Air Canvas

A Social Internship

Report

Submitted in the partial fulfillment of the requirements for the

award of the degree of

Bachelor of Technology

in

Department of Computer Science and Engineering

By

Hemanth Srivathsav (2010030113)

Shreya Chintawar (2010030155)

Peri Vishwanadha Sastry (2010030470)

Komuravelli Sai Kamal (2010030532)

under the supervision of

Dr. KVSN Rama Rao

Professor

(DEEMED TO BE UNIVERSITY)

Department of Computer Science and Engineering

K L University Hyderabad,

Aziz Nagar, Moinabad Road, Hyderabad – 500 075, Telangana, India.

March, 2022

DECLARATION

The Social Internship Report entitled "Air Canvas" is a record of bonafide work of Hemanth Srivathsav(2010030113),Shreya Chintawar(2010030155) ,Peri Vishwanadha Sastry (2010030470) ,Komuravelli Sai Kamal(2010030532) submitted in partial fulfillment for the award of B.Tech in the Department of Computer Science and Engineering to the K L University, Hyderabad. The results embodied in this report have not been copied from any other Departments/ University/ Institute.

<Signature of the Students >

Vishwanadha

(PERI VISHWANADHA SASTRY)

(N. Hemanth Srivathsav)

(Shreya Chintawar)

Kamal (K. Sai Kamal)

CERTIFICATE

This is to certify that the Social Internship Report entitled "Air Canva" is being submitted by Hemanth Srivathsav(2010030113), Shreya Chintawar(2010030155), Peri Vishwanadha Sastry (2010030470) , Komuravelli Sai Kamal(2010030532)submitted in partial fulfillment for the award of B.Tech in CSE to the K L University, Hyderabad is a record of bonafide work carried out under our guidance and supervision.

The results embodied in this report have not been copied from any other departments/ University/institutes.

Signature of the Supervisor

Dr. KVSN Rama Rao

Professor

Signature of the HOD

Signature of the External Examiner

ACKNOWLEDGEMENT

First and foremost, we thank the lord almighty for all his grace & mercy showered upon us, for completing this Social Internship successfully.

We take grateful opportunity to thank our beloved **Founder and Chairman** who has given constant encouragement during our course and motivated us to do this Social Internship. We are grateful to our Principal **Dr. L. Koteswara Rao** who has been constantly bearing the torch for all the curricular activities undertaken by us.

We pay our grateful acknowledgement & sincere thanks to our Head of the Department **Dr. Chiranjeevi Manike** for her exemplary guidance, monitoring and constant encouragement throughout the course of the Social Internship. We thank **Dr. KVSN Rama Rao** of our department who has supported throughout this Social Internship holding a position of supervisor.

We whole heartedly thank all the teaching and non-teaching staff of our department without whom we won't have made this Social Internship a reality. We would like to extend our sincere thanks especially to our parent, our family members and friends who have supported us to make this Social Internship a grand success.

ABSTRACT

This paper presents a real-time video-based pointing method that allows sketching and writing text over the air in front of a camera. The proposed method tracks the fingertips of the thumb and index fingers through the Video that is captured from a webcam or any other device. Tracking of the Hand landmarks is used in order to achieve the objective.

Static chalkboards and paper-based lessons don't connect with students in the digital age. Teachers forced to rely on chalk to reach students are doomed to fail. Forcing lessons into lectures or on chalkboards in the classroom will lead students to tune out before the class starts.

Interactive smart boards invite students to engage with the lessons. Teachers aren't limited in what they can present to students. Movies, PowerPoint presentations, and graphics can be used in addition to standard text-based lessons. In this blog, we'll take a look at smartboard technology in the classroom and how teachers can better engage with students.

Writing in the air has been one of the most fascinating and challenging research areas in the field of image processing and pattern recognition in recent years. It contributes immensely to the advancement of an automation process and can improve the interface between man and machine in numerous applications. Several research works have been focusing on new techniques and methods that would reduce the processing time while providing higher recognition accuracy.

The project takes advantage of this gap and focuses on developing motion-tracking and mapping the motion that can potentially serve as a software for writing from the air. It will use computer vision to trace the path of the finger.

It can be an effective communication method that reduces mobile and laptop usage by eliminating the need to write.

