iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** VI **Month of publication:** June 2021

DOI: https://doi.org/10.22214/ijraset.2021.35421

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VI Jun 2021- Available at www.ijraset.com

Media-Player Controlling by Hand Gestures

D. S. S. Varshika[1], Gopu Joseph Priyanka Reddy[2], B. Chandana[3], Dr. S N Chandra Shekhar[4], Dr. K. Sateesh Kumar[5]
[1, 2, 3]B.Tech IV year, ECE, [4]Assistant Professor, [5]Associate Professor, Department of Electronics and Communication Engineering, Sreenidhi Institute of Science and Technology, Hyderabad, Telangana, India.

Abstract: ***In this Project we try to control our media player using hand gestures with the help of OpenCV and Python. Computer applications require interaction between human and computer. This interaction needs to be unrestricted and it has made it challenging to traditional input devices such as keyboard, mouse, pen etc. Hand gesture is an important component of body languages in linguistics. Human computer interaction becomes easy with the use of the hand as a device. Use of hand gestures to operate machines would make interaction interesting. Gesture recognition has gained a lot of importance. Hand gestures are used to control various applications like windows media player, robot control, gaming etc. Use of gesture makes interaction easy, convenient and does not require any extra device. Vision and audio recognition can be used together. But audio commands may not work in noisy environments.***

Keywords: Object, Detection, Python OpenCV

I. INTRODUCTION

With the development in Computer Vision and Human Machine Interaction the Computer holds most important role in our daily life. Human Computer Interaction can provides several advantages with the introducing the different natural forms of device free communication. Gesture recognition is one of the several types of them to interact with the humans gestures are the natural form of action which we often used in our day to day life.

But in computer application to interact humans with machine the interaction with devices like keyboard, mouse etc. must be requires. As the various hand gestures are frequently used by humans so the aim of this project is to reduce external hardware interaction which is required for computer application, and hence this causes system more reliable for use with ease. This paper implements gesture based recognition technique to handing multimedia application. In this system, a gesture recognition scheme is been proposed as an interface between human and machine.

In our system we represent some low-complexity algorithm and some hand gestures to decrease the gesture recognition complexity and which becomes easier to control real-time systems.

A. Python Libraries Used

NumPy OpenCV-Python OpenCV is a Computer Vision library. It is a collection of C functions with a few C++ classes that implement popular Image Processing and Computer Vision algorithms. Computer vision is the science that means to give a comparative, if not better capacity to a machine or PC. Computer vision is worried about the programmed extraction, investigation and comprehension of valuable data from a single picture or a grouping of pictures. Some of the basic image processing capabilities include filtering, edge detection, corner detection, sampling and interpolation, color conversion, morphological operations, histograms and many more. Color detection using OpenCV has many advantages like, it allows the detection of a specific color in a livestream video content. In this OpenCV color detection system there are four major modules, activated webcam, scan object, match frame parts and system results. Users can open webcam by clicking the webcam button.

II. LITERATURE SURVEY

In 2015, Chong Wang, “Super pixel-Based Hand Gesture Recognition with Kinect Depth Camera” proposed the system which uses kinect depth camera. It is based on a compact representation in the form of super pixels, which efficiently capture the shape, texture and depth features of the gestures. Since this system uses kinect depth camera, the cost of system is more.

In 2014, Swapnil D. Badgujar,“Hand Gesture Recognition System” proposed the system which recognize the unknown input gestures by using hand tracking and extraction method. This system is applied to recognize the single gesture. There is assumption of stationary background so that system will have smaller search region for tracking. This system only control mouse with the finger using it on web cam.

In 2014, Viraj Shinde, Tushar Bacchav, Jitendra Pawar and Mangesh Sanap developed "Hand Gesture Recognition System Using Camera". They focus on using pointing behaviors for a natural interface to classify the dynamic hand gesture, they developed a simple and fast motion history image based method. This paper presents low complexity algorithm and gestures recognition complexity and more suitable for controlling real time computer system. It is applicable only for the application Of power point presentation.

