FACIAL AND HAND GESTURE BASED MEDIA PLAYER

A project report submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY-H, HYDERABAD

In partial fulfillment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

Submitted by
IV B.Tech II Semester

P. THIRUPATHI	**(16UC1A0539)**
B. VANI	**(16UC1A0514)**
D. VAISHNAVI	**(16UC1A0520)**
S. SONY	**(16UC1A0547)**

Under the Esteemed Guidance of
Dr. A. Kiran Mayee, M.Tech, Ph.D.
Prof & Principal

Department of Computer Science and Engineering

TALLA PADMAVATHI COLLEGE OF ENGINEERING

(Affiliated to JNTUH, Hyderabad and approved by AICTE, New Delhi)
Somidi, Kazipet, Warangal Urban-506003.

(2016-2020)

TALLA PADMAVATHI COLLEGE OF ENGINEERING

(Affiliated to JNTUH, Hyderabad and approved by AICTE, New Delhi)
Somidi, Kazipet, Warangal Urban, Telangana-506003.

Department of Computer Science and Engineering

Certificate

This is to certify that the project work entitled

“FACIAL AND HAND GESTURE BASED MEDIA PLAYER”

is the bonfide work done by

P. THIRUPATHI	(16UC1A0539)
B. VANI	(16UC1A0514)
D. VAISHNAVI	(16UC1A0520)
S. SONY	(16UC1A0547)

In the Department of Computer Science and Engineering, Talla Padmavathi College of Engineering, Somidi, affiliated to JNTU-Hyderabad in partialfulfillment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering during 2016-2020.

This work has been carried out under my guidance and supervision.

The results embodied in this Project report have not been submitted in any University or Organization for the award of any degree or diploma.

INTERNAL GUIDE:

DR. A. KIRAN MAYEE M.Tech, Ph.D.
PROFESSOR & PRINCIPAL
DEPT OF CSE
T.P.C.E
SOMIDI, WARANGAL URBAN

HEAD OF DEPARTMENT:

ASST. PROF. K. DEEPIKA
ASSOCIATE PROFESSOR & HEAD
DEPT OF CSE
T.P.C.E
SOMIDI, WARANGAL URBAN

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

Before getting into the thickest of things, we would like to thank the personalities who were part of my project in numerous ways, those who gave me outstanding support from birth of the project.

We are extremely thankful to our beloved chairman and establisher **Talla Mallesham** & Managing Directors **Mr. Talla Vamshi** and **Mrs. Talla Chaithanya** forproviding necessary infrastructure and resources for the accomplishment of our project at **Talla Padmavathi College of Engineering, Warangal Urban**.

We are highly indebted to **Prof. Dr. A. Kiran Mayee, Principal of Talla Padmavathi College of Engineering**, for her support during thetenure of the project.

We are very much obliged to our beloved **Asst. Prof. K. Deepika,Head of the Department of Computer Science & Engineering**, Talla Padmavathi College of Engineering for providing the opportunity to undertake this project and encouragement in completion of this project.

We hereby wish to express our deep sense of gratitude to **Dr.A.Kiran Mayee, Principal**, Department of ComputerScience and Engineering, Talla Padmavathi Engineering College for the esteemed guidance, moral support and invaluable advice provided by her for the success of the project.

We are also thankful to all the staff members of Computer Science and Engineering department who have co operated in making our project a success. We would like to thank all our parents and friends who extended their help, encouragement and moral support either directly or indirectly in our project work.

Thanks for Your Valuable Guidance and kind support.

DECLARATION

We hereby declare that project report entitled **“FACIAL AND HAND GESTURE BASED MEDIA PLAYER”** is a genuine project work carried out by us,in **B.Tech (Computer Science and Engineering)** degree course of **JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD**
and has not been submitted to any other courses or University for award of any degree by us.

Signature of the Student

1.
2.
3.
4.

ABSTRACT

Advanced technologies are the key factor of today’s upcoming world.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 or say the system couldn’t detect the users face then it immediately stops the video.

We have implemented features for controlling other features of media player such as volume up and volume down, backward and forward video using hand gestures. Hand gestures plays a crucial role in increasing and reducing the volume. Look based media player also work as the field of computer vision based on hand gesture interfaces for Human Computer Interaction (HCI).

Currently, we have built an application which gives accurate result in terms of performance. This media player also helps the users to have better experience in daily life. It also helps with the user friendly performance.

CONTENTS

List of Figures

List of Tables

1. INTRODUCTION

1. INTRODUCTION

While watching a video when someone interrupts you and you have to look somewhere else or go away from the system for some time so you need to drag back the video from where you left. Well we got a solution to this problem. A media player that gets paused when user is not looking at it. The media player resumes again as soon as the user looks at it again. This can be done using the web camera. The media player will be played continuously as long as the camera detects the users face.

The media player pauses as soon as users face is not completely detected. This system also provides the feature of controlling other functions of media players such as play, pause, volume up, volume down, next using hand gestures helps to change to a new video.

Direct use of hands as an input device is an attractive method for improving natural Human Computer Interaction which has evolved from text based interfaces from 2D graphical based interfaces, multimedia supported interfaces, to fully fledged multi-participant Virtual Environment systems. This enhanced media player can help in minimizing human efforts. In future, this technique can Get better experience of using media player by not missing any part of video etc.

This task of hand gesture recognition is one of the important and elemental problem in computer vision. With recent advances in information technology and media, automated human interactions systems are build which involve hand processing task like hand detection, hand recognition and hand tracking.

This prompted my interest so we planned to make a software system that could recognize human gestures through computer vision, which is a sub field of artificial intelligence. The purpose of my software through computer vision was to program a computer to “understand” a scene or features in an image.

1.2 STATEMENT OF THE PROBLEM

“Face and Hand Gesture Recognition Using Camera” is based on concept of Image processing. In recent year there is lot of research on gesture recognition using Kinect sensor on using HD camera but camera and Kinect sensors are more costly. This paper is focus on reduce cost and improve robustness of the proposed system using simple web camera.

Gesture recognition has been adapted for various research applications from facial gestures to complete bodily human action. Several applications have emerged and created a stronger need for this type of recognition system. Static gesture recognition is a pattern recognition problem; as such, an essential part of the pattern recognition pre-processing stage, namely, feature extraction, should be conducted before any standard pattern recognition techniques can be applied. Features correspond to the most discriminative information about the image under certain lighting conditions.

1.3 OBJECTIVES

First objective of this project is to create a complete system to detect, recognize and interpret the hand gestures and face through computer vision to pause the video.

Second objective of the project is therefore to propose a new low-cost, high speed and color image acquisition system.

1.4 SCOPE

This enhanced media player can help in minimizing human efforts. In future, this technique can be used to control system using HCI like pdf reader, power point etc. we are doing this by using face detection and hand gestures for controlling varied features of the media player.

1.5 PURPOSE

The aim of the project is to monitor and control the media player. This media player can increase the level of human computer interaction (HCI) than the current scenario.

1.6 LIMITATIONS

- System may not work in less light up work space.
- For face detection complete face must be visible

2. LITERATURE SURVEY

2. LITERATURE SURVEY

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, then next steps are to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration are taken into account for developing the proposed system.

2.1 Information Retrieval:

Many researchers have proposed numerous methods for hand gesture recognition systems. Generally, such systems are divided into two basis approaches namely glove − based and vision − based approaches. In glove based analysis, detection of the hand is eliminated by the sensors on the hand and 3D model of the hand is easily subjected to the virtual world and analysis comes next. Such systems are optimal for body motion capture purposes and widely used in industry. On the other hand vision-based analysis are more natural and useful for real time applications.

A healthy human can easily identify a hand gesture, however fora computer to recognize hand gesture first the hand should be detected in the acquired image and recognition of that hand should be done in a similar way as humans do. Yet this is a more challenging approach to implement because of the limitations of such a natural system. The vision-based approaches are carried out by using one or more cameras to capture and analyze 2D or 3D shapes of hands.

Detection and gesture analysis of the hands is a growing literature topic and has many user environment limitations for most of the studies. Segmentation of the hand is the first step of such systems. Exceptionally, such systems in [10], [12] and [13] gloves or finger marks have been used to extract the hand posture information in the frame and ease the hand segmentation process by eliminating the varying skin color issue of the problem. This technique allows the system to detect hands in a straightforward manner and it is more robust to change in lightning conditions and it is also independent of the user's skin color.

2.2 Detection Methods:

In recent years, with the introduction of a new approach [33], which has a high detection rate,new studies are mostly concentrated on Boosting and HMM. The most tempting side of using those methods is that they usually work with gray scale images instead of colored images and thus it eliminates the drawbacks of such color based noise issues. This innovative approach is using a well-known technique namely Adaboost classifier which was mentioned in [32]. Adaboost classifier is an effective tool to select appropriate features for face detection. This feature extraction technique does not need skin color information and have less computation time with the use of integral image concept. But the drawback of this method is that it re-quires a training process.This process often needs huge-sized sample images to have a high detection rate.

By considering the drawbacks of training based methods and since there is a huge amount of studies based on training in literature, a feature invariant method was chosen to locate faces in this study. Also the main objective of this study is recognizing hands and detection of face using an intermediate tool.

Approach	**Instance Study**
Feature Invariant Methods	
● Facial Features	Facial Components Analysis [21]
● Texture	Gray Scale Classification [23]
● Skin Color	Adaptive Gaussian Mixture Model [16]
● Multiple Features	Skin color, shape analysis and facial components
Template Matching Methods	
● Predefined Templates	Shape Template Matching [35]
● Deformable Templates	Skin Active Shape Model for Face Alignment. [20]
Appearance-Based Methods	
● Principal Components Analysis	Event Detection by Eigenvector Decomposition Eigenface Decomposition for Face Recognition[34]
● Neural Network	Motion Pattern Classification by Neural Networks
● Support Vector Machine (SVM)	SVM with Fisher Kernels [12]
● Bayes Classifier	Dynamic BN for Gesture Recognition [18]
● Hidden Markov Model (HMM)	Input-Output HMM for Gesture Recognition [19]
● Boosting and Ensemble	Detection using Boosted Features.[22]

an intermediate tool. Investigation of a new technique for face detection would be tempting and Multiple Features in table 2.1 was a good choice for detection of face. Skin color is the central invariant feature of this study. It is indicated that human skin color is independent of human race and the wavelength of the exposed light [36]. This fact is also valid for the transformed color spaces of common video formats and thus skin color can be defined as a global skin color cloud in the color space and this cloud is called skin locus of that color space[3]. The thresholds of the skin locus is too large to extract current skin color in an input image correctly. Since shadows, illumination and pigmentation of human skin conditions would vary in a wide range, it is reasonable to adapt this general thresholds and narrow the skin locus for the current conditions. Skin locus is supposed to include all the skin pixels in an image with some other skin like pixels. Those false positive pixels should be eliminated by a fine skin segmentation.