TABLE OF CONTENT

S.no	Topics	Page Number
1.	Introduction	9
2.	Literature Survey	10
3.	Methodology	11
4.	Implementation	12 - 15
5.	Result Discussion	16 - 18
6.	Conclusion	19
7.	Social Contribution	20
8.	References	21 - 22

1. INTRODUCTION

In the era of digital world, traditional art of writing is being replaced by digital art. Digital art refers to forms of expression and transmission of art form with digital form. Relying on modern science and technology is the distinctive characteristics of the digital manifestation. Traditional art refers to the art form which is created before the digital art. From the recipient to analyse, it can simply be divided into visual art, audio art, audio-visual art and audio-visual imaginary art, which includes literature, painting, sculpture, architecture, music, dance, drama and other works of art. Digital art and traditional art are interrelated and interdependent. Social development is not a people's will, but the needs of human life are the main driving force anyway. The same situation happens in art. In the present circumstances, digital art and traditional art are inclusive of the symbiotic state, so we need to systematically understand the basic knowledge of the form between digital art and traditional art. The traditional way includes pen and paper, chalk and board method of writing. The essential aim of digital art is of building hand gesture recognition system to write digitally. Digital art includes many ways of writing like by using keyboard, touch-screen surface, digital pen, stylus, using electronic hand gloves, etc. But in this system, we are using hand gesture recognition with the use of machine learning algorithm by using python programming, which creates natural interaction between man and machine.

Sketching On Air is possible through technology namely open cv, python. Open cv is mainly known as an open-source computer vision and machine learning software. Most of these algorithms are used to detect and recognize faces, identify objects, classify human activities in videos track camera movements, track moving objects, extract 3D ones.

Air Canvas aims to be an interface between man and machine in various applications of E-Learning and students to experience fun-based learning and increase their confidence throughout their learning journey. The techniques and methods used here would reduce the consumption of resources while replicating the results and providing the same functionality.

2. LITERATURE SURVEY

S.no	Title	Authors	Publishing	Techniques & dataset	Pros	Cons
1.	A. Robust Hand Recognition with Kinect Sensor	[3] Saira Beg, M. Fahad Khan and Faisal Baig, "Text Writing in Air," Journal of Information Display Volume 14, Issue 4, 2013	International Research Journal of Engineering and Technology (IRJET)	the system proposed used the depth and colour information from the Kinect sensor to detect the hand shape. As for gesture recognition, even with the Kinect sensor.	The resolution of this Kinect sensor is only 640×480. It works well to track a large object, e.g., the human body.	It is still a very challenging problem. But tracking a tiny thing like a finger is complex.
2.	B. LED fitted finger movements	[4] Alper Yilmaz, Omar Javed, Mubarak Shah, "Object Tracking: A Survey", ACM Computer Survey. Vol. 38, Issue. 4, Article 13, Pp. 1-45, 2006	International Research Journal of Engineering and Technology (IRJET)	A method in which an LED is mounted on the user's finger, and the web camera is used to track the finger. The character drawn is compared with that present in the database	It returns the alphabet that matches the pattern drawn. It requires a red-coloured LED pointed light source is attached to the finger. Also, it is assumed that there is no red-coloured object other than the LED light within the web camera's focus.	A device is being used to determine the pattern. Will not recognize if multiple red lights are being gestured.
3.	C. Augmented Desk Interface	[5] Yuan-Hsiang Chang, Chen-Ming Chang, "Automatic Hand-Pose Trajectory Tracking System Using Video Sequences", INTECH, pp. 132- 152, Croatia, 2010	International Research Journal of Engineering and Technology (IRJET)	Augmented segmented desk interface approach for interaction was proposed. This system makes use and a video projector and charge-coupled device (CCD) camera so that using the fingertip; users can operate desktop applications.	In this system, each hand performs different tasks. The left hand is used to select radial menus, whereas the right hand is used for selecting objects to be manipulated. It achieves this by using an infrared camera.	Determining the fingertip is computationally expensive, so this system defines search windows for fingertips.

3. METHODOLOGY

MEDIAPIPE:

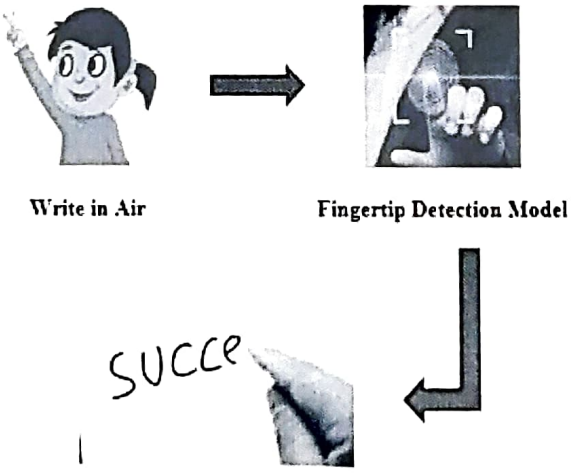
Mediapipe is a cross-platform library developed by Google that provides amazing ready-to-use ML solutions for computer vision tasks. OpenCV library in python is a computer vision library that is widely used for image analysis, image processing, detection, recognition, etc.