In 2014, N. Krishna Chaitanya and R. Janardhan Rao presents "Controlling of windows media player application using hand gesture recognition", this system uses various hand gestures as input to operate the windows media player application. This system uses single hand gestures and its directional motion which defines a particular gesture for the above mentioned application. In this system decision tree has been used for classification. This system only supports windows media player application and not any others.

In 2012, Ram Rajesh J., Sudharshan R., Nagarjunan D. and Aarthi R., "Remotely controlled PowerPoint presentation navigation using hand gestures" developed the system in which slides of power point presentation are controlled without using any marker and gloves. In this system the developer used the segmentation algorithm for hand detection. After detecting hand calculation is for active figures. If the fingers are not stretched properly while making a gesture then application did not work properly.

In 2012, Ruize Xu, Shengli Zhou and Wen J. Li, "MEMS Accelerometer Based Non specific-User Hand Gesture Recognition", had build a system which can recognize various hand gestures such as up, and down, right, and left, cross and round. Three various modules were developed that recognizes various hand gestures. The Signals by MEMS (MicroElectromechanical System) 3- axes accelerometers were provided as input. The motion of the hand in three perpendicular direction is been detected by three accelerometers and send to the system by Bluetooth. Segmentation algorithm was been applied and finally the various hand gestures were recognized by matching gestures that were already saved in the system. People mostly prefer the internet to have daily update on weather, news etc. So for this purpose they perform keyboard and mouse operations. This system gives less accuracy in finding the terminal points of gestures due to small size of database of hand gesture

III. EXISTING METHODOLOGY

This Many various systems have been developed that are being controlled by gesture. These systems consist of games, sign language recognition, all these systems can be controlled by facial gestures, hand gestures can also control mouse. A system was developed in 2012 that recognizes seven various hand gestures consists of various gestures such as up, and down, right, and left, cross and round.

Three various modules were built in this system to recognize various hand gestures. Signals by MEMS 3-axes accelerometers were been given as input to the system.

The gesture of the hand in three perpendicular directions is been detected by 3 accelerometers and been transmitted to the system by Bluetooth. Segmentation algorithm has been applied and finally the gestures are recognized and compares with the gesturers already been saved in the system. People get daily information about news weather etc with the use of the internet. To get these above information people have to use mouse and keyboard which can be prevented by this system.

An article was been presented in 2011 by Ginu Thomas, A Review of Various Hand Gesture Recognition Techniques in which he compared the results achieved by several hand gesture recognitions techniques present. The various techniques used are edges method, and pixel by pixel comparison, orientation histogram.

A database has been used that store various static hand gestures inputs. These inputs were the subset of ASL i.e. American sign languages. Filtering of the input image has been done to remove the noise present in the input image and then segmentation was done to the input image to analyze it. The input image was then converted into feature vector and then it was compared with the stored, trained set of hand gestures.

A system developed by Anupam Agrawal in 2010 had various used hand gestures to operate the VLC media player application. The K nearest neighbour algorithm has been used to recognize the various hand gestures. A VLC media player system that has been controlled by various hand gestures consists of play, and pause, Full screen, and stop, increase volume, and decrease volume features.

Lucas Kanade Pyramidical Optical Flow algorithm has been used to detect hand gestures from the input video. This algorithm present in the system detects moving points in the input received by the input video. After this K -MEAN has been used to locate the centre of the hand. Using this centre point also known as centroid of the hand, hand is been recognized. The above mentioned system used a database that consists of various hand gestures and then input image was been compared with this saved image and accordingly the purposed output command was been performed by VLC media player.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VI Jun 2021- Available at www.ijraset.com

IV. PROPOSED SYSTEM

A. HSV Scale Image

First the image is captured by the web-cam, then various image processing is done on it. The original image is then converted into HSV ie Hue Saturation and Value. This is done to detect the portion of the hand and separate it from the background. HSV defines a type of color space. Value is defined as brightness. In HSV, hue represents a color. In this system we have considered Hue in a range from 0 to 20. Saturation indicates the range of grey in the color space. We have considered saturation range from 55 to 255. Value is brightness of the color and varies with color saturation. In this system we have considered a range from 0 to 255 for Value, when the value is 0 the color space will be totally black. With the increase in the value, the color space brightens and shows various colors.

