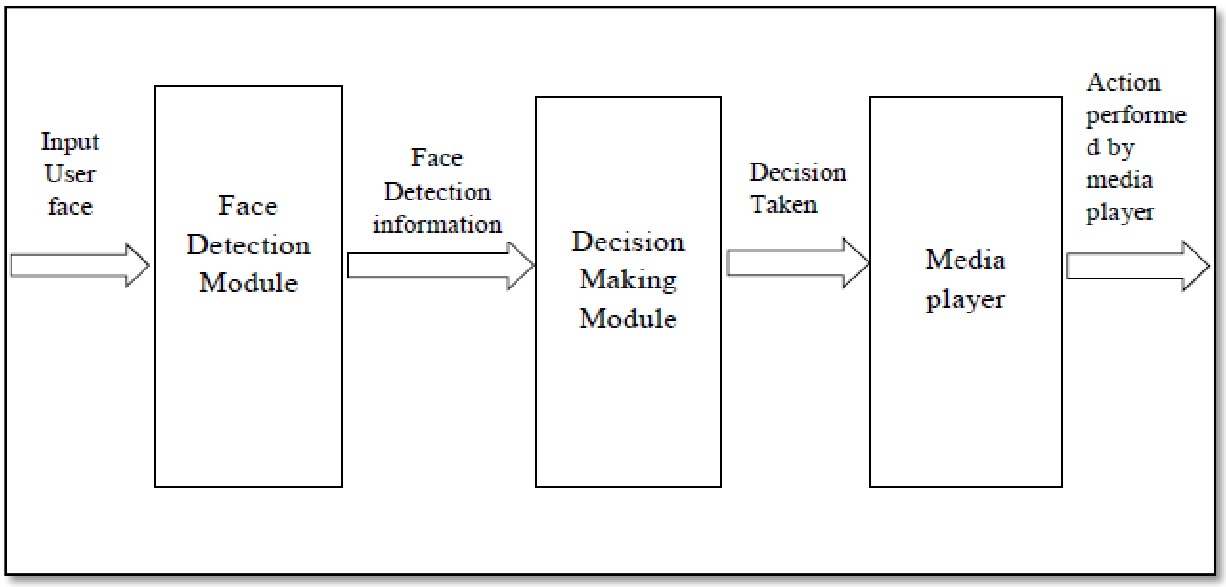
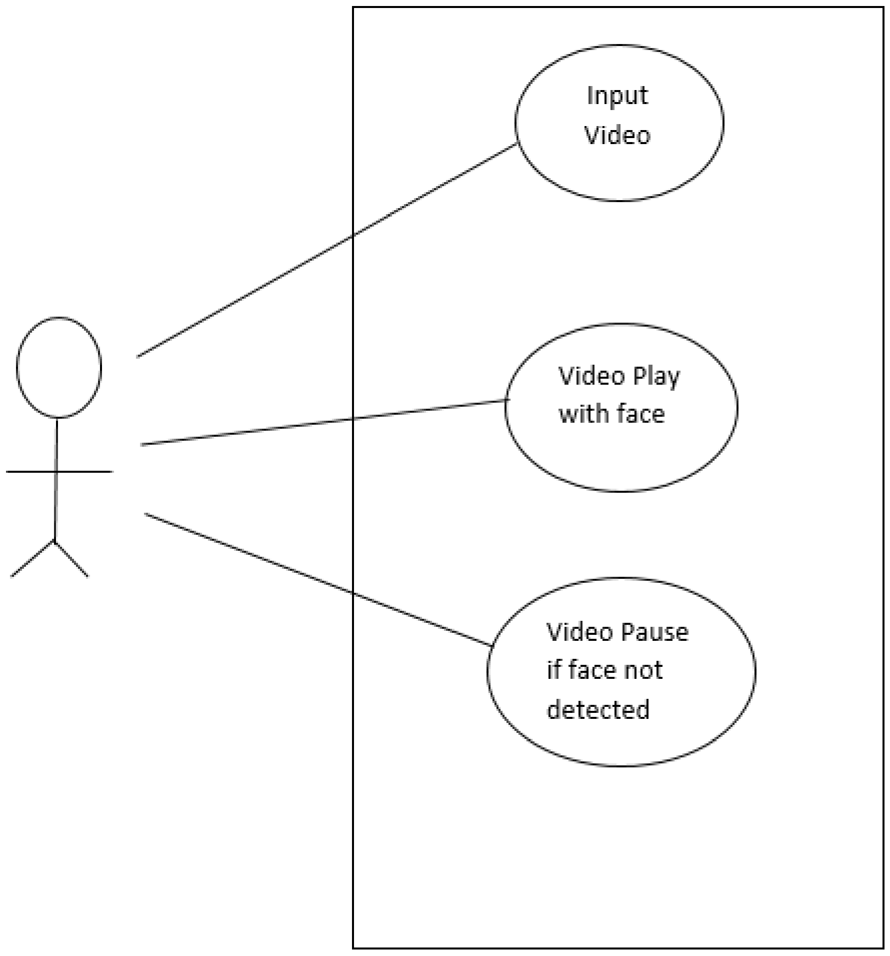