ML PIPELINE:

MediaPipe Hands utilizes an ML pipeline consisting of multiple models working together. ML pipeline is a means of automating the machine learning workflow by enabling data to be transformed and correlated into a model that can then be analyzed to achieve outputs. This type of ML pipeline makes the process of inputting data into the ML model fully automated

To detect initial hand locations, we designed a single-shot detector model optimized for real-time uses in a manner similar to the face detection model in MediaPipe Face Mesh.

After the palm detection over the whole image our subsequent hand landmark model performs precise keypoint localization of 21 3D hand-knuckle coordinates inside the detected hand regions via regression, that is direct coordinate prediction. The model learns a consistent internal hand pose representation and is robust even to partially visible hands and self-occlusions.



Write in Air **Fingertip Detection Model**

Traced Trajectory

4. IMPLEMENTATION

```
from tkinter import Frame
import cvzone.HandTrackingModule
from cvzone.HandTrackingModule import HandDetector
import cv2
from collections import deque
from matplotlib.pyplot import draw
import numpy as np


length=0


def rescale_frame(frame, scale):    # works for image, video, live video
    width = int(frame.shape[1] * scale)
    height = int(frame.shape[0] * scale)
    dimensions = (width, height)
    return cv2.resize(frame, dimensions, interpolation=cv2.INTER_AREA)

cap = cv2.VideoCapture(0)
cap.set(3, 1440)



detector = HandDetector(detectionCon=0.85, maxHands=2)


# Giving different arrays to handle colour points of different colour
bpoints = [deque(maxlen=1024)]
gpoints = [deque(maxlen=1024)]
rpoints = [deque(maxlen=1024)]
ypoints = [deque(maxlen=1024)]
whitepoints = [deque(maxlen=1024)]

# These indexes will be used to mark the points in particular arrays of specific colour
blue_index = 0
green_index = 0
red_index = 0
yellow_index = 0
white_index = 0

#The kernel to be used for dilation purpose
kernel = np.ones((5,5),np.uint8)

colors = [(255, 0, 0), (0, 255, 0), (0, 0, 255), (0, 255, 255),(255,255,255)]
colorIndex = 0

# Here is code for Canvas setup
paintWindow = np.zeros((660,860,3)) + 255
paintWindow = cv2.rectangle(paintWindow, (40,1), (140,65), (0,0,0), 2)
paintWindow = cv2.rectangle(paintWindow, (160,1), (255,65), colors[0], -1)
paintWindow = cv2.rectangle(paintWindow, (275,1), (370,65), colors[1], -1)
paintWindow = cv2.rectangle(paintWindow, (390,1), (485,65), colors[2], -1)
```

```
paintWindow = cv2.rectangle(paintWindow, (505,1), (600,65), colors[3], -1)


cv2.putText(paintWindow, "ERASE", (49, 33), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 0), 2, cv2.LINE_AA)
cv2.putText(paintWindow, "BLUE", (185, 33), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2, cv2.LINE_AA)
cv2.putText(paintWindow, "GREEN", (298, 33), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2, cv2.LINE_AA)
cv2.putText(paintWindow, "RED", (420, 33), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2, cv2.LINE_AA)
cv2.putText(paintWindow, "YELLOW", (520, 33), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (150,150,150), 2, cv2.LINE_AA)

cv2.namedWindow('Paint', cv2.WINDOW_AUTOSIZE)



xp,yp=0,0



while True:
    # Get image frame
    success, img = cap.read()

    img= rescale_frame(img, scale=.8)


    img=cv2.flip(img,1)



    # Adding the colour buttons to the live frame for colour access
    img = cv2.rectangle(img, (40,1), (140,65), (122,122,122), -1)
    img = cv2.rectangle(img, (160,1), (255,65), colors[0], -1)
    img = cv2.rectangle(img, (275,1), (370,65), colors[1], -1)
    img = cv2.rectangle(img, (390,1), (485,65), colors[2], -1)
    img = cv2.rectangle(img, (505,1), (600,65), colors[3], -1)
    cv2.putText(img, "ERASE", (49, 33), cv2.FONT_HERSHEY_SIMPLEX, 0.56, (0, 0, 0), 1, cv2.LINE_AA)