B. Threshold Image

In this module we have obtained threshold image using HSV ranges of colour detection. Here we find the biggest contour for hand detection. Find Biggest Contour() with the use of OpenCV function i.e. cvFindContours() to make a list of various contours. In order to binary threshold image, a contour is a region of white pixels. Each region is approximated by a bounding box, and the contour corresponding to the largest box is taken and transmitted.

C. Filtered Image

In this module we extract the noisy pixels detected other than hand border. By calling extractContourInfo () we extract noisy pixels and analyze the contour.

D. Calculate Center Of Gravity

In section, the box surrounding the contour is used to receive a centre. This is enough as the underlying shape is a rectangular card, due to which the contour and box are almost same. Therefore, a rounding box around a hand can simply have other COG or angle from the hand itself; in such case, it is very important to analyze the hand contour other than a rounding box, by using moments. In this system we used spatial moments to obtain the Centre of Gravity of an input binary image. This same method can be used to a contour to obtain its center or centroid. In this we can calculate second ordered mixed moments, which will give info about the spread of pixels around the centroid. Second order moments can be brought together to return the angle of the contour's major axis with respect to the x-axis. The OpenCV moments notation, the m () moments function takes following two arguments, a and b, which are been used as powers for x and y. The I() function is the intensity for the pixel defined by its (x, y) coordinate. s is the no of pixels that make up the shape. Then we take the contour, then is the angle of its major axis to the horizontal, with the +yaxis pointing downwards.

V. PROJECT DESCRIPTION

This hand gesture recognition technique will not only replace the use of mouse to control the VLC player but also provide different gesture vocabulary which will be useful in controlling the application.
Develops a vision based low cost input device for controlling the VLC player.
Provides device free communication.
Helpful to dumb people & can also use this while driving.

A. Code Requirements

This application is written in Python 3.6 and it uses the very famous OpenCV library. OpenCV is a computer vision and machine learning software library that includes many common image analysis algorithms that will help us build custom, intelligent computer vision applications.

B. Video Tracking

To perform video tracking, an algorithm analyzes sequential video frames and outputs the movement of targets between the frames. There are a variety of algorithms, each having strengths and weaknesses. Considering the intended use is important when choosing which algorithm to use. There are two major components of a visual tracking system: target representation and localization, as well as filtering and data association.

Video tracking is the process of locating a moving object (or multiple objects) over time using a camera. It has a variety of uses, some of which are: human-computer interaction, security and surveillance, video communication and compression, augmented reality, traffic control, medical imaging and video editing.

C. Steps for Implementation:

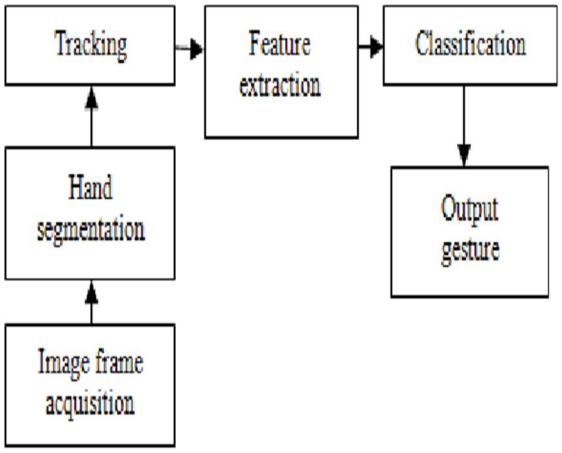


Fig1: Implementation block diagram

1) Step 1: Import Libraries and capture camera
2) Step 2: Convert frames Into hsv
3) Step 3: Track hand on color basis
4) Step 4: Create mask on the basis of color and filter actual color
5) Step 5: Invert pixel value and then enchance the result for better output
6) Step 6: Find Contours for specific colored object
7) Step 7: Find Max area contour and draw it on live feed
8) Step 8: Find Convexity detect for counting Values and Apply Cosin method
9) Step 9: Bind hand gestures with keyboard keys.
10) Step 10: Enjoy your output