According to face detection method introduced in [2], colored images are investigated in2 steps namely coarse skin color segmentation and fine skin color segmentation. For coarse skin color segmentation fixed skin color thresholds in nRGB color space are used (skin locus).

2.3 Recognition Methods:

Many studies in literature use different skin locus thresholds for different color spaces to locate faces or hands in images. [25] starts with an RGB image and by apply a dimension reduction algorithm to propose its own skin locus in two dimension to compare its performance with HSV skin locus. [24] and [29] compare HSV/HSI,RGB,TSL and YcbCr color space skin locus performances. [26] starts the segmentation of skin color in YUV color space to have a quick result and then tune the current skin color with a quasi-automatic method which needs some user input. In [27], chrominance along with luminance information in YCbCr color space was used to segment the skin color for current conditions and histogram clustering is used for fine skin segmentation.

In feature invariant face detection methods skin segmentation is followed by face verification. Once the skin color is segmented in the obtained image, skin pixels are grouped by a region growing algorithm to extract blobs. Those blobs are investigated if they are face or not. Many methods have been proposed for this purpose in literature. The most simple one is that calculating height to width ratios of the segmented blobs [31]. Typically a frontal face's height to width ratio would be in a certain interval and this information would yield elimination of some blobs which are not definitely corresponding to real face in the image. Height to width ratio alone is not sufficient to detect the face blob because skin like pixels might still construct similar height to width ratio blobs. in [28] symmetry analysis and facial components analysis are introduced to detect faces. Detecting symmetric eyes, lips and nose would be the best method to recognize detect a face however it contributes an important computation time to the algorithm. [2] propose a method between the simplicity and the computational complexity of these two methods and presents blob's mismatch area method.

A clear summary of such algorithms are shown in table

Reference	Primary Method of Recognition	Number of Gestures Recognized	Background to Gesture Images	Additional Markers Required	Number of Training Images
39	Hidden Markov Models	97	General	Multi-colored gloves	400
5	Entropy Analysis	6	No	No	400
41	Linear ap-proximation to non-linear point distribution models	26	Blue Screen	No	7441
42	Finite State machine modeling	7	Static	Markers on glove	10 sequences of 200 frames each
43	Fast Template Matching	46	Static	Wrist band	100 examples per gesture

In literature, if a gesture recognition system is constructed on a training based methods, then the number of gestures that will be recognized could be increased. However if it is an invariant method or state based method the number of gestures could not be increased easily. However,the system would be faster and independent of the training process.

3. ANALYSIS

3. ANALYSIS

3.1 Existing System

Mostly existing systems use eye recognition. Due to which results aren't accurate. Face recognition and hand gestures are not implemented properly together and not even individually.

3.2 Disadvantages

There is a problem with the existing system. First, it involves a high computational cost, since we would have to repeat a large number of gestures similarity computations for every gesture.

3.3 Proposed System:

In this project we are using face recognition and hand gestures for controlling media player. Face recognition is used for pausing and playing. Various hand gestures are used for controlling other functions of media player.

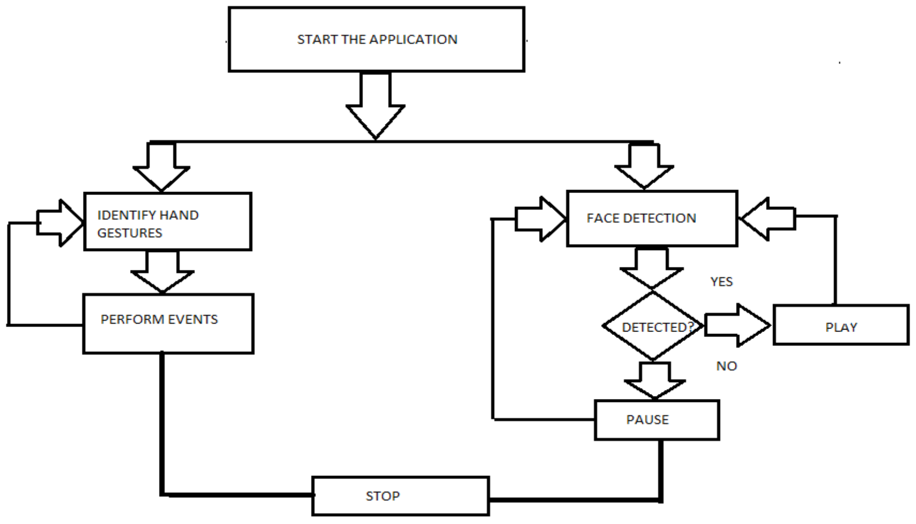


FIG: PROPOSED SYSTEM

3.4 Advantages

Facial and hand gesture based media player following advantages:

- Users cannot miss any part of the video.
- 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.
- You can also forward and backward the video if required.
- It saves time and electricity.
- It gives accurate result

3.5 System Used:

The following are the requirements we used to run proposed system.

Hardware System Configuration:

ANY CONTEMPORARY PC

Software System Configuration:

Operating System	➔	UBUNTU / UNIX
Scripts	➔	Python
Dependencies	➔	Imutils, numpy, opencv-python, vlc-ctrl

4. DESIGN

4.1 UML DIAGRAMS:

Class Diagram: .,OL

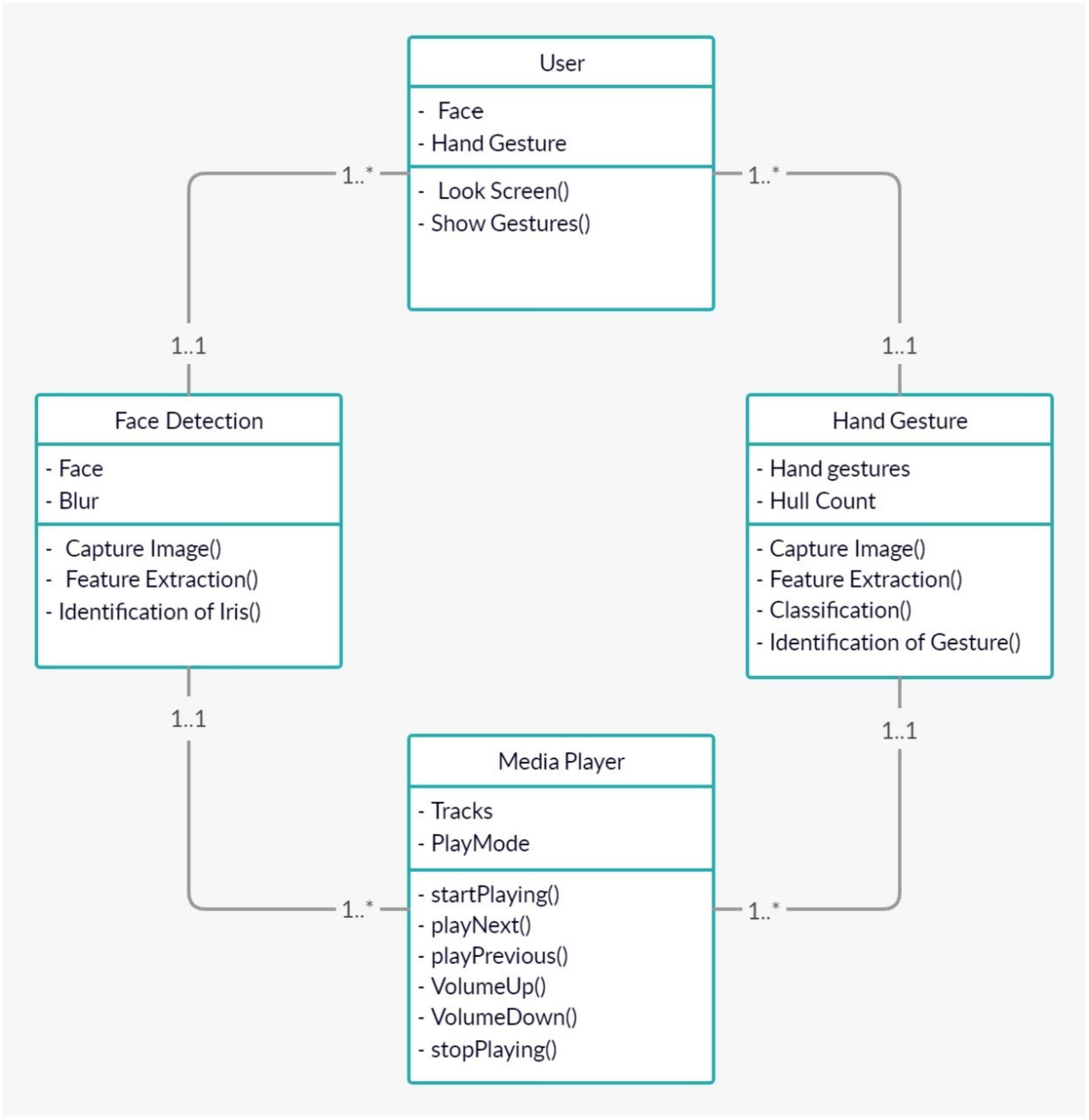


Fig.4.1. Class Diagram

Use Case Diagram:

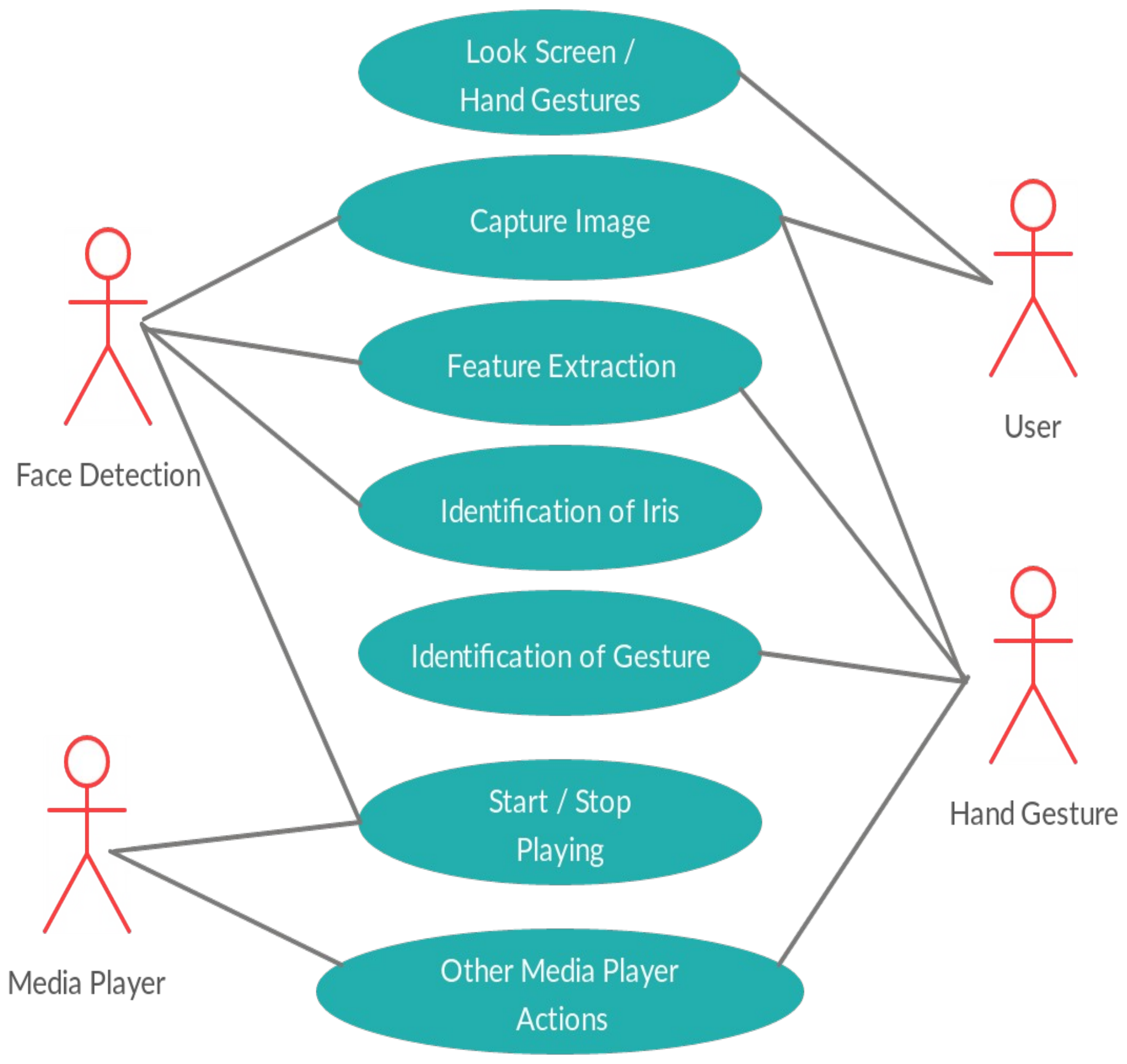


Fig.4.2. Use case Diagram

Data Flow Diagram:

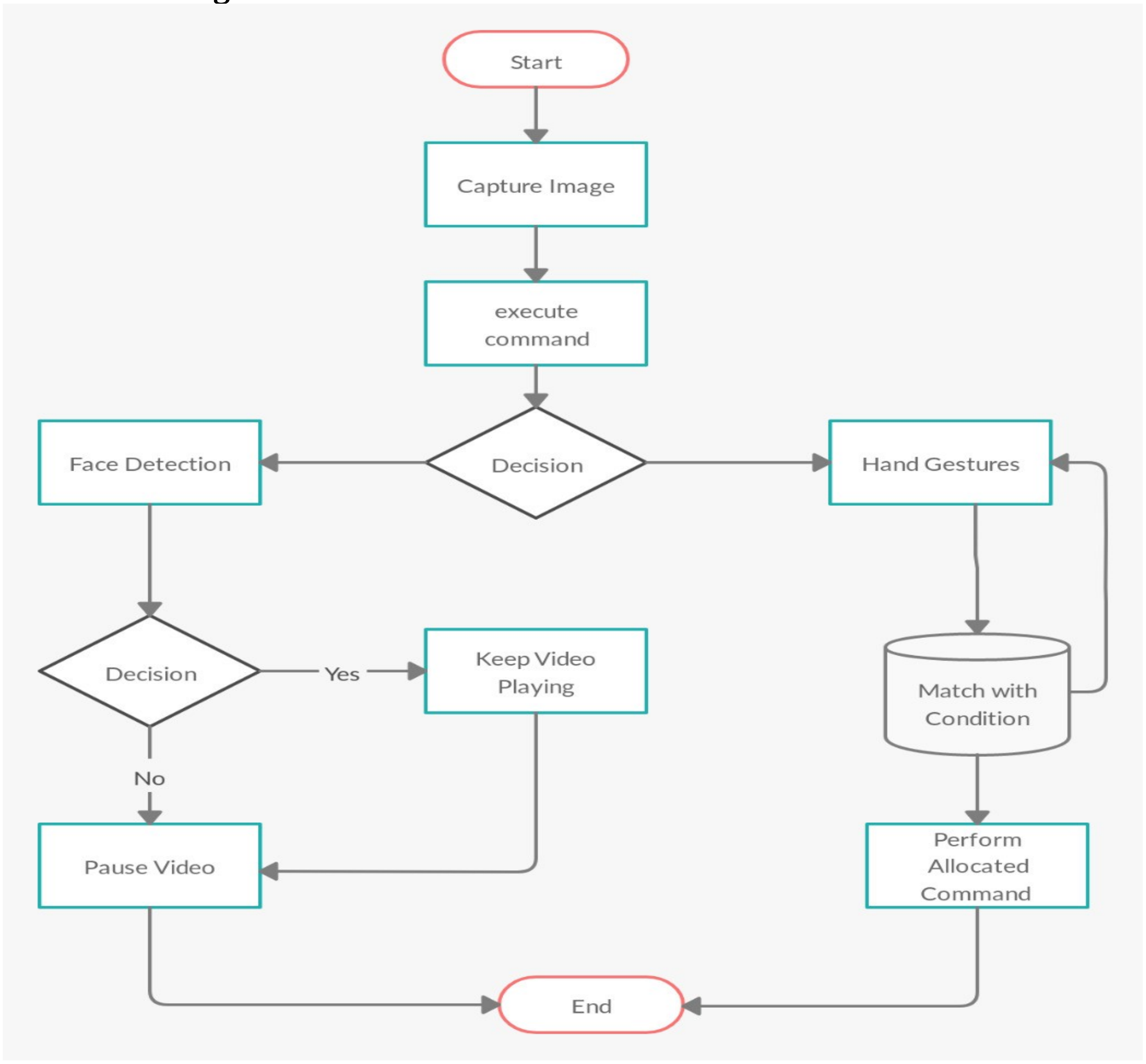


Fig.4.3 Data Flow Diagram

Sequence Diagram:

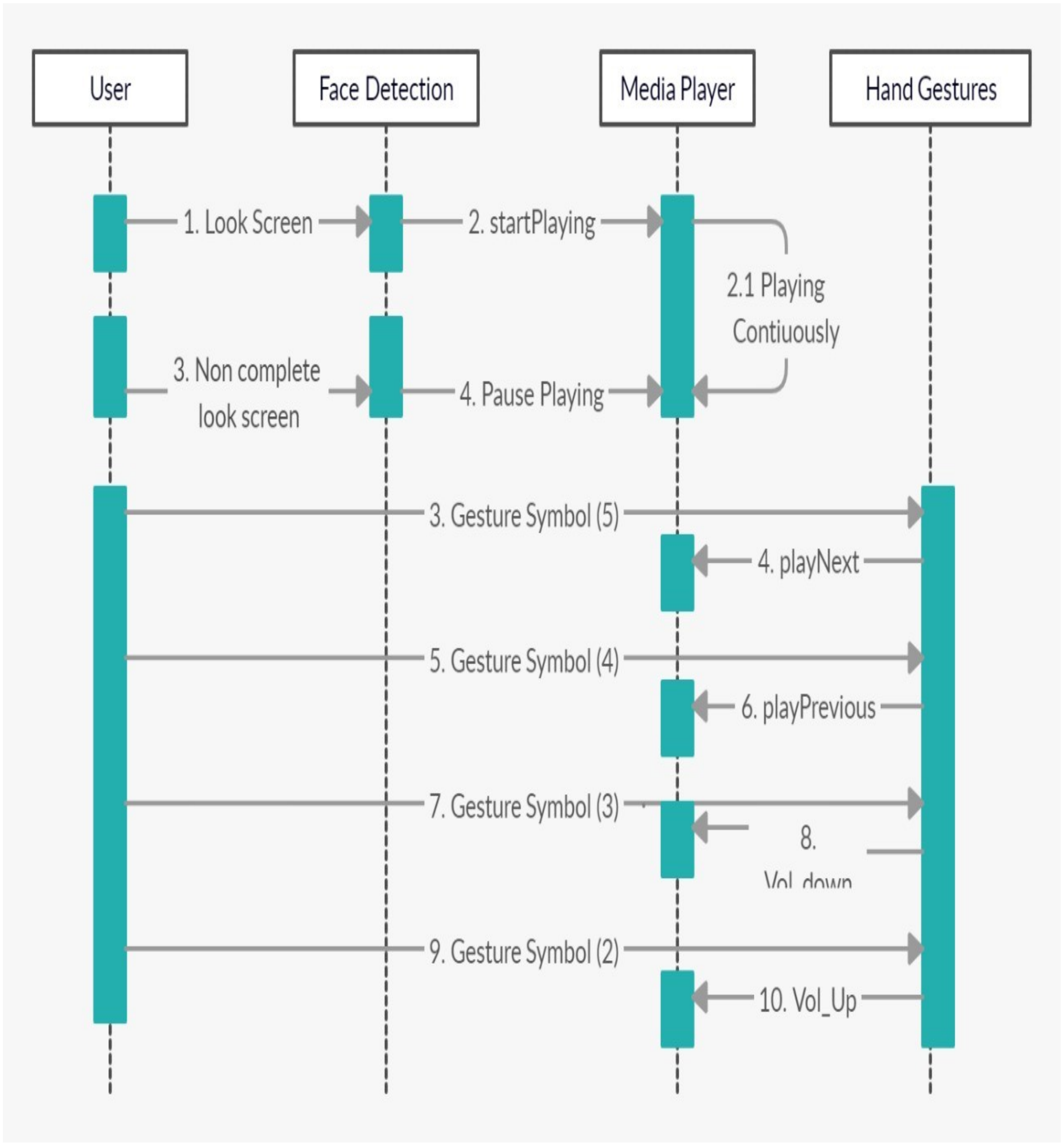
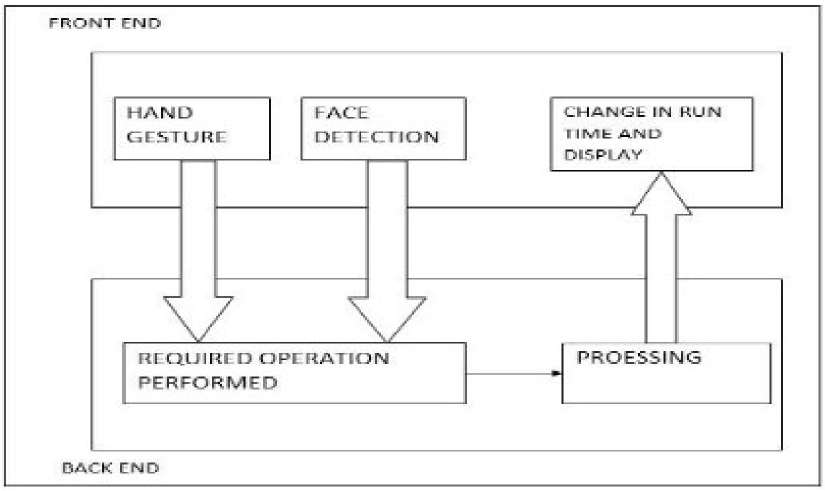


Fig.4.4. Sequence Diagram

5.CODING AND IMPLEMENTATION

5.1 IMPLEMENTATION

Here as soon as we start the media player the media player will be opened with face detection mode i.e. it will start detecting human face and if a face is detected the video will keep on playing if there is no face the video will be paused. This media player has an option to run media player with either face detection mode or hand gesture mode and it can switch between them in run time. In hand gesture mode we control the mouse cursor to operate the media player.



Module

1. Face Detection
2. Hand gesture
3. Media player

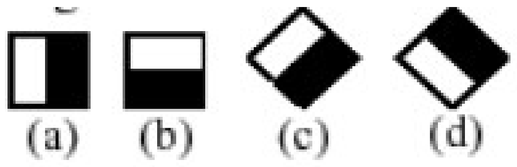
5.1.a Face Detection:

OpenCV's face detector uses a method that Paul Viola and Michael Jones had published in 2001. Usually this approach detects the objects in images which combines four key concepts:

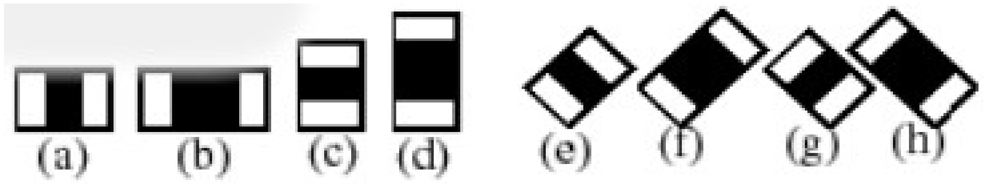
- Simple rectangular features which are called as Haar features.
- An integral Image for quick feature detection.
- A cascaded classifier to combine many features.

Following are the OpenCV's haar like features:-

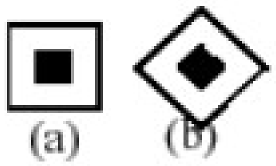
1. **Edge features**



2. **Line features**



3. **Centre-Surround features**



In this section the camera starts detecting a human face and till the time we are looking at the media player screen the video will be played and if we look somewhere else the video will be automatically paused. It will start playing the video as soon as it detects a face again.

5.1.b Hand gesture:

In this section the camera looks for the color bands. The user will wear 2 bands on his fingers they are green and blue color bands. Here the blue color band controls mouse cursor and green color band performs left click operation. Thus we can control the media player.

HSV (Hue, Saturation, Value) Color scheme:-

Used classica lmethod to detect skin pixels By setting Upper & Lower bound values

Hmin &lt;=H &lt;= Hmax { Hmin 0 Hma x 20......(1)

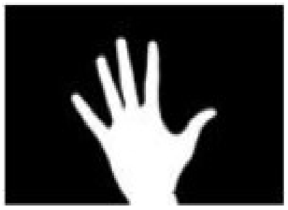
Smin &lt;=S &lt;= Smax { Smin 45 Sma x 255.....(2)

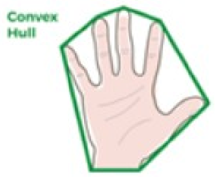
STEPS:-

1. **INPUT IMAGE**

2. **HSV**

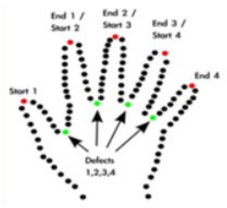
3. **BINARY IMAGE**





4. **CONVEX HULL**

5. **CONVEXITY DEFECTS**



5.1.c Media player:

The media player is the important concept which will be used to play video and it will be controlled by functionalities like hand gesture and face detection.

5.2 CODING

Main.py

```
import math
from tkinter import font as tkFont
import numpy as np
import cv2  # required 3+
import tkinter as tk
from threading import Thread
import queue as Queue
import time
import os

from commands import *


# Frontal face classifier is imported here
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

request_queue = Queue.Queue()
result_queue = Queue.Queue()
t = None
debug = False
enable_commands = False
REALLY_NOT_DEBUG = True
CHANGE_VOLUME = False
COOLDOWN = 5
LAST_TIME = time.time()


def submit_to_tkinter(cb, *args, **kwargs):
request_queue.put((cb, args, kwargs))
return result_queue.get()


def debug_toggle():
global debug
debug = not debug


def toggle_commands():
global enable_commands
enable_commands = not enable_commands
```

```
def main_tk_thread():
global t

def timertick():
try:
cb, args, kwargs = request_queue.get_nowait()
except Queue.Empty:
pass
else:  # if no exception was raised
retval = cb(*args, **kwargs)
result_queue.put(retval)
# reschedule after some time
t.after(10, timertick)

# create main Tk window
t = tk.Tk()
t.title("Debug controls")
t.geometry('%dx%d+%d+%d' % (320, 320, 850, 200))
# set font for labels
font = tkFont.Font(family="Arial", size=18, weight=tkFont.BOLD)
# create buttons, labels
tc = tk.Button(text='enable commands', name='ec', command=toggle_commands, width='15')
tc.place(x=20, y=210)
b = tk.Button(text='debug mode', name='dbg', command=debug_toggle, width='15')
b.place(x=20, y=260)
hull = tk.Label(t, name="hull", text="None", font=font)
hull.place(x=20, y=10)
defects = tk.Label(t, name="defects", text="None", font=font)
defects.place(x=20, y=60)
defects_filtered = tk.Label(t, name="defects_filtered", text="None", font=font)
defects_filtered.place(x=20, y=110)
command = tk.Label(t, name="command", text="None", font=font)
command.place(x=20, y=160)
en_command = tk.Label(t, name="en_command", text="None")
en_command.place(x=160, y=215)
en_dbg = tk.Label(t, name="en_dbg", text="None")
en_dbg.place(x=160, y=265)
# start timer a.k.a. scheduler
timertick()
# main Tk loop
t.mainloop()
```

```
# setters for Tk GUI elements
def hull_label(a):
t.children["hull"].configure(text=str("All hulls = %s " % a))


def defects_label(a):
t.children["defects"].configure(text=str("All defects = %s" % a))


def defects_filtered_label(a):
t.children["defects_filtered"].configure(text=str("Defects filtered = %s" % a))


def command_label(a):
t.children["command"].configure(text=str("Command = %s" % a))


def en_command_label(a):
t.children["en_command"].configure(text=str("(%s)" % a))

def en_dbg_label(a):
t.children["en_dbg"].configure(text=str("(%s)" % a))


def check_command(c, exe):
if c == 0:
return "NOACTION"
elif c == 1:
if REALLY_NOT_DEBUG and exe:
move_next()
return "NEXT"
elif c == 2:
if REALLY_NOT_DEBUG and exe:
move_prev()
return "PREVIOUS"
elif c == 3:
if CHANGE_VOLUME and exe:
vol_down()
return "VOLUME CONTROL DOWN"
elif c == 4:
if CHANGE_VOLUME and exe:
vol_up()
return "VOLUME CONTROL UP"
return None
```

```
if __name__ == '__main__':
t = Thread(target=main_tk_thread)
t.start()

# Flag is used to pause and play the video [ if flag is 1 then the video plays else it doesn't ]
Pauseflag = 0

cap = cv2.VideoCapture(0)

while cap.isOpened():
# 1. Constructing Region Of Interest (ROI)
# read frame from camera
_, img = cap.read()
# add rectangle for 'target scanning area' 300x300
cv2.rectangle(img, (350, 350), (50, 50), (0, 255, 0), 0)
# crop input image to 300x300
crop_img = img[50:350, 50:350]
# convert image to gray scale
gray = cv2.cvtColor(crop_img, cv2.COLOR_BGR2GRAY)
gray1 = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
#face identification
faces = face_cascade.detectMultiScale(gray1, 1.3, 5)
if debug:
cv2.imshow('Gray scale', gray)
# use Gaussian blur
blur = cv2.GaussianBlur(src=gray, ksize=(35, 35), sigmaX=0)
if debug:
cv2.imshow('Blurred', blur)
# apply threshold to get black and white (binary) image (ROI)
_, thresh1 = cv2.threshold(blur, 127, 255, cv2.THRESH_BINARY_INV +
cv2.THRESH_OTSU)
if debug:
cv2.imshow('Threshold', thresh1)

# 2. Analyze ROI
# find contours
contours, hierarchy = cv2.findContours(thresh1, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
# contours, hierarchy = cv2.findContours(thresh1.copy(), cv2.RETR_TREE,
cv2.CHAIN_APPROX_NONE)

# get max contour area
cnt = max(contours, key=lambda x: cv2.contourArea(x))
```

```
# convex hull
x, y, w, h = cv2.boundingRect(cnt)
cv2.rectangle(crop_img, (x, y), (x + w, y + h), (0, 0, 255), 0)
hull = cv2.convexHull(cnt)

# contours
drawing = np.zeros(crop_img.shape, np.uint8)
cv2.drawContours(drawing, [cnt], 0, (0, 255, 0), 0)
cv2.drawContours(drawing, [hull], 0, (0, 0, 255), 0)
hull = cv2.convexHull(cnt, returnPoints=False)

# we find convexity defects, which is the deepest point of deviation on the contour
defects = cv2.convexityDefects(cnt, hull)
count_defects = 0
cv2.drawContours(thresh1, contours, -1, (0, 255, 0), 3)
for i in range(defects.shape[0]):
s, e, f, d = defects[i, 0]
start = tuple(cnt[s][0])
end = tuple(cnt[e][0])
far = tuple(cnt[f][0])
# dist
a = math.sqrt((end[0] - start[0]) ** 2 + (end[1] - start[1]) ** 2)
b = math.sqrt((far[0] - start[0]) ** 2 + (far[1] - start[1]) ** 2)
c = math.sqrt((end[0] - far[0]) ** 2 + (end[1] - far[1]) ** 2)
angle = math.acos((b ** 2 + c ** 2 - a ** 2) / (2 * b * c)) * 57
if angle <= 90:
count_defects += 1
cv2.circle(crop_img, far, 3, [255, 0, 0], -1)
# dist = cv2.pointPolygonTest(cnt,far,True)
cv2.line(crop_img, start, end, [0, 255, 0], 2)
if debug:
cv2.circle(crop_img, far, 5, [0, 0, 255], -1)

# show analyzed input/output
if debug:
all_img = np.hstack((drawing, crop_img))
cv2.imshow('Contours vs image', all_img)

cv2.imshow('Input', img)

k = cv2.waitKey(50)
# got ESC key? if yes - exit!
if k == 27:
break
```

```
elif k == 99:  # for 'c' toggle command execution
print ('c input')
toggle_commands()
elif k == 100:  # for 'd' toggle debug mode
print ('d input')
debug_toggle()

# do not 'change' command to quickly and wait after last one
if time.time() - LAST_TIME > COOLDOWN and enable_commands:
exe = True
LAST_TIME = time.time()
else:
exe = False

# check what command to execute and run it
com = check_command(count_defects, exe)

delta = time.time() - LAST_TIME
to_next = COOLDOWN - delta
if to_next < 0:
to_next = 0

if REALLY_NOT_DEBUG and exe:
for (x, y, w, h) in faces:
                play()
                Pauseflag = 1 # Face is detected hence play the video continuesly
          if Pauseflag == 0:
                pause()
          Pauseflag = 0


# submit some data to GUI
submit_to_tkinter(hull_label, str(hull.shape[0]))
submit_to_tkinter(defects_label, str(defects.shape[0]))
submit_to_tkinter(defects_filtered_label, str(count_defects))
submit_to_tkinter(en_command_label, str("%s, cooldown %.2fs." % (enable_commands,
to_next)))
submit_to_tkinter(en_dbg_label, debug)
if com:
submit_to_tkinter(command_label, com)
```

Commands.py:

```
from subprocess import check_call

__all__ = ['play', 'pause', 'move_prev', 'move_next', 'vol_up', 'vol_down']


def play():
check_call(['vlc-ctrl', 'play'])


def pause():
check_call(['vlc-ctrl', 'pause'])


def move_next():
check_call(['vlc-ctrl', 'next'])


def move_prev():
check_call(['vlc-ctrl', 'prev'])


def vol_up():
check_call(['vlc-ctrl', 'volume', '+0.1'])


def vol_down():
check_call(['vlc-ctrl', 'volume', '-0.1'])
```

6. TESTING

6. TESTING

PURPOSE:

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.1 Types of Tests

Unit testing:

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Functional Testing:

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input: Identified classes of valid input must be accepted.

Invalid Input: Identified classes of invalid input must be rejected.

Functions: Identified functions must be exercised.

Output: Identified classes of application outputs must be exercised.

Systems/Procedures: Interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

6.2 Test Strategy and Approach

Field-testing will be performed manually and functional tests will be written in detail.

Test Objectives:

- All field entries must work properly.
- Media player must be running state.
- The entry screen, messages and responses must not be delayed.

Features to be tested:

- Verify that the entries are of the correct format.
- No duplicate entries should be allowed.
- All links should take the user to the correct page.

6.3 Test Results

All the test cases mentioned above passed successfully. No defects encountered.

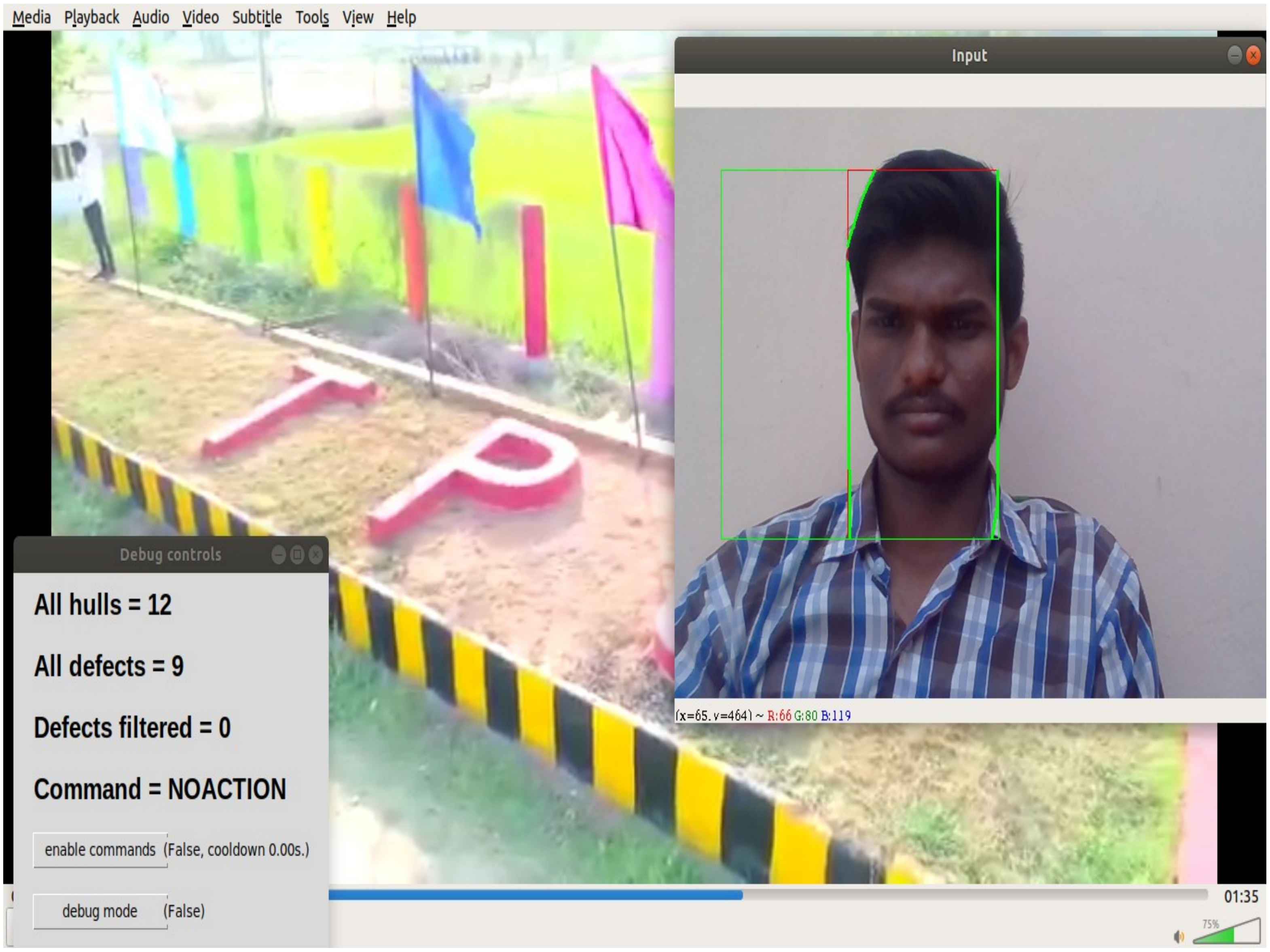
7. RESULTS AND OUTPUT SCREENSHOTS

7.1 RESULTS

Using the proposed system the following results can be observed.

Result of Media Player Control:

ubuntu@pc$python main.py

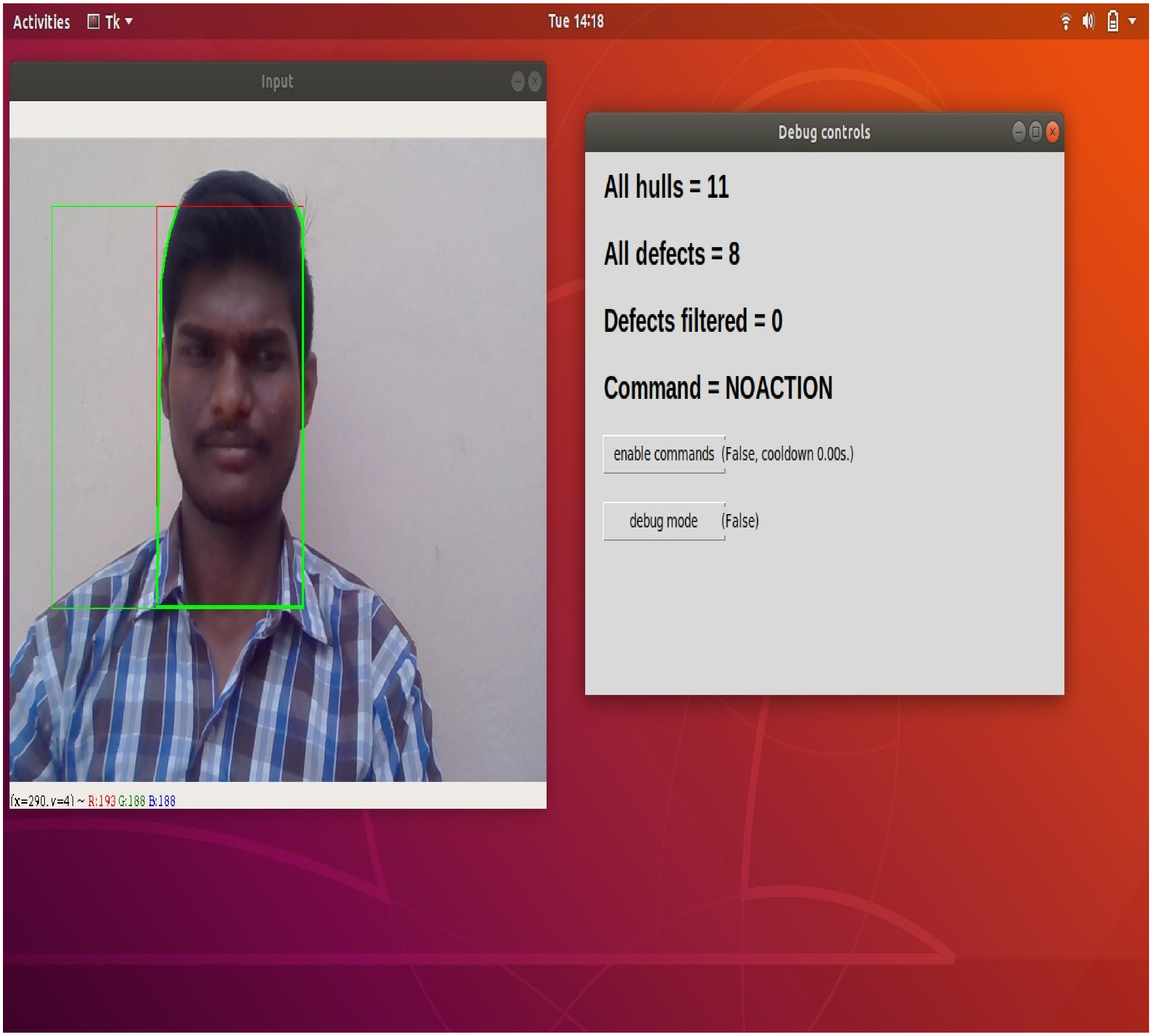


```
(x=65, y=464) ~ R:66 G:80 B:119
```

```
All hulls = 12

All defects = 9

Defects filtered = 0

Command = NOACTION

enable commands (False, cooldown 0.00s.)

debug mode (False)
```

7.2 OUTPUT SCREENSHOTS

Home page:



All hulls = 11

All defects = 8

Defects filtered = 0

Command = NOACTION

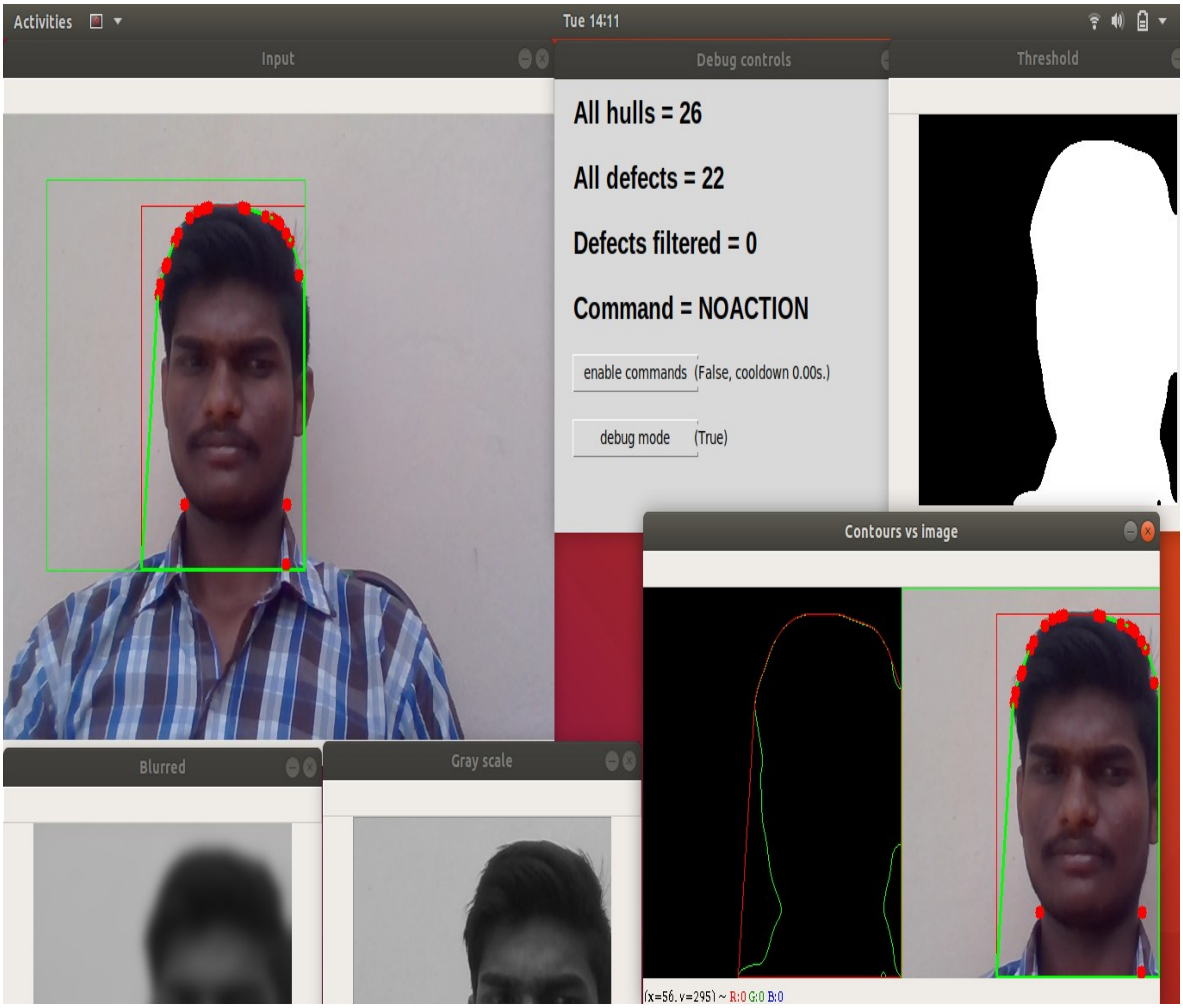
enable commands (False, cooldown 0.00s.)

debug mode (False)

(x=290, y=4) ~ R:193 G:188 B:188

Fig 7.2.1 Home page

Play-Processing Results:



All hulls = 26

All defects = 22

Defects filtered = 0

Command = NOACTION

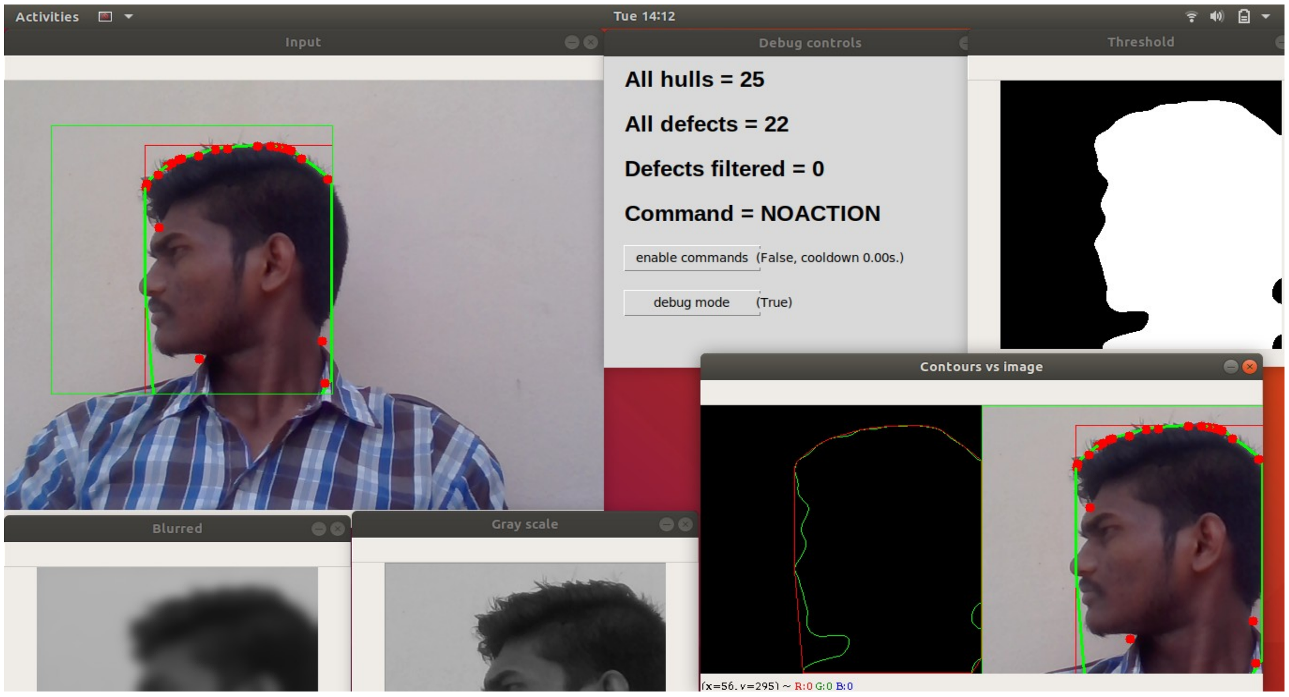
enable commands (False, cooldown 0.00s.)

debug mode (True)

(x=56, y=295) ~ R:0 G:0 B:0

Fig. 7.2.2 Play-Processing Results

Stop-Processing Results:



All hulls = 25

All defects = 22

Defects filtered = 0

Command = NOACTION

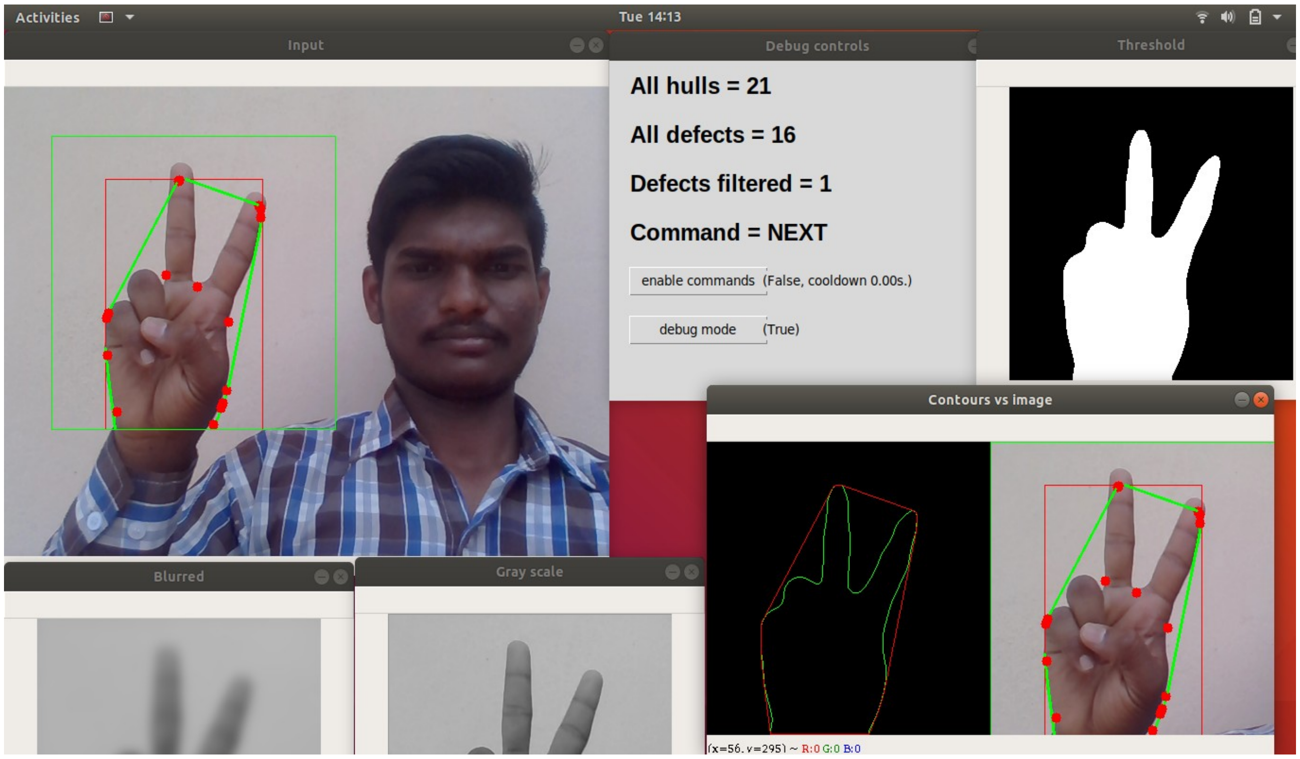
enable commands (False, cooldown 0.00s.)

debug mode (True)

(x=56, y=295) ~ R:0 G:0 B:0

Fig. 7.2.3 Pause-Processing Result

Next-Processing Results:



All hulls = 21

All defects = 16

Defects filtered = 1

Command = NEXT

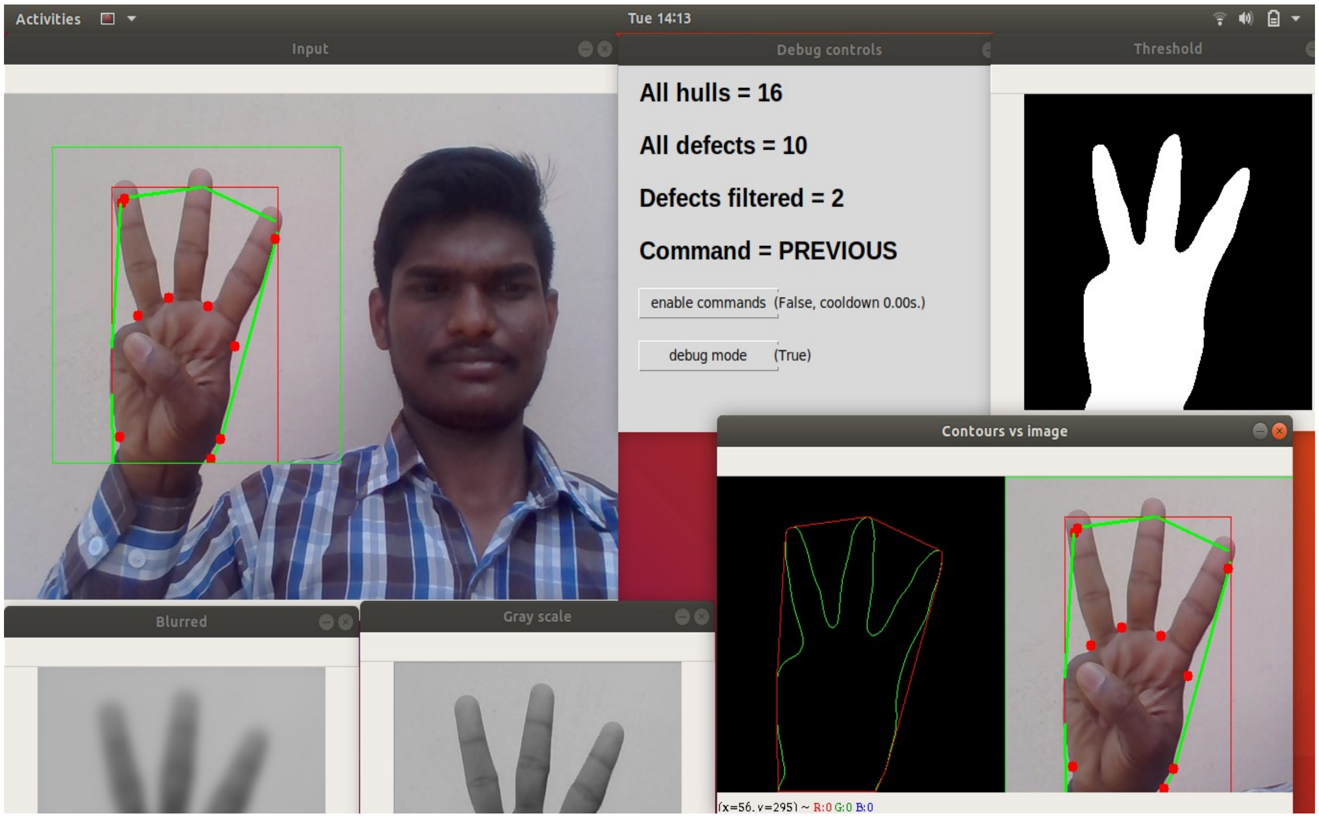
enable commands (False, cooldown 0.00s.)

debug mode (True)

(x=56, y=295) ~ R:0 G:0 B:0

Fig. 7.2.4 Next-Processing

Previous-Processing Results:



All hulls = 16

All defects = 10

Defects filtered = 2

Command = PREVIOUS

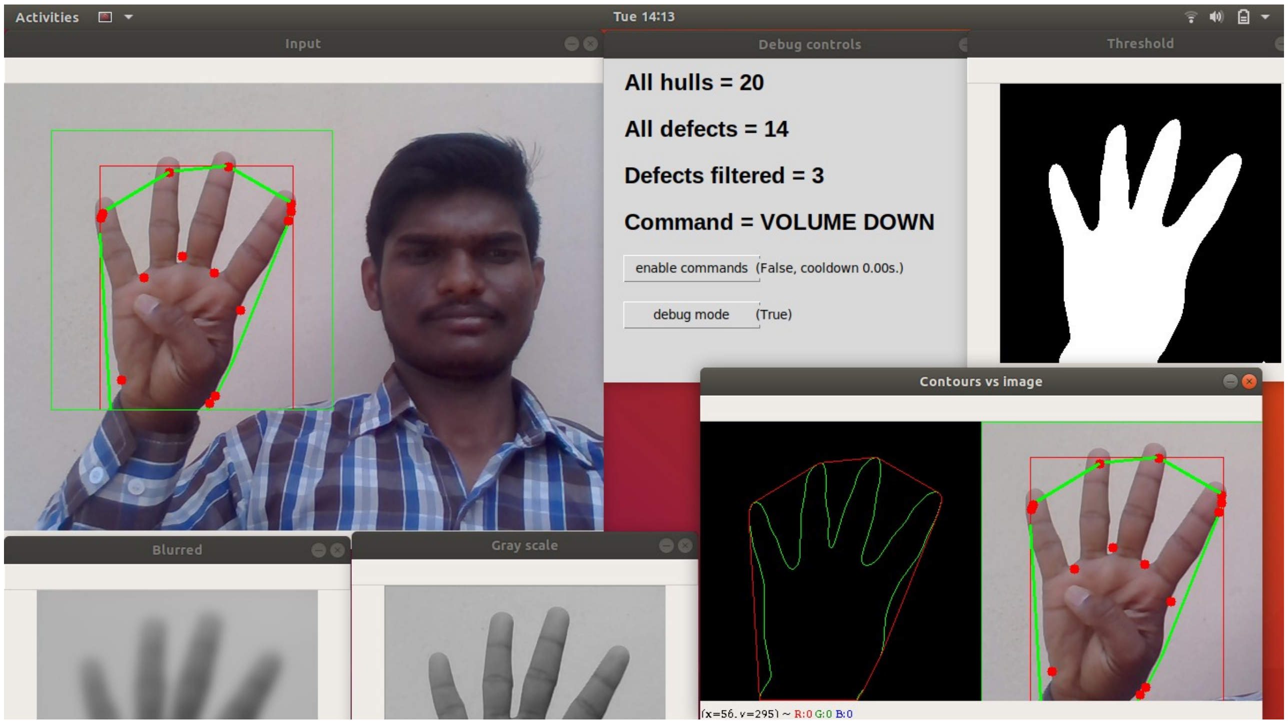
enable commands (False, cooldown 0.00s.)

debug mode (True)

(x=56, y=295) ~ R:0 G:0 B:0

Fig. 7.2.5 Previous-Processing

Volume-Down Processing Results:



All hulls = 20

All defects = 14

Defects filtered = 3

Command = VOLUME DOWN

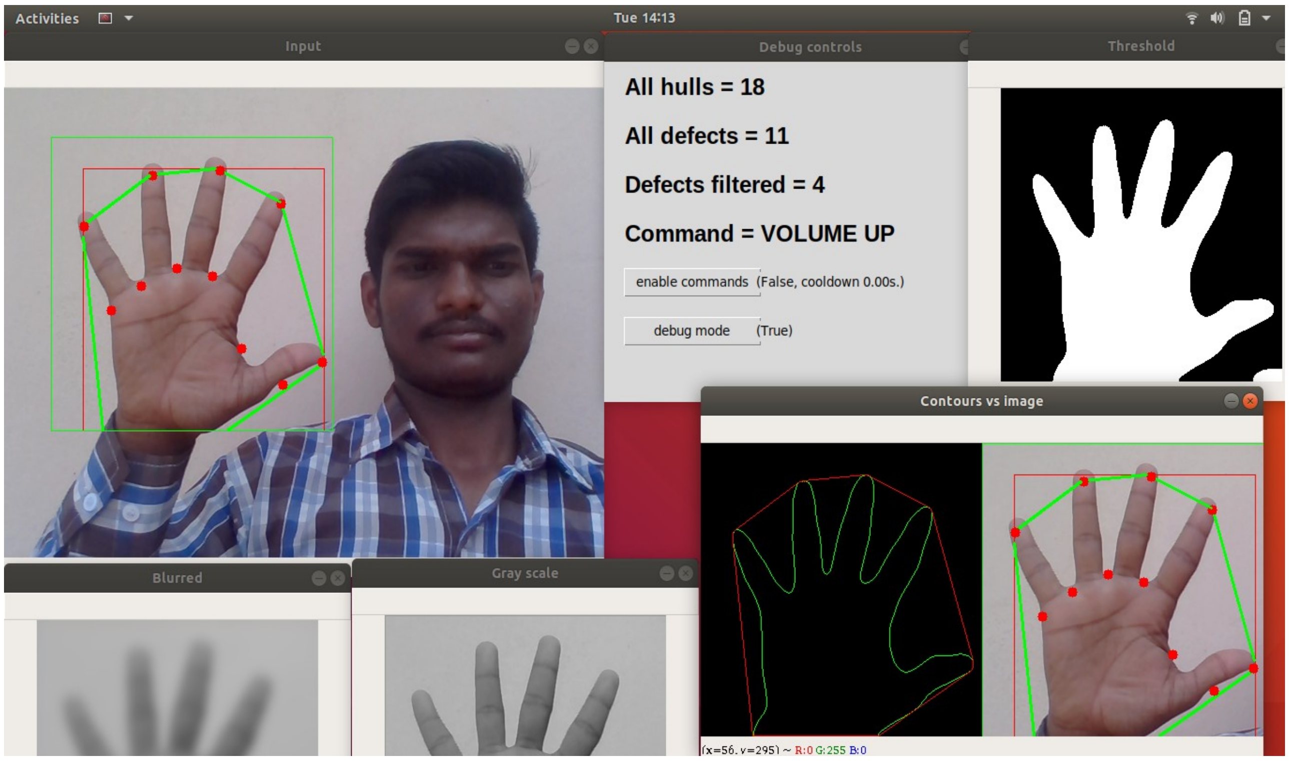
enable commands (False, cooldown 0.00s.)

debug mode (True)

(x=56, y=295) ~ R:0 G:0 B:0

Fig. 7.2.6 Volume-Down

Volume-Up Processing Results:



All hulls = 18

All defects = 11

Defects filtered = 4

Command = VOLUME UP

enable commands (False, cooldown 0.00s.)

debug mode (True)

(x=56, y=295) ~ R:0 G:255 B:0

Fig.7.2.7 Volume-Up

8. CONCLUSION & FUTURE WORK

8.1 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 and hand gestures 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), and controlling functions as forwarding , back-warding, volume up and volume down(for which hand gestures are used).

8.2 FUTURE WORK:

This project can be more interactive with the help of tracking real time hand movements and controlling mouse pointer on screen. The short coming of requiring a plain background can be overcome with the help of Background Image Subtractions or Machine Learning Techniques.

The performance of the proposed method highly depends on the result of hand detection. If there are moving objects with the color similar to that of the skin, the objects exist in the result of the hand detection and then degrade the performance of the hand gesture recognition.

9. APPENDIX

9.1 Introduction to UML:

UML Approach:

UML stands for Unified Modeling Language. UML is a language for specifying, visualizing and documenting the system. This is the step while developing any product after analysis. The goal from this is to produce a model of the entities involved in the project which later need to be built.

Definition: UML is a general purpose visual modeling language that is used to specify, visualize, construct, and document the artifacts of the software system.

UML is a language: It provides vocabulary and rules for Communications.These vocabulary and rules focus on conceptual and physical representation of a system. So, UML is standard language for software blueprint.

UML is a language for visualization: The UML more than just a bunch ofgraphical symbols. Rather behind each symbol in UML notation is a well defined semantics. In this manner, one we can write a model in UML, and other tools can also interpret that model.

UML is a language for specifying: Specifying means building models thatare precise, unambiguous and complete. In particular, the UML address the specification of all the important analysis, design and implementation decisions that must be made in developing and displaying a software intensive system.

UML is language for constructing: UML models can be directly connectedto a variety of programming languages and it is sufficiently expressive and free from any ambiguity to permit the direct execution of models.

UML is a language for Documenting: UML provides variety of documentsin addition raw executable codes. These artifacts include:

- Requirements
- Architecture
- Design
- Source Code
- Project Plans
- Tests
- Prototypes

Goal of UML:

The primary goal in the design of the UML is:

- Provide users with a ready-to-use, expressive visual modeling language so they can develop and exchange meaningful models.
- Provide extensibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development processes.
- Provide a formal basis for understanding the modeling language.
- Encourage the growth of the OO tools market.
- Support higher-level development concepts such as collaborations, frameworks, patterns and components.

Uses of UML:

The UML is intended primarily for software intensive systems. It has been used effectively for such domain as

1. Enterprise information system
2. Banking and financial services
3. Telecommunications
4. Transportations
5. Defense/Aerospace
6. Retails
7. Medical Electronics
8. Scientific Fields
9. Distributed Web

Rules of UML:

The UML has semantic rules for

- **Names:** it will call things, relationship and diagrams.
- **Scope:** the content that gives specific meaning to a name.
- **Visibility**: how those names can be seen and used by others.
- **Integrity:** how things properly and consistently relate to another.
- **Execution:** what it means is to run or simulate a dynamic model.

Building blocks of UML:

The vocabulary of the UML encompasses 3 kinds of building blocks:

1. Things
2. Relationships
3. Diagrams

1. **Things**: things are the data abstractions that are first class citizens ina model. Things are of 4 types
   - Structural Things
   - Behavioral Things
   - Grouping Things
   - An notational things
2. **Relationships**: Relationships tie the things together.Relationships
   in the UML are
   - Dependency
   - Association
   - Generalization
   - Specialization
3. **Diagrams**: Diagrams in the UML are of 2 types
   - Static Diagrams
   - Dynamic Diagrams

   - Class Diagram
   - Object Diagram
   - Component Diagram
   - Deployment Diagram

Dynamic Diagrams consists of

- Use case Diagram
- Sequence Diagram
- Collaboration Diagram
- State chart Diagram
- Activity Diagram

9.2 SOFTWARE ENVIRONMENT

OVERVIEW OF PYTHON:

Python is a general-purpose, object-oriented programming language with high-level programming capabilities. It has become famous because of its apparent and easily understandable syntax, portability, and easy to learn.

Python’s flexibility is why the first step in every Python project must be to think about the project’s audience and the corresponding environment where the project will run. It might seem strange to think about packaging before writing code, but this process does wonders for avoiding future headaches.

Python is a programming language that includes features of C and Java. It provides the style of writing an elegant code like C, and for object-oriented programming, it offers classes and objects like Java. Python is used to create web and desktop applications, and some of the most popular web applications like Instagram, YouTube, Spotify all have been developed in Python. You can also develop the next big thing by using Python.

For a long timemany operating systems, including Mac and Windows, lacked built-in package management. Only recently did these OSes gain so-called “app stores”, but even those focus on consumer applications and offer little for developers.

Computing as we know it is defined by the ability to execute programs. Every operating system natively supports one or more formats of program they can natively execute.

Features:

Python is gaining good popularity in the programming community; there are many reasons behind this.

- **Interpreted Language:** Python is processed at run time by Python Interpreter.
- **Object-Oriented Language:** It supports object-oriented features and techniques of programming.
- **Interactive Programming Language:** Users can interact with the python interpreter directly for writing programs.
- **Easy language:** Python is easy to learn, especially for beginners.
- **Straightforward Syntax:** The formation of python syntax is simple and straightforward, which also makes it popular.
- **Easy to read:** Python source-code is clearly defined and visible to the eyes.
- **Portable:** Python codes can be run on a wide variety of hardware platforms having the same interface.
- **Extendable:** Users can add low level-modules to Python interpreter.
- **Scalable:** Python provides an improved structure for supporting large programs then shell-scripts.

Advantages:

The diverse application of the Python language is a result of the combination of features which give this language an edge over others. Some of the benefits of programming in Python include:

1. **Presence of Third Party Modules**
2. **Extensive Support Libraries**
3. **Open Source and Community Development**
4. **Learning Ease and Support Available**
5. **User-friendly Data Structures**
6. **Productivity and Speed**

OVERVIEW OF OPENCV:

OpenCV is the huge open-source library for the computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today’s systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human. When it integrated with various libraries, such as Numpuy, python is capable of processing the OpenCV array structure for analysis. To Identify image pattern and its various features we use vector space and perform mathematical operations on these features.

The first OpenCV version was 1.0. OpenCV is released under a BSD license and hence it’s free for both **academic** and **commercial** use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. When OpenCV was designed the main focus was real-time applications for computational efficiency. All things are written in optimized C/C++ to take advantage of multi-core processing.

Applications of OpenCV: There are lots of applications which are solved using OpenCV, some of them are listed below

- Face recognition
- Automated inspection and surveillance
- number of people – count (foot traffic in a mall, etc)
- Vehicle counting on highways along with their speeds
- Interactive art installations
- Anamoly (defect) detection in the manufacturing process (the odd defective products)
- Street view image stitching
- Video/image search and retrieval
- Robot and driver-less car navigation and control
- object recognition
- Medical image analysis
- Movies – 3D structure from motion
- TV Channels advertisement recognition

OpenCV Functionality:

- Image/video I/O, processing, display (core, img proc, high gui)
- Object/feature detection (obj detect, features2d, non free)
- Geometry-based monocular or stereo computer vision (calib3d, stitching, videos tab)
- Computational photography (photo, video, super res)
- Machine learning & clustering (ml, flann)
- CUDA acceleration (gpu)

Image-Processing:

Image processing is a method to perform some operations on an image, in order to get an enhanced image and or to extract some useful information from it.

If we talk about the basic definition of image processing then **“**Image processing is the analysis and manipulation of a digitized image, especially in order to improve its quality**”.**

Digital-Image :

An image may be defined as a two-dimensional function f(x, y), where x and y are spatial(plane) coordinates, and the amplitude of fat any pair of coordinates (x, y) is called the intensity or grey level of the image at that point.

In another word An image is nothing more than a two-dimensional matrix (3-D in case of coloured images) which is defined by the mathematical function f(x, y) at any point is giving the pixel value at that point of an image, the pixel value describes how bright that pixel is, and what colour it should be.

Image processing is basically signal processing in which input is an image and output is image or characteristics according to requirement associated with that image.

Image processing basically includes the following three steps:

1. Importing the image
2. Analysing and manipulating the image
3. Output in which result can be altered image or report that is based on image analysis

It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDA and OpenCL interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a templated interface that works seamlessly with STL containers.

10. REFERENCES

REFERENCES:

[1] “Controlling Multimedia Applications Using Hand Gesture Recognition” by Neha S. Rokade, Harsha R. Jadhav, Sabiha A. Pathan , Uma Annamalai.

[2] “OpenCV-Python Tutorials Documentation” by Alexander Mordvintsev & Abid K.

[3] “Enhanced Look Based Media Player With Hand Gesture Recognition” by Harshada Naroliya, Tanvi Desai, Shreeya Acharya, Varsha Sakpal.

[4] Viola and Jones, "Rapid object detection using a boosted cascade of simple features", Computer Vision and Pattern Recognition, 2001

[5] A Vision based Hand Gesture Interface for Controlling VLC Media Player,Siddharth Swarup Rautaray,Indian Institute of Information Technology,Allahabad,Anupam Agrawal,Indian Institute of Information Technology,Allahabad

[6] Viola and Jones, "Rapid object detection using a boosted cascade of simple features", Computer Vision and Pattern Recognition, 2001.

[7] Mr. Sanjay Mena, “A Study on Hand Gesture Recognition”, White Paper.

[8] Ruslana Makovetsky and Gayane Persian,“Face Detection and Tracking with Web Camera”, White paper.