    cv2.putText(img, "BLUE", (185, 33), cv2.FONT_HERSHEY_SIMPLEX, 0.56, (0, 0,0), 1, cv2.LINE_AA)
    cv2.putText(img, "GREEN", (298, 33), cv2.FONT_HERSHEY_SIMPLEX, 0.56, (0, 0, 0), 1, cv2.LINE_AA)
    cv2.putText(img, "RED", (420, 33), cv2.FONT_HERSHEY_SIMPLEX, 0.56, (0, 0, 0), 1, cv2.LINE_AA)
    cv2.putText(img, "YELLOW", (520, 33), cv2.FONT_HERSHEY_SIMPLEX, 0.56, (0, 0, 0), 1, cv2.LINE_AA)

    # Find the hand and its landmarks
    lmListL=[]
    lmListR=[]
    infols=[]

    #img = detector.findHands(img)  # with draw
    hands,img = detector.findHands(img ,draw=1)  # without draw

    if len(hands)!= 0 :
        hand1 = hands[0]

        lmListL= hand1["lmList"]
        for i in range(len(lmListL[8])):

            x1=lmListL[8][0]
            y1=lmListL[8][1]
            #print(x1,y1)
```

```
for i in range(len(lmListL[12])):
    x2=lmListL[12][0]
    y2=lmListL[12][1]
    #print(x2,y2)
ll=len(lmListL)-1
print(hands)
fingers2L = detector.fingersUp(hand1)
print(fingers2L)
lengthL, info, _ = detector.findDistance(lmListL[4], lmListL[8],img|False)

#cv2.line(img,lmListL[4],lmListL[8],(0,200,0),4)
#if (xp==0 and yp == 0):
  # xp,yp=x1,y1
if (lengthL ) <= 35:
    cv2.circle(img, (lmListL[4]), 15, (0, 0, 0), -1)
    if (xp==0 and yp == 0):
        xp,yp=x1,y1
    cv2.line(paintWindow,(xp,yp),(x1,y1),colors[colorIndex],10)
    xp,yp=x1,y1
else:
    xp,yp=0,0
#if (fingers2L[0] and fingers2L[1]):


#selection mode
if (fingers2L[1] and fingers2L[2]):
    xp,yp=0,0
    cv2.rectangle(img,(x1,y1+15),(x2,y2-10),colors[colorIndex],-1)

    if y1 <= 65:
        if 40 <= x1 <= 140: # Clear Button
            bpoints = [deque(maxlen=512)]
            gpoints = [deque(maxlen=512)]
            rpoints = [deque(maxlen=512)]
            ypoints = [deque(maxlen=512)]
            whitepoints = [deque(maxlen=512)]

            blue_index = 0
            green_index = 0
            red_index = 0
            yellow_index = 0
            white_index = 0

            img[67:,:,:] = 255
            colorIndex = 4 # white
        elif 160 <= x1 <= 255:
                colorIndex = 0 # Blue
        elif 275 <= x1 <= 370:
                colorIndex = 1 # Green
        elif 390 <= x1<= 485:
                colorIndex = 2 # Red
        elif 505 <= x1<= 600:
                colorIndex = 3 # Yellow
    else :
        if colorIndex == 0:
            bpoints[blue_index].appendleft(y1)
        elif colorIndex == 1:
```

```
            gpoints[green_index].appendleft(y1)
        elif colorIndex == 2:

            rpoints[red_index].appendleft(y1)
        elif colorIndex == 3:
            ypoints[yellow_index].appendleft(y1)
        elif colorIndex == 4:
            whitepoints[white_index].appendleft(y1)
    # Append the next deques when nothing is detected to avois messing up
    else:
        bpoints.append(deque(maxlen=512))
        blue_index += 1
        gpoints.append(deque(maxlen=512))
        green_index += 1
        rpoints.append(deque(maxlen=512))
        red_index += 1
        ypoints.append(deque(maxlen=512))
        yellow_index += 1
        whitepoints.append(deque(maxlen=512))
        white_index += 1



if len(hands) == 2:
    hand2 = hands[1]
    lmListR= hand2["lmList"]
    fingers2R = detector.fingersUp(hand2)
    lengthR, info, _ = detector.findDistance(lmListR[4], lmListR[8],img|False)
    #cv2.line(img,lmListR[4],lmListR[8],(0,200,0),4)
    if (lengthR)<35:
        cv2.circle(img, (lmListR[4]), 15, (0,0,0),-1)

# Show all the windows
#Hori = np.concatenate((paintWindow,img),axis=0)
#imnn=cv2.resize(img,(960,540))
#imnp=cv2.resize(paintWindow,(960,540))
#numpy_vertical = np.vstack((imnn, imnp))
#resize

##cv2.imshow('App',numpy_vertical)

cv2.imshow("Paint",paintWindow)
# Display
cv2.imshow("Image", img)
if cv2.waitKey(1) & 0xFF==ord('q'):
    break
cap.release()
cv2.destroyAllWindows()
```

5. RESULT DISCUSSION

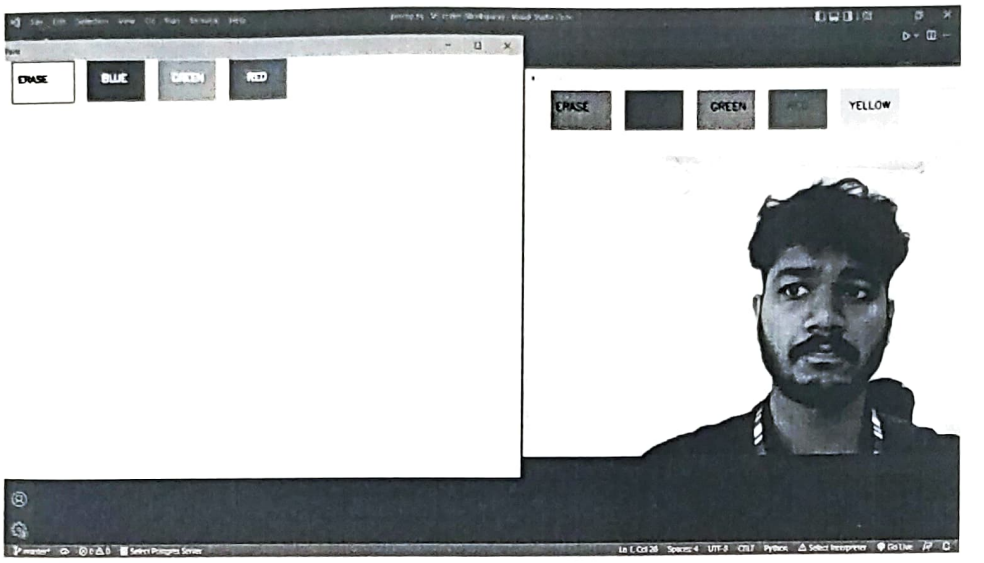


Fig: 1.1

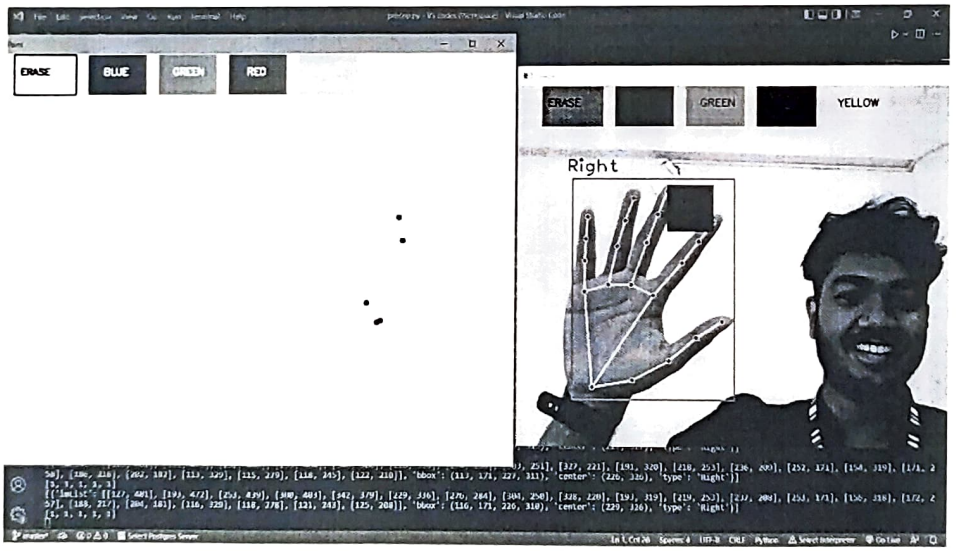


Fig: 1.2

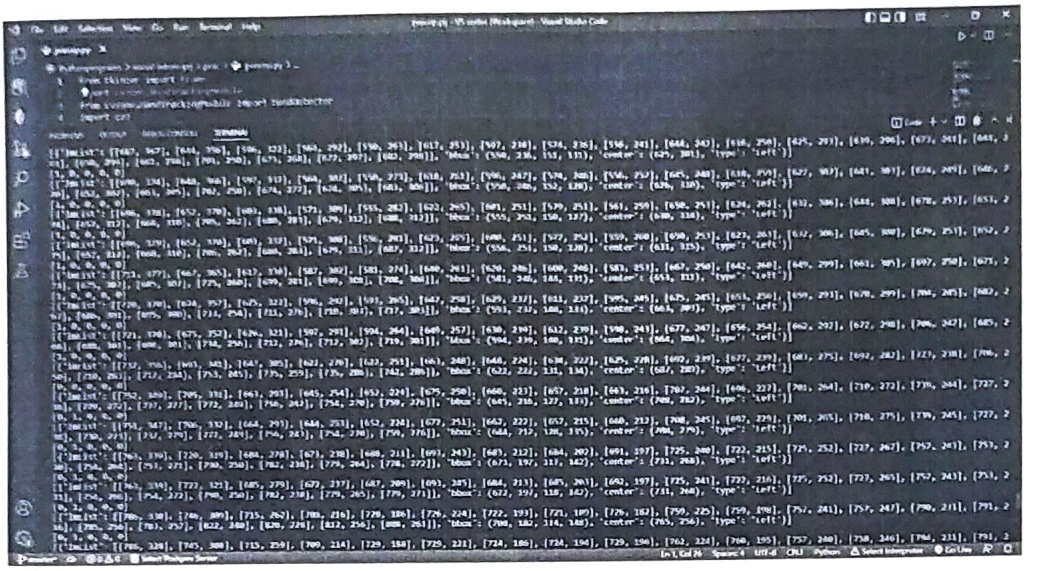


Fig: 1.3

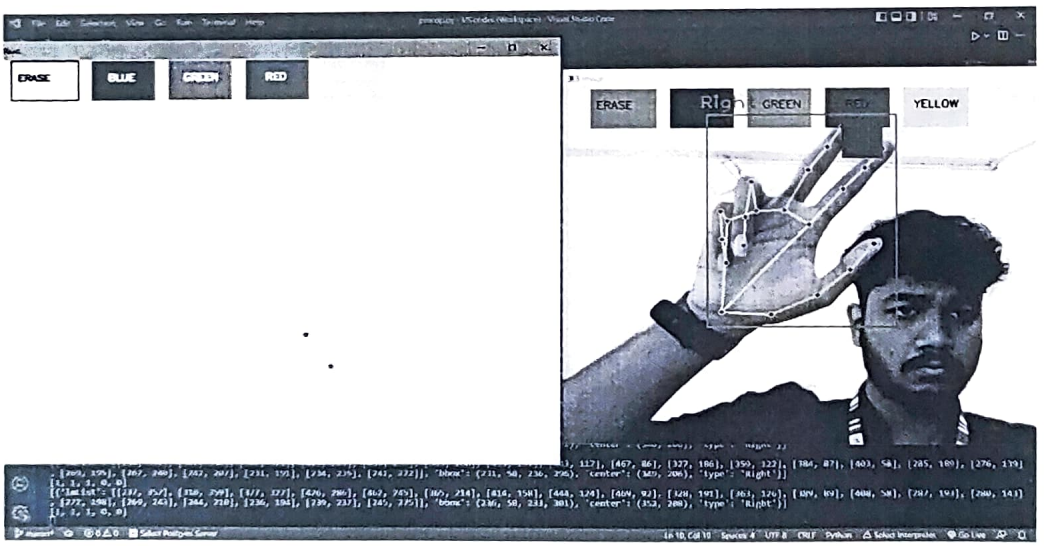


Fig: 1.4

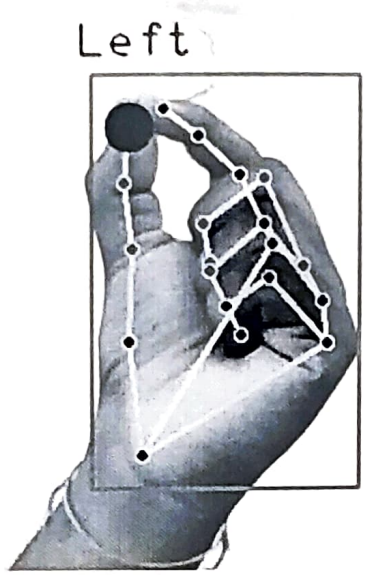


Fig: 1.5

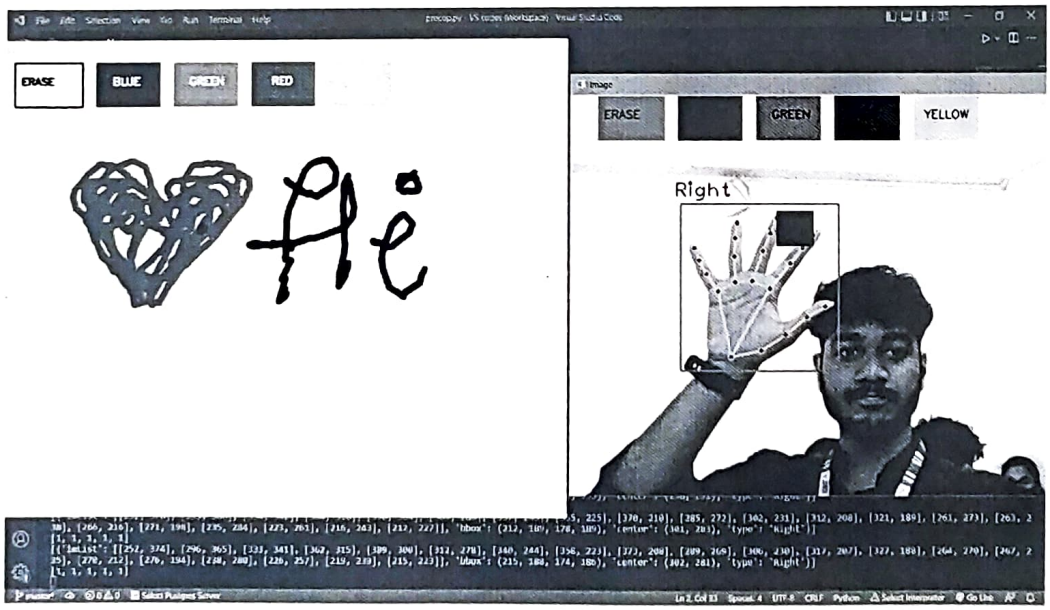


Fig: 1.6

6. CONCLUSION AND FUTURE WORK

The system has the potential to challenge traditional writing methods. It can reduce the use of a stylus and improve the learning experience. Extending the functionality, the system can also be used to control IoT devices shortly. Drawing in the air made possible through projection systems. The system will be an excellent software using which people could better interact with the digital world. Augmented Reality can make text come alive.

There are some limitations of the system which can be improved in the future. Firstly, using a handwriting recognizer in place of a character recognizer will allow the user to write word by word, making writing faster. Secondly, hand gestures with a pause can be used to control the real-time system instead of using the number of fingertips. Thirdly, our system sometimes recognizes fingertips in the background and changes their state. Air-writing systems should only obey their master's control gestures and should not be misled by people around

7. SOCIAL CONTRIBUTION

It will also serve a great purpose in Fostering futuristic Learning, helping especially people to communicate easily. Even senior citizens or people who find it difficult
to use keyboards will able to use system effortlessly.
Bring content and conversations together in a single, portable experience that's always up to date—perfect for the new hybrid workplace/ Classrooms.Group interaction. Interactive writing promotes collaboration among students and group discussion and participation. They can be an effective tool for brainstorming due to the fact that drawing figures and saved to be shared and distributed to students later

8. REFERENCES

[1] Y. Huang, X. Liu, X. Zhang, and L. Jin, "A Pointing Gesture Based Egocentric Interaction System: Dataset, Approach, and Application," 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Las Vegas, NV, pp. 370-377, 2016.

[2] P. Ramasamy, G. Prabhu, and R. Srinivasan, "An economical air writing system is converting finger movements to text using a web camera," 2016 International Conference on Recent Trends in Information Technology (ICRTIT), Chennai, pp. 1-6, 2016.

[3] Saira Beg, M. Fahad Khan and Faisal Baig, "Text Writing in Air," Journal of Information Display Volume 14, Issue 4, 2013

[4] Alper Yilmaz, Omar Javed, Mubarak Shah, "Object Tracking: A Survey", ACM Computer Survey. Vol. 38, Issue. 4, Article 13, Pp. 1-45, 2006

[5] Yuan-Hsiang Chang, Chen-Ming Chang, "Automatic Hand-Pose Trajectory Tracking System Using Video Sequences", INTECH, pp. 132- 152, Croatia, 2010

[6] Erik B. Sudderth, Michael I. Mandel, William T. Freeman, Alan S. Willsky, "Visual Hand Tracking Using Nonparametric Belief Propagation", MIT Laboratory For Information & Decision Systems Technical Report P2603, Presented at IEEE CVPR Workshop On Generative Model-Based Vision, Pp. 1-9, 2004

[7] T. Grossman, R. Balakrishnan, G. Kurtenbach, G. Fitzmaurice, A. Khan, and B. Buxton, "Creating Principal 3D Curves with Digital Tape Drawing," Proc. Conf. Human Factors Computing Systems (CHI' 02), pp. 121- 128, 2002.

[8] T. A. C. Bragatto, G. I. S. Ruas, M. V. Lamar, "Real-time Video-Based Finger Spelling Recognition System Using Low Computational Complexity Artificial Neural Networks", IEEE ITS, pp. 393-397, 2006

[9] Yusuke Araga, Makoto Shirabayashi, Keishi Kaida, Hiroomi Hikawa, "Real Time Gesture Recognition System Using Posture Classifier and Jordan Recurrent Neural Network", IEEE World Congress on Computational Intelligence, Brisbane, Australia, 2012

[10] Ruiduo Yang, Sudeep Sarkar, "Coupled grouping and matching for sign and gesture recognition", Computer Vision and Image Understanding, Elsevier, 2008

[11] R. Wang, S. Paris, and J. Popovic, "6D hands: markerless hand-tracking for computer-aided design," in Proc. 24th Ann. ACM Symp. User Interface Softw. Technol., 2011, pp. 549–558.

[12] Maryam Khosravi Nahouji, "2D Finger Motion Tracking, Implementation For Android Based Smartphones", Master's Thesis, CHALMERS Applied Information Technology,2012, pp 1-48

[13] EshedOhn-Bar, Mohan ManubhaiTrivedi, "Hand Gesture Recognition In Real-Time For Automotive Interfaces," IEEE Transactions on Intelligent Transportation Systems, VOL. 15, NO. 6, December 2014, pp 2368-2377

[14] P. Ramasamy, G. Prabhu, and R. Srinivasan, "An economical air writing system is converting finger movements to text using a web camera," 2016 International Conference on Recent Trends in Information Technology (ICRTIT), Chennai, 2016, pp. 1- 6.

[15] Kenji Oka, Yoichi Sato, and Hideki Koike, "Real-Time Fingertip Tracking and Gesture Recognition," IEEE Computer Graphics and Applications, 2002, pp.64-71.

[16] H.M. Cooper, "Sign Language Recognition: Generalising to More Complex Corpora", Ph.D. Thesis, Centre for Vision, Speech and Signal Processing Faculty of Engineering and Physical Sciences, University of Surrey, UK, 2012

[17] Y. Huang, X. Liu, X. Zhang, and L. Jin, "A Pointing Gesture Based Egocentric Interaction System: Dataset, Approach, and Application," 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Las Vegas, NV, pp. 370-377, 2016

[18] Vladimir I. Pavlovic, Rajeev Sharma, and Thomas S. Huang, "Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review," IEEE Transactions on Pattern Analysis and Machine Intelligence, VOL. 19, NO. 7, JULY 1997, pp.677-695

[19] Guo-Zhen Wang, Yi-Pai Huang, Tian-Sheeran Chang, and Tsu-Han Chen, "Bare Finger 3D Air-Touch System Using an Embedded Optical Sensor Array for Mobile Displays", Journal Of Display Technology, VOL. 10, NO. 1, JANUARY 2014, pp.13-18

[20] Napa Sae-Bae, Kowsar Ahmed, Katherine Isbister, NasirMemon, "Biometric-rich gestures: a novel approach to authentication on multi-touch devices," Proc. SIGCHI Conference on Human Factors in Computing System,2005, pp.977-986

[21] W. Makela, "Working 3D Meshes and Particles with Finger Tips, towards an Immersive Artists' Interface," Proc. IEEE Virtual Reality Workshop, pp. 77-80, 2005.

[22] A.D. Gregory, S.A. Ehmann, and M.C. Lin, "inTouch: Interactive Multiresolution Modeling and 3D Painting with a Haptic Interface," Proc. IEEE Virtual Reality (VR' 02), pp. 45-52, 2000.

[23] W. C. Westerman, H. Lamiraux, and M. E. Dreisbach, "Swipe gestures for touch screen keyboards," Nov. 15 2011, US Patent 8,059,101

[24] S. Vikram, L. Li, and S. Russell, "Handwriting and gestures in the air, recognizing on the fly," in Proceedings of the CHI, vol. 13, 2013, pp. 1179–1184.

[25] X. Liu, Y. Huang, X. Zhang, and L. Jin. "Fingertip in the eye: A cascaded CNN pipeline for the real-time fingertip detection in egocentric videos," CoRR, abs/1511.02282, 2015.