Fig. Architectural Design

b. User Interface Design:



c. Algorithmic description of each modules:

1. Media Player:

    a. Get started with Media player
    b. Choose the video which you want to play.
    c. Load the video.

2. Face Detection:

    a. Open the Web Camera.
    b. Capture the frames.
    c. Detect the face.
    d. Track the face.
    e. If face is detected
        then play the video.
    f. If face is not detected
        then pause the video and start detecting face.

d. System Modeling:

1. Dataflow Diagram:

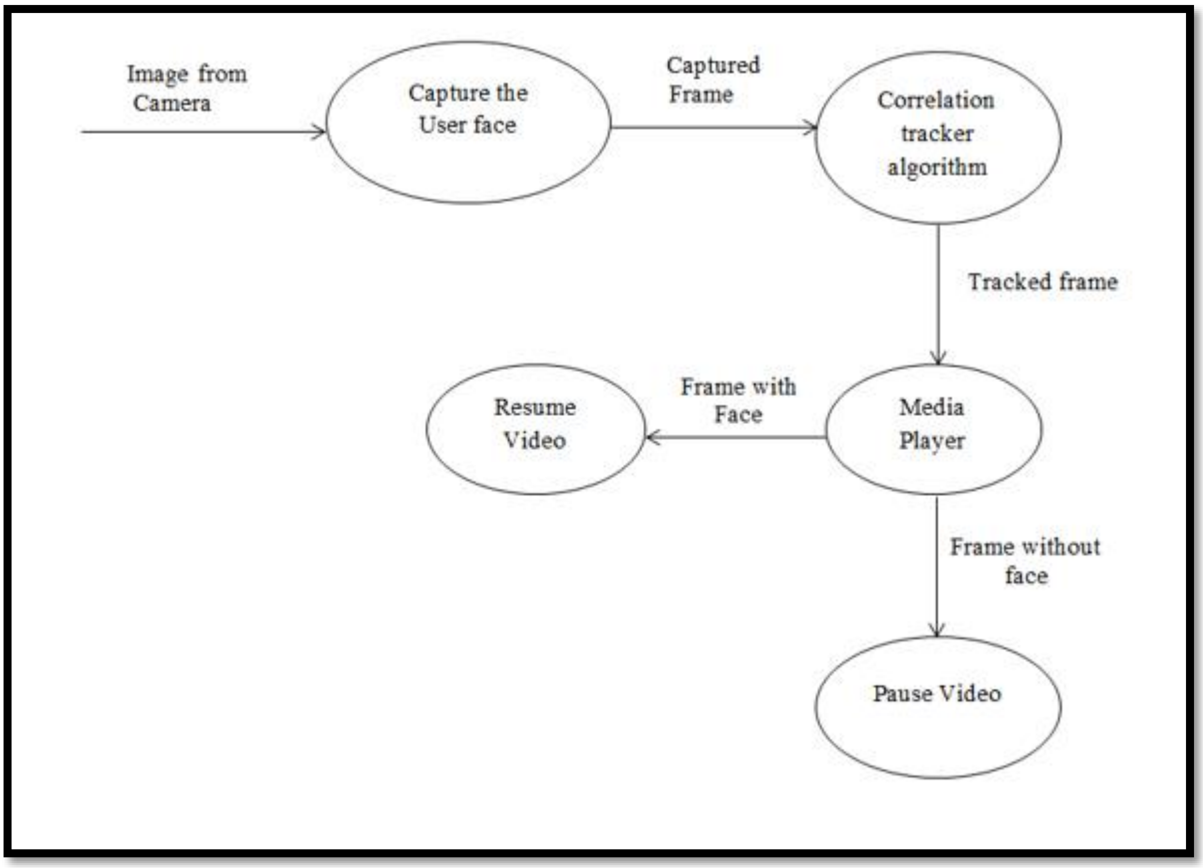


Fig. Data Flow Diagram

2. Sequence Diagram:

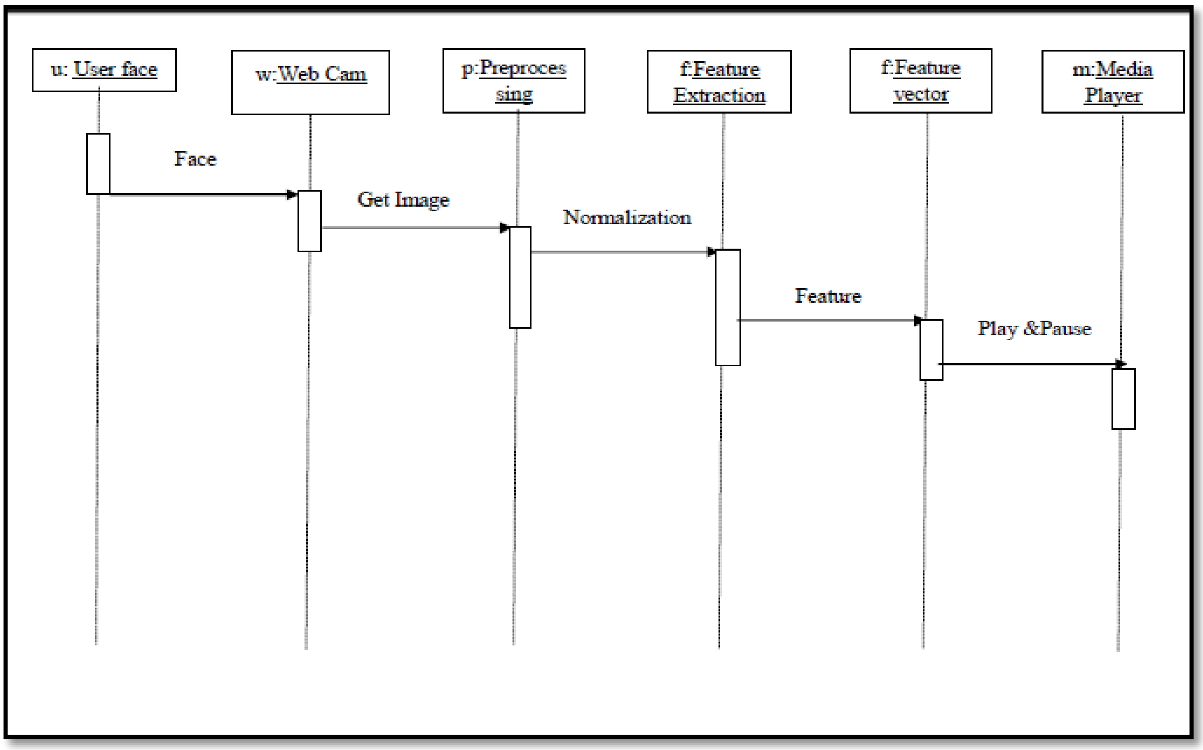


Fig. Sequence Diagram