VI. CODE

```
#Step -1
import import_ipynb
import cv2
import numpy as np
import math
import pyautogui as p
import time as t

#Read Camera
cap = cv2.VideoCapture(0,cv2.CAP_DSHOW)
def nothing(x):
    pass
#window name
cv2.namedWindow("Color Adjustments",cv2.WINDOW_NORMAL)
cv2.resizeWindow("Color Adjustments", (300, 300))
cv2.createTrackbar("Thresh", "Color Adjustments", 0, 255, nothing)
```

```
#COlor Detection Track
cv2.createTrackbar("Lower_H", "Color Adjustments", 0, 255, nothing)
cv2.createTrackbar("Lower_S", "Color Adjustments", 0, 255, nothing)
cv2.createTrackbar("Lower_V", "Color Adjustments", 0, 255, nothing)
cv2.createTrackbar("Upper_H", "Color Adjustments", 255, 255, nothing)
cv2.createTrackbar("Upper_S", "Color Adjustments", 255, 255, nothing)
cv2.createTrackbar("Upper_V", "Color Adjustments", 255, 255, nothing)


while True:
    _,frame = cap.read()
    frame = cv2.flip(frame,2)
    frame = cv2.resize(frame,(600,500))
    # Get hand data from the rectangle sub window
    cv2.rectangle(frame, (0,1), (300,500), (255, 0, 0), 0)
    crop_image = frame[1:500, 0:300]

    #Step -2
    hsv = cv2.cvtColor(crop_image, cv2.COLOR_BGR2HSV)
    #detecting hand
    l_h = cv2.getTrackbarPos("Lower_H", "Color Adjustments")
    l_s = cv2.getTrackbarPos("Lower_S", "Color Adjustments")
    l_v = cv2.getTrackbarPos("Lower_V", "Color Adjustments")

    u_h = cv2.getTrackbarPos("Upper_H", "Color Adjustments")
    u_s = cv2.getTrackbarPos("Upper_S", "Color Adjustments")
    u_v = cv2.getTrackbarPos("Upper_V", "Color Adjustments")
    #Step -3
    lower_bound = np.array([l_h, l_s, l_v])
    upper_bound = np.array([u_h, u_s, u_v])

    #Step - 4
    #Creating Mask
    mask = cv2.inRange(hsv, lower_bound, upper_bound)
    #filter mask with image
    filtr = cv2.bitwise_and(crop_image, crop_image, mask=mask)

    #Step - 5
    mask1  = cv2.bitwise_not(mask)
    m_g = cv2.getTrackbarPos("Thresh", "Color Adjustments") #getting track bar value
    ret,thresh = cv2.threshold(mask1,m_g,255,cv2.THRESH_BINARY)
    dilata = cv2.dilate(thresh,(3,3),iterations = 6)

    #Step -6
    #findcontour(img,contour_retrival_mode,method)
    cnts,hier = cv2.findContours(thresh,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)


    try:
        #print("try")
```

```
 #Step -7
 # Find contour with maximum area
cm = max(cnts, key=lambda x: cv2.contourArea(x))
#print("C==",cnts)
epsilon = 0.0005*cv2.arcLength(cm,True)
data= cv2.approxPolyDP(cm,epsilon,True)

hull = cv2.convexHull(cm)

cv2.drawContours(crop_image, [cm], -1, (50, 50, 150), 2)
cv2.drawContours(crop_image, [hull], -1, (0, 255, 0), 2)

#Step - 8
# Find convexity defects
hull = cv2.convexHull(cm, returnPoints=False)
defects = cv2.convexityDefects(cm, hull)
count_defects = 0
#print("Area==",cv2.contourArea(hull) - cv2.contourArea(cm))
for i in range(defects.shape[0]):
    s,e,f,d = defects[i,0]

    start = tuple(cm[s][0])
    end = tuple(cm[e][0])
    far = tuple(cm[f][0])
    #Cosin Rule
    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)) * 180) / 3.14
    #print(angle)
    # if angle <= 50 draw a circle at the far point
    if angle <= 50:
        count_defects += 1
        cv2.circle(crop_image,far,5,[255,255,255],-1)

print("count==",count_defects)

#Step - 9
# Print number of fingers
if count_defects == 0:

    cv2.putText(frame, " ", (50, 50), cv2.FONT_HERSHEY_SIMPLEX, 2,(0,0,255),2)
elif count_defects == 1:

    p.press("space")
    cv2.putText(frame, "Play/Pause", (50, 50), cv2.FONT_HERSHEY_SIMPLEX, 2,(0,0,255), 2)
elif count_defects == 2:
    p.press("up")

    cv2.putText(frame, "Volume UP", (5, 50), cv2.FONT_HERSHEY_SIMPLEX, 2,(0,0,255), 2)
```

```
        elif count_defects == 3:
            p.press("down")

            cv2.putText(frame, "Volume Down", (50, 50), cv2.FONT_HERSHEY_SIMPLEX, 2,(0,0,255), 2)
        elif count_defects == 4:
            p.press("right")

            cv2.putText(frame, "Forward", (50, 50), cv2.FONT_HERSHEY_SIMPLEX, 2,(0,0,255), 2)
        else:
            pass

    except:
        pass
    #step -10
    cv2.imshow("Thresh", thresh)
    #cv2.imshow("mask==",mask)
    cv2.imshow("filter==",filtr)
    cv2.imshow("Result", frame)

    key = cv2.waitKey(25) &0xFF
    if key == 27:
        break
cap.release()
cv2.destroyAllWindows()
```

