**A PRELIMINARY REPORT ON**

**“look-Based media player”**

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY,PUNE

IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS

OF

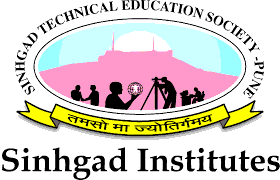
**THIRD YEAR COMPUTER ENGINEERING**

**SUBMITTED BY**

**1]HARSH BHOYAR 125**

**2]SHIVAM JADE 175**

**3]ROSHNI GURUBAXANI 124**



**DEPARTENT OF COMPUTER ENGINEERING**

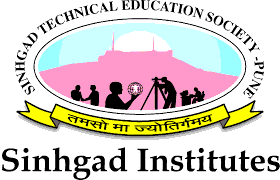
**STES’S**

**SMT.KASHIBAI NAVALE COLLEGE OF ENGINEERING**

**VADGAON BK, OFF SINHGAD ROAD,PUNE 411041**

**SAVITRIBAI PHULE PUNE UNIVERSITY**

**2019-2020**



**CERTIFICATE**

This is to certify that the project report entitles

**“look-based media player”**

Submitted By

**1]HARSH BHOYAR 125**

**2]SHIVAM JADE 175**

**3]ROSHNI GURUBAXANI 124**

is a bonafide work carried out by them under the supervision of **Prof Swapnil.Patil** and it is approved for the partial fulfilment of the requirement of third year computer engineering.

**(Prof Swapnil.Patil) (Dr.P.N.Mahalle)**

Guide Head,

Department of Computer EngineeringDepartment

**(Dr.A.V.Deshpande)**

Principal,

Smt.Kashibai Navale College of Engineering Pune-41

Place:Pune Date:03/12/2019

**ACKNOWLEDGEMENT**

I would like to express my special thanksof gratitude to **Prof Swapnil.Patil**, Head of Department Prof. Dr.P.N.Mahalle, and principal Dr. A.V. Deshpande who gave **me** the golden opportunity to do this miniproject on the topic “**Look-Based media player”**, which also helped me in doing image processing and I came to know about so many new things and I am really thankful to them.

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals. We would like to extend our sincere thanks to all of them.

We are highly indebted to (Smt.Kashibai Navale College of Engineering ,Pune) for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

We would like to express our gratitude towards parents & member of (Smt.Kashibai Navale College of Engineering ,Pune) for their kind co-operation and encouragement which helped us in completion of this project.

We would like to express our special gratitude and thanks to subject teacher for giving us such attention and time.

Our thanks and appreciations also goes to colleague in developing the project and people who have willingly helped me out with their abilities.

**CONTENTS**

**CHAPTER TITLE PAGE NO.**

* **Abstract 5**
* **Problem Definition 6**
* **Introduction 7**
* **Project description 8**
* **Motivation 11**
* **Software Requirement Specification 12**
* **Architecture 13**
* **Algorithm 14**
* **Screenshots 16**
* **Reference 17**
* **Conclusion 18**

**ABSTRACT**

In this project we are developing an advanced media player which plays and pauses the video by detecting the users face looking at screen or not. System monitors whether the user is looking at the screen or not using a web camera. If yes then doesn't interrupts the video and allows it to play. In case if the user is not looking at the screen or say the system couldn't detect the users face then it immediately stops the video.

Keywords: Face, media player, face detection.

**PROBLEM DEFINITION**

The goal of our project is to create an advanced media player based on look. We have set the following objectives for our media player to achieve the goal:

1. The user interface of media player should efficient and user friendly.

2. The media player should be accurate in terms of result.

3. The media player pause the video as soon as the user face is not detected

4. We are trying to make this project with an aim for 10.

**INTRODUCTION**

Usually when you are watching a video and someone calls you, you have to look somewhere else or go away from pc for some time so you miss some part of the video. Later you need to drag back the video from where you saw it. Well here is a solution to this problem. A look based media player that pauses itself when user is not looking at it. The player starts running again as soon as the user looks at it again. This is done using the camera or web camera on top of the computer. As long as the camera detects the users face looking at it, the media is played

**Project description**

Object Detection is a computer technology related to computer vision, image processing and deep learning that deals with detecting instances of objects in images and videos. We will do object detection using something known as haar cascades.

**What are Haar Cascades?**  
Haar Cascade classifiers are an effective way for object detection. This method was proposed by Paul Viola and Michael Jones.

Haar Cascade is a machine learning-based approach where a lot of positive and negative images are used to train the classifier.

* **Positive images –** These images contain the images which we want our classifier to identify.
* **Negative Images –** Images of everything else, which do not contain the object we want to detect.

At this purpose, OpenCV offers us a whole set of ready use [Haar Feature-based cascade classifiers](https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_objdetect/py_face_detection/py_face_detection.html) trained to detect faces. In the code the CascadeClassifier() function is used to load the classifier.

First of all make sure you have OpenCV installed. You can install it using pip:

pip install opencv-python

Now, things to note while detecting the faces in images:

* The detection works only on grayscale images. So it is

important to convert the color image to grayscale.

* For color conversion, we use the function cv2.cvtColor(input\_image, flag) where flag determines the type of conversion.
* For BGR  Gray conversion we use the flags cv2.COLOR\_BGR2GRAY
* **detectMultiScale**function is used to detect the faces. It takes 3 arguments— the input image, *scaleFactor* and *minNeighbours*. *scaleFactor* specifies how much the image size is reduced with each scale. *minNeighbours*specifies how many neighbors each candidate rectangle should have to retain it..
* *faces* contains a list of coordinates for the rectangular regions where faces were found. We use these coordinates to draw the rectangles in our image.

Similarly, we can detect faces in videos. As you know videos are basically made up of frames, which are still images. So we perform the face detection for each frame in a video.

The only difference here is that we use an infinite loop to loop through each frame in the video. We use *cap.read()*to read each frame. The first value returned is a flag that indicates if the frame was read correctly or not. We don’t need it. The second value returned is the still frame on which we will be performing the detection.

**Motivation**

The motivation behind the project is that

* This enhanced media player can help in minimizing human efforts. In future, this technique can be used to control systems using HCI like pdf reader, power point etc.
* To learn different modules like Opencv, PyQt5,easygui.
* The video stops as user changes their view from the video thereby no need of users to keep on dragging back to the point from where they missed.
* It saves time and electricity.
* It gives accurate result.

**Software AND Requirements**

**Softwares:**

* Python 3.9.0(64-bit)
* K-Lite\_Codec\_Pack\_1587\_Basic.exe

**Requirements:**

* OpenCV 3.2.0 should be installed
* Web-cam
* PyQt5
* PyQt5-stubs

**ARCHITECTURE**

In this project we are using face recognition for controlling media player. Face recognition is used for pausing and playing

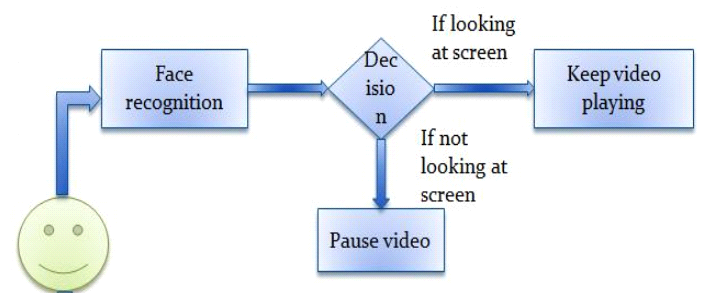


Fig -1: Basic Block diagram of system.

**ALGORITHM**

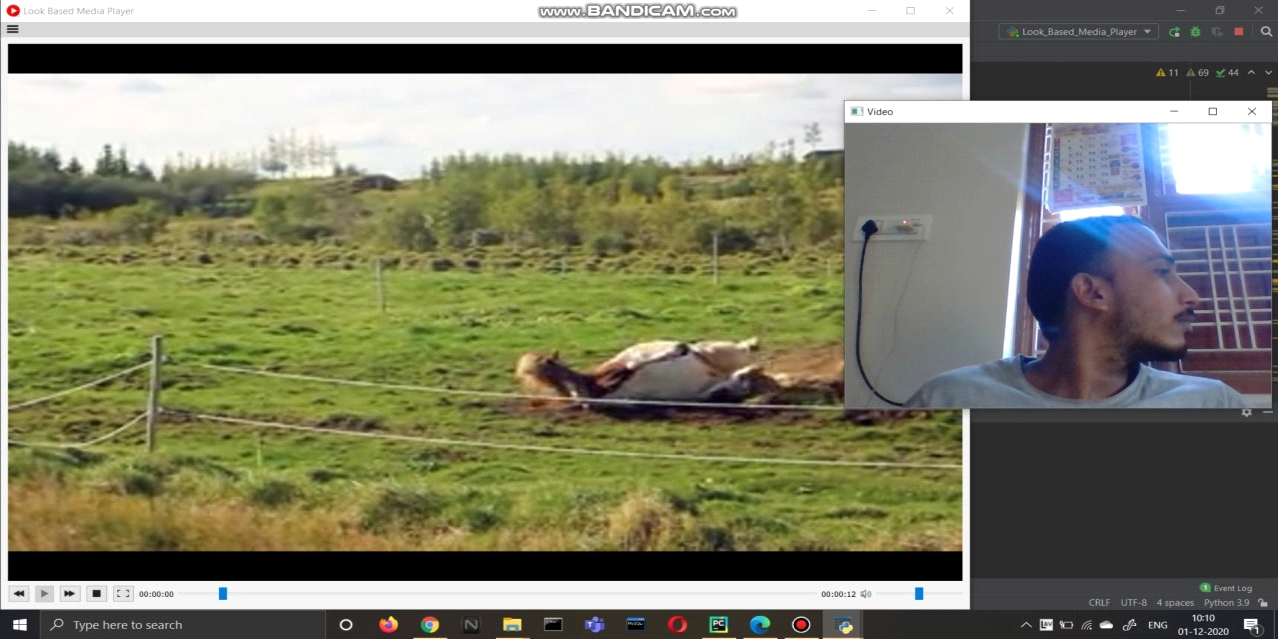
In the Viola–Jones object detection framework, the Haar-like features are therefore organized in something called a classifier cascade to form a strong learner or classifier. The key advantage of a Haar-like feature over most other features is its calculation speed.Haar-like features are digital image features used in object recognition. They owe their name to their intuitive similarity with Haar wavelets and were used in first real-time face detector.

In the detection phase of the Viola–Jones object detection framework, a window of the target size is moved over the input image, and for each subsection of the image the Haarlike feature is calculated. This difference is then compared to a learned threshold that separates non-objects from objects. Because such a Haar-like feature is only a weak learner or classifier (its detection quality is slightly better than random guessing) a large number of Haar-like features are necessary to describe an object with sufficient accuracy. In the Viola– Jones object detection framework, the Haar-like features are therefore organized in something called a classifier cascade to form a strong learner or classifier. The key advantage of a Haar-like feature over most other features is its calculation speed. Due to the use of integral images, a Haar-like feature of any size can be calculated in constant time (approximately 60 microprocessor instructions for a 2-rectangle feature).

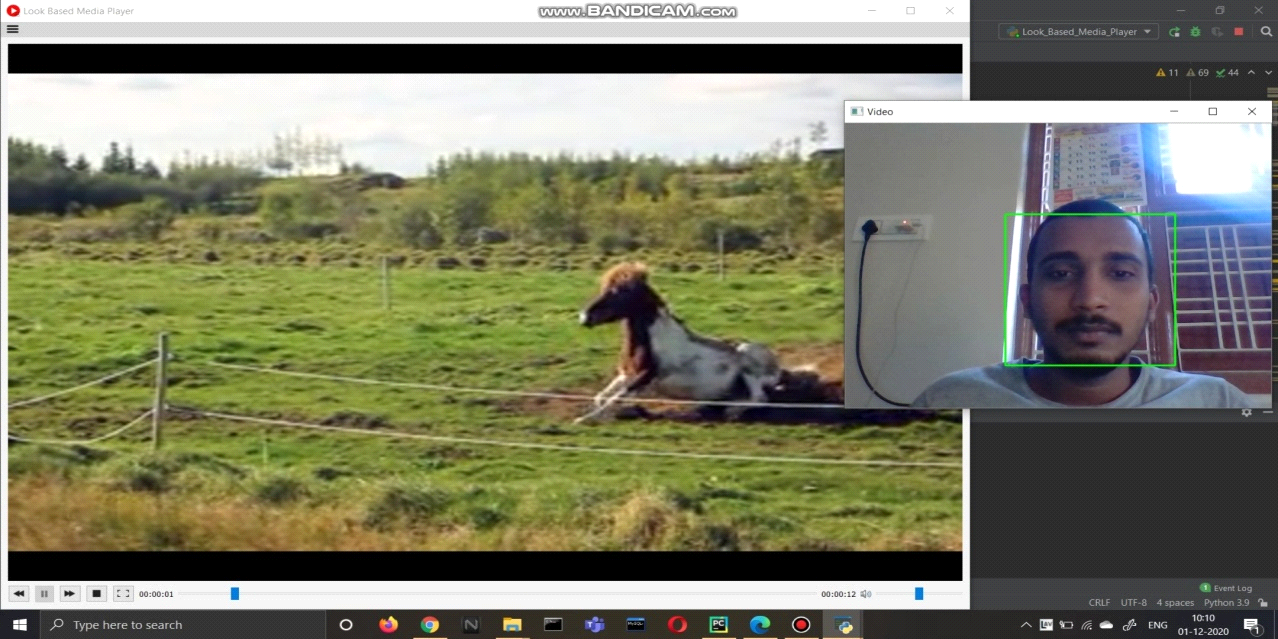
Open CV's algorithm is currently using the following Haarlike features which are the input to the basic classifiers:

* Feature = w1 x RecSum(r1) + w2 x RecSum(r2)
* Weights can be positive or negative.
* Weights are directly proportional to the area.
* Calculated at every point and scale

**Screenshots**



Note – The video is paused.



Note – The video got resumed

**References**

* [**https://www.geeksforgeeks.org/python-haar-cascades-for-object-detection**](https://www.geeksforgeeks.org/python-haar-cascades-for-object-detection)
* [**https://www.geeksforgeeks.org/introduction-to-opencv/**](https://www.geeksforgeeks.org/introduction-to-opencv/)
* [**https://towardsdatascience.com/face-detection-in-2-minutes-using-opencv-python**](https://towardsdatascience.com/face-detection-in-2-minutes-using-opencv-python)**-**

**CONCLUSION**

The main concern of this project is to help the user get best experience of using a media player. We have tried to achieve this goal by automating the media player in a wide extent. We are doing this by using face recognition for controlling varied features of the media player such as pausing and starting the video again when the user isn’t looking at the screen(for which face recognition is used).