3. Activity Diagram:

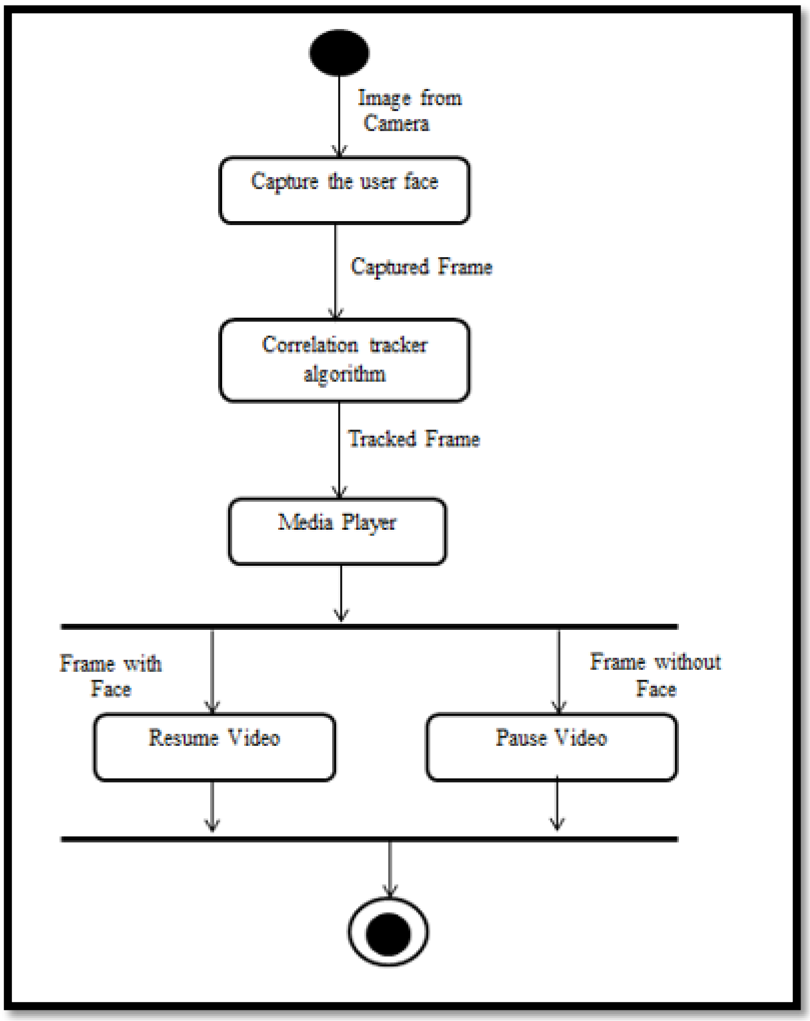


Fig. Activity Diagram

g

5.IMPLEMENTATION

a. Environmental Setting for Running the Project:

Steps to setup environment variable for Python:

1. Click start, then Control Panel, then System.
2. Click Advanced, then Environment Variable.
3. Add the location of the bin folder for the PATH variable in System Variables.
   Then following is a typical value for the Python
   Path variable:
   C:\Users\USER\AppData\Local\Programs\Python\Python3.5.1\Scripts

b. Detailed Description of Methods:

Face Detection:

Get image from video stream. Divide image into two parts and assign Face Detection. Face detection is a process, which is to analysis the input image and to determine the number, location, size, position and the orientation of face. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. A window of the target size is moved over the input image, and for each subsection of the image the Haar-like feature is calculated. Now we use Open-cv in which number of pre-trained classifiers available for face, eyes and smiles etc. After that create face detector in Open CV for that we need to load required xml classifiers. Then load our input image or video in Grayscale mode. Now we find the faces in the image. If the faces are found it returns position of detected face as Rect(x, y, w, h). After detection of face video will be play it.

Play and Pause:

When the Face detection module detect the face then it send output to the Decision Making Module and this module taking the decision information if the face is detected then keep video playing and if the face is not detected then pause the video and this send to media player.

c. Implementation Details:

1. def OnOpen(self):

In this method get the directory path then we check file is present or not using os.path.isfile() method if file is present then print the file.

2. def detectAndTrackLargestFace(self):

In this function Haar cascade algorithm is used to detect the face. Open the camera capture the user face video. We use cv2.namedWindow() to create face detection window. This method also continuously detecting user face. The face detected frame is passed to the correlation_tracker algorithm which is been provide by the dlib library, this library helps to keep a track on the detected face. We have used the dlib library because opencv only provides the function to detect the face from the incoming frames not keeping track on it.

3. def OnPlay(self):

When face is detected then continuously playing a video.

4.def OnPause(self):

When face is not detected then immediately pause the video.

5. def OnStop(self):

The function indicates that when playing a video the player should play for video time limit. Whenever video reach the time limit then automatically stop the video.

6. INTEGRATION AND TESTING

Integration Testing is a logical extension of Unit Testing. In its simplest form, two units that has already been tested are combined into a unit module. It means this refers to an integrating multiple unit modules into one unit.

This idea is to test combinations of pieces and eventually all the modules making up a process are tested together.

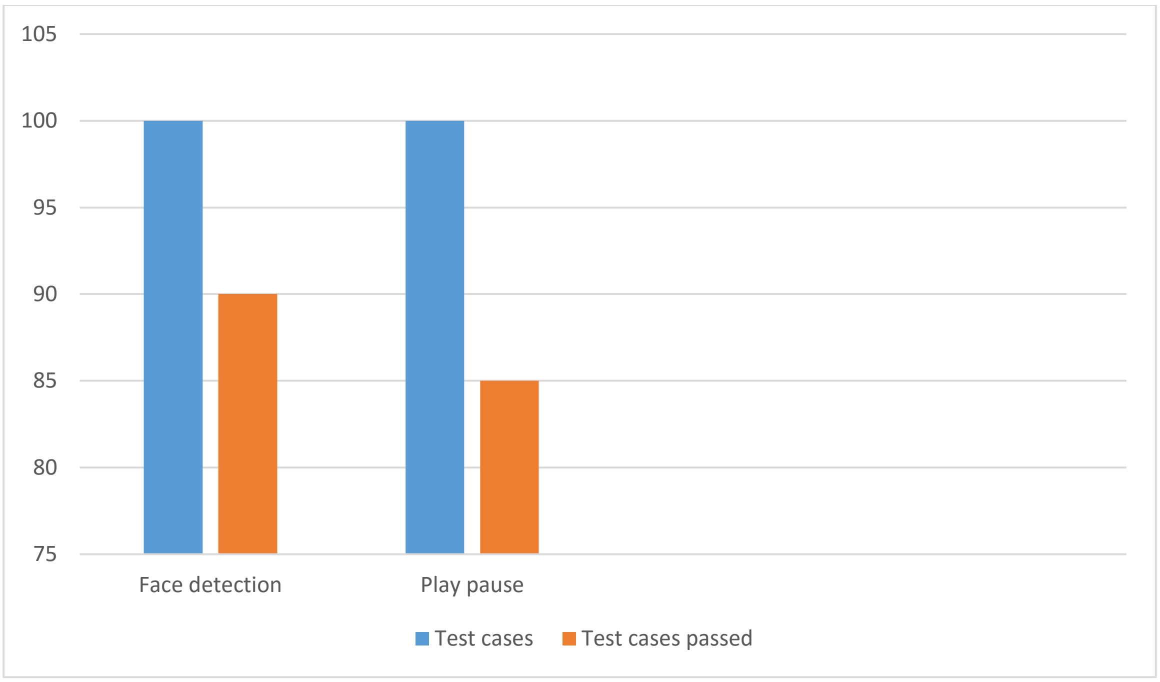
No.	Test case Title	Description	Expected outcome
1.	Start Media player and opening the Web Camera.	Choosing the video from Media Player and Opening web camera at the same time.	Two windows should open. One is showing video and another is Web Camera.
2.	Capturing the frames from Web Camera.	Capturing live frames one by one from web camera.	Frame should be successfully captured.
3.	Detecting the Face using Haar Casecade algorithm.	Detect the face and pass it to the tracking function.	Face should be detected successfully.
4.	Tracking the detected face.	Keep on tracking and keep playing the video.	After detection keep playing video.
5.	Absence of face while tracking	Absence of face while tracking then pause the video and back to face detection module.	Video should pause in absence of face.

7. PERFORMANCE ANALYSIS

The performance analysis of the project is based on:

- Resolution of the camera.
- Number of frames sent per second.
- Quality of video.
- Angle of view of the camera.

Performance is increases as a frames are stores while video in process and after detecting correct face.

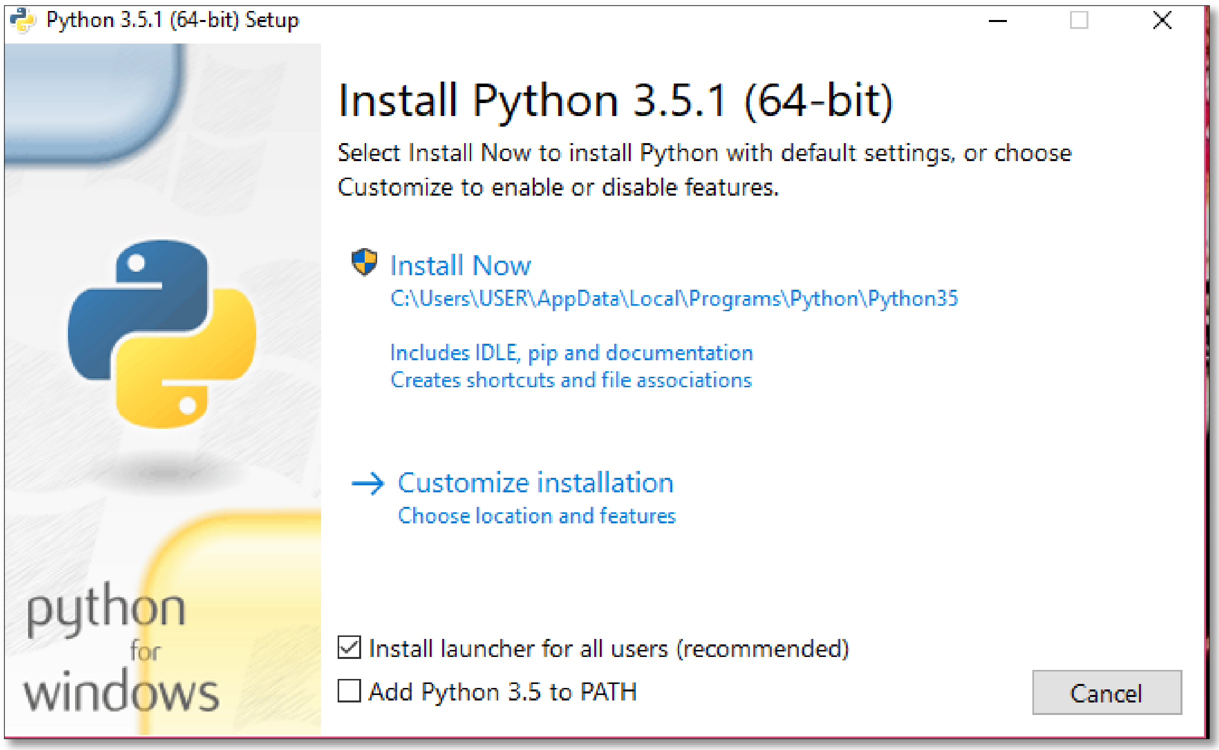


8. APLICATIONS

- This type of Media player used in Laptop’s.
- It is used to increase the level of human computer interaction.

9. INSTALLATION GUIDE AND USER MANUAL

Install Python:



Python 3.5.1 (64-bit) Setup

Install Python 3.5.1 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

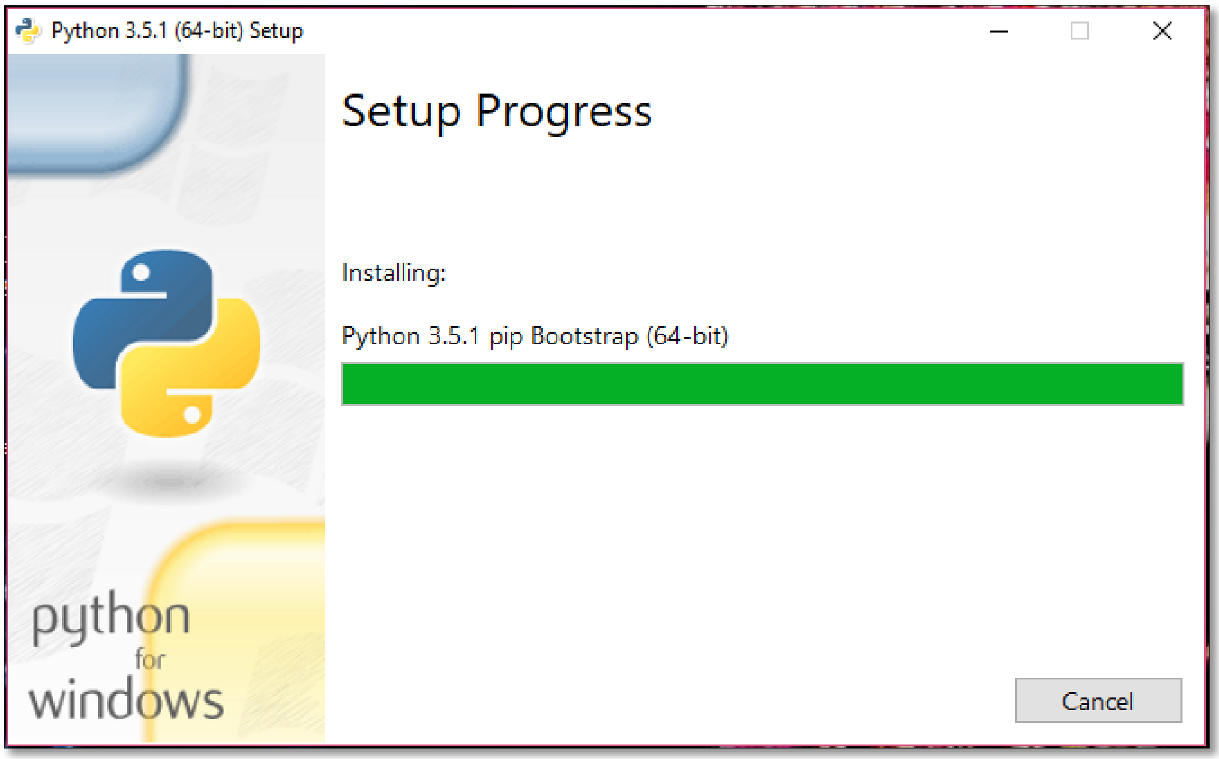
Install Now
C:\Users\USER\AppData\Local\Programs\Python\Python35

Includes IDLE, pip and documentation
Creates shortcuts and file associations

Customize installation
Choose location and features

☑ Install launcher for all users (recommended)
☐ Add Python 3.5 to PATH

Click on Install Now.

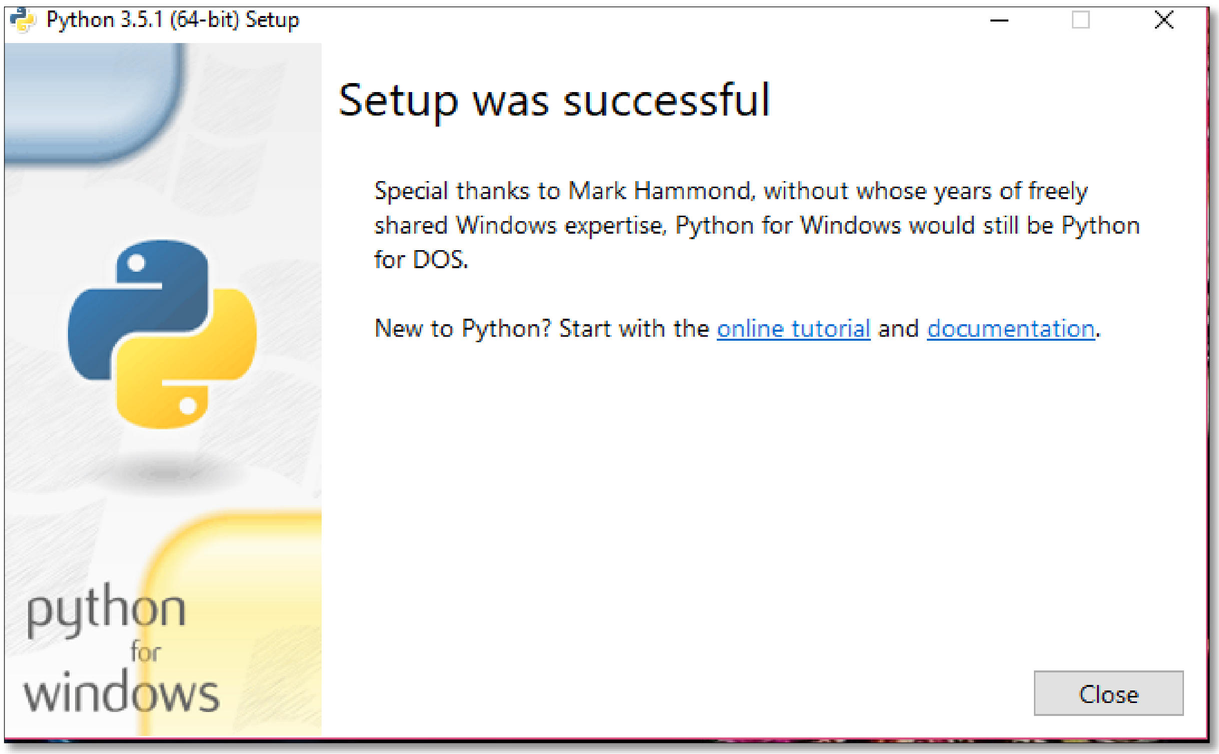


Python 3.5.1 (64-bit) Setup

Setup Progress

Installing:

Python 3.5.1 pip Bootstrap (64-bit)



Setup was successful

Special thanks to Mark Hammond, without whose years of freely shared Windows expertise, Python for Windows would still be Python for DOS.

New to Python? Start with the online tutorial and documentation.

If the setup run successfully, you should see a message "**Setup was successful message.**" Close and continue with the next steps.

Install OpenCV:

1. First of all open windows command prompt.
2. Now we will install OpenCV package by using "pip install opencv -python".

10. ETHICS

As a Computer Science and Engineering student, I believe it is unethical to,

1. Surf the internet for personal interest and non-class related purposes during classes.
2. Make a copy of software for personal or commercial use.
3. Make a copy of software to friends.
4. Download pirated software from the Internet.
5. Distributed pirated software from the Internet.
6. Buy a software with a single user license and install it on multiple computer.
7. Share a pirated copy of software.
8. Install a pirated copy of software.

11.REFERENCES

[1] PetcharatPattanasethanonand CharuaySavithi, “Human Face Detection and Recognition using Web-Cam”, Journal of Computer Science ISSN 2012.
[2] Prof S.V. Viraktamath, MukundKatti, AdityaKhatawkar&PavanKulkarni, “Face Detection and Tracking using OpenCV.”, Transactions on Computer Networks & Communication Engineering, Vol. 1, No. 3,3 July-August 2013.
[3] Swapna Agarwal and SaiyedUmer, “Media Player controlled by Facial Expressions and Gestures”, IEEE Conference 201--5.