VII. EXPERIMENTAL RESULTS

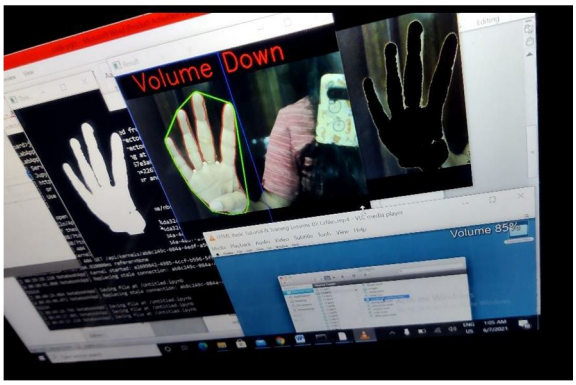


Fig2: Result of volume down

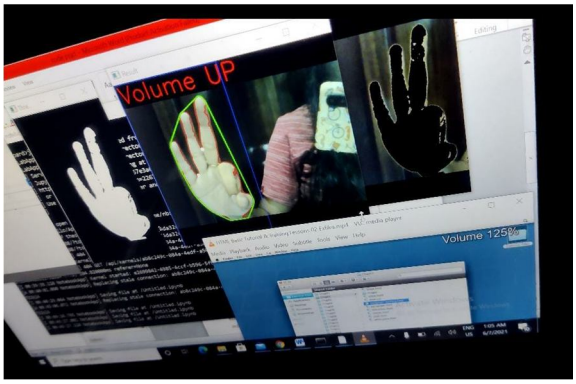


Fig3: Result of volume up

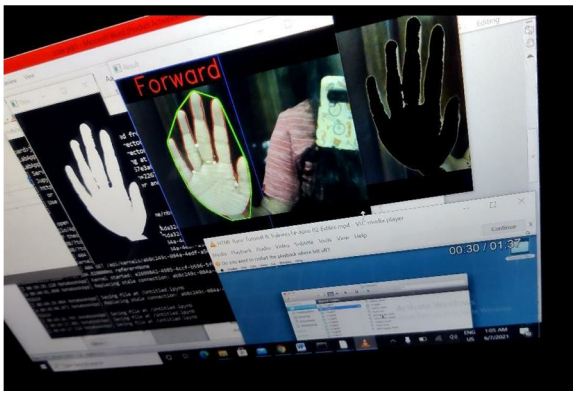


Fig4: Result of forward

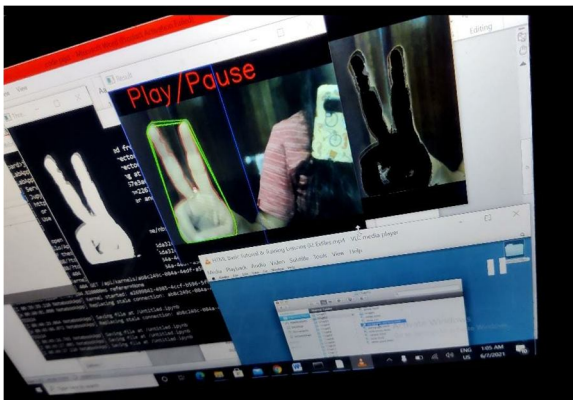


Fig5: Result of play/pause

VIII. MERITS AND DEMERITS

A. Advantages

1) The system is user-friendly and has simple interface.

2) Can be used in manufacturing company

B. Limitation

1) Data need to be entered properly otherwise, outcome may won't be accurate

IX. CONCLUSION

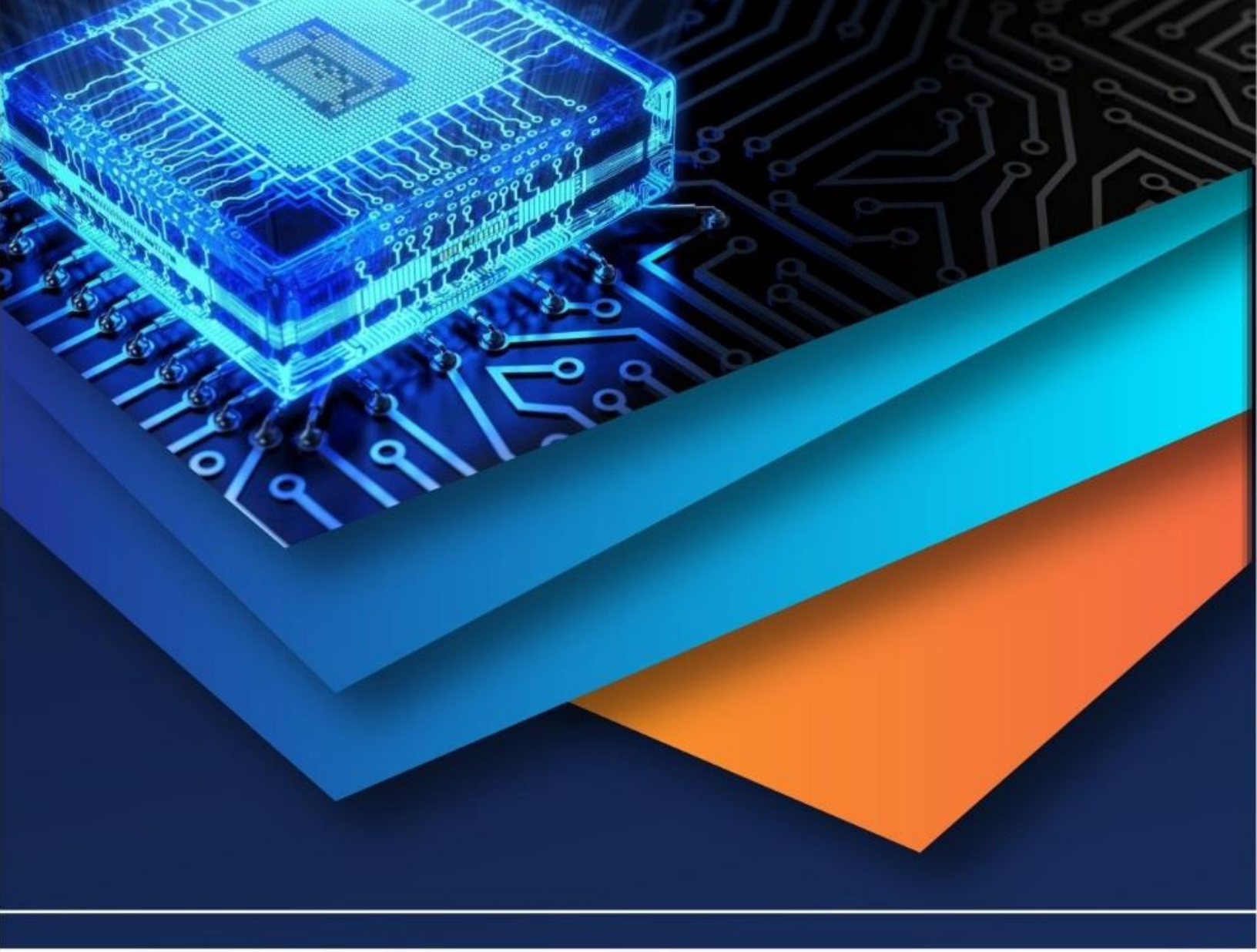
Hence, In proposed work, the gestures are used to control the multiple applications like PDF reader, multimedia player and power point presentation by avoiding the physical contact with mouse and keyboard. By using the gesture commands any one can use the system to operate different applications. In some previously implemented system the costly 3-D sensors like kinect are used for gesture recognition. To reduce the cost we are using simple web camera. The separate training set is not require to recognize the gestures, so there is no need to maintain any database for storing the frames of images. Our focus in future is on the extending the gestures for the gaming and voice for mute.

X. ACKOWLEDGEMENT

Firstly, we are grateful to Sreenidhi Institute of Science and Technology for giving us the opportunity to work on this project. We are fortunate to have worked under the supervision of our internal guide Dr. S N Chandra Shekhar, guide Dr. J. Chattopadhyay. His guidance and ideas have made this project work. We are thankful to Dr.K.Sateesh kumar for being the in charge for this project and conduction reviews. We are also thankful to the HOD of Electronics and Communication Engineering, Dr. S.P.V. Subba Rao for giving us access to all the resources that went into building this project.

REFERENCES

[1] N.Krishna Chaitanya and R.Janardhan Rao "Controlling of windows media player using hand recognition system", The International Journal Of Engineering And Science (IJES), Vol. 3, Issue 12, Pages 01-04, 2014.
[2] Chong Wang, Zhong Liu and Shing-Chow Chan Superpixel-Based Hand Gesture Recognition WithKinect Depth Camera", IEEE Trans. Multimedia, vol. 17, no. 1, Jan. 2015.
[3] Swapnil D. Badgujar, Gourab Talukdar, Omkar Gondhalekar and Mrs. S.Y. Kulkarni "Hand Gesture Recognition System", International Journal of Scientific and Research Publications, vol. 4, Issue 2, Feb. 2014, pp. 2250-3153.
[4] Viraj Shinde, Tushar Bacchav, Jitendra Pawar and Mangesh Sanap "Hand Gesture Recognition System Using Camera", International Journal of Engineering Research and Technology (IJERT), Vol. 3, Issue 1, January – 2014.
[5] Anupam Agrawal and Siddharth Swarup Rautaray "A Vision based Hand Gesture Interface for Controlling VLC Media Player", International Journal of Computer Applications, vol.10, no.7, Nov.2010.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY