A Minor Project Report

On

**VIRTUAL PAINTING**

Submitted in partial fulfillment of requirements for the award of the Degree of

**BACHELOR OF TECHNOLOGY**

in

**ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**

Under the guidance of

**Dr. N.M. SARAVANA KUMAR**

**(HEAD OF THE DEPARTMENT)**

**Submitted By**

|  |  |  |
| --- | --- | --- |
| NIKITHA | Y S | (927621BAD035) |
|  |  |  |
| NIVEDHA | M | (927621BAD036) |
| SHURUTHI | R S | (927621BAD048) |

|  |  |  |
| --- | --- | --- |
| SUBAA | R | (927621BAD052) |

**DEPARTMENT OF**

**ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**

**M.KUMARASAMY COLLEGE OF ENGINEERING**

**(**Autonomous)

KARUR – 639113

APRIL 2023

I

**M.KUMARASAMY COLLEGE OF ENGINEERING**

**VISION**

To emerge as a leader among the top institutions in the field of technical education.

**MISSION**

* Produce smart technocrats with empirical knowledge who can surmount the global challenges.
* Create a diverse, fully engaged, learner-centric campus environment to provide quality education to the students.
* Maintain mutually beneficial partnership with our alumni, industry and professional associations.

**DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**

**VISION**

To reach excellent standards of quality-education by using latest tools and to be the centre of excellence by promoting knowledge centric education, innovation and state-of-art research in the field of Artificial Intelligence and Data Science.

**MISSION**

M1: To impart quality and value based education and contribute towards the innovation of computing, expert system, Data Science to raise satisfaction level of all stakeholders.

M2: To educate the future Computing engineers with strong fundamentals by continuously improving the teaching learning methodologies using contemporary aids.

M3: Enabling students to get expertise in critical skills with Artificial Intelligence domain and facilitate socially responsive research and innovation.

M4: To encourage professional development of students that will inculcate ethical values and leadership skills while working with the community to address societal issues

II

**PROGRAM EDUCATIONAL OBJECTIVE**

**PEO1:** To apply mathematical, scientific and engineering concepts essential for a data architect/data scientistto address the various challenges using emerging AI technologies.

**PEO2:** To impart knowledge to create, analyze, design, implementation and test a novel solution required for broader social context.

**PEO3**: To hone personality skills, leadership qualities, social commitment, social responsibilities, possess professional and ethical attitude through life-long learning and multidisciplinary approach.

**PROGRAM SPECIFIC OUTCOMES**

**I. Professional Skills**: Comprehend the technological advancements and practice professional ethics and the concerns for societal and environmental well-being.

**II. Successful career:** Apply knowledge of theoretical computer science to assess the hardware and software aspects of computer systems.

**PROGRAM OUTCOMES**

**PO1:** Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

**PO2:** Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

**PO3**: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

III

**PO4:** Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

**PO5:** Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

**PO6:** The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

**PO7:** Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

**PO8:** Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

**PO 9:** Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

**PO10:** Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

**PO11:** Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

**PO12:** Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

IV

**COURSE OUTCOME**

**CO1:** Identify the problem by applying acquired knowledge.

**CO2:** Analyze and categorize executable project modules after considering risks.

**CO3:** Choose efficient tools for designing project modules.

**CO4:** Combine all the modules through effective team work after efficient testing.

**CO5:** Elaborate the completed task and compile the project report

VI

**M.KUMARASAMY COLLEGE OF ENGINEERING**

**(Autonomous Institution affiliated to Anna University, Chennai)**

**BONAFIDE CERTIFICATE**

Certified that this project report **“VIRTUAL PAINTING”** is the bonfire

work of **“NIKITHA Y S (927621BAD035), NIVEDHA M**

**(927621BAD036), SHURUTHI R S (927621BAD048), SUBAA R(927621BAD052)”** who carried out the minor project work during the academic year 2023 under our supervision. Certified further, that to the best of our knowledge the work reported herein does not form partof any other minor project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

|  |  |
| --- | --- |
| **Signature** | **Signature** |
| DR.N.M. SARAVANA KUMAR | Dr. N.M. SARAVANA KUMAR |
| **HEAD OF THE DEPARTMENT,** | **HEAD OF THE DEPARTMENT,** |
| Department of Artificial Intelligence and  Data Science, | Department of Artificial Intelligence and Data Science, |
| M.Kumarasamy College of Engineering, | M.Kumarasamy College of Engineering, |
| Thalavapalayam, Karur-639113. | Thalavapalayam, Karur-639113. |

VII

**ABSTRACT**

Virtual AI painter using Open CV and Media pipe is an application that tracks the movement of an object. Using this tracking feature the user can draw on the screen by moving the object (which in our project is the human hand) in the air, in front of the webcam.OpenCV (Open Source Computer Vision) - is a programming language library consisting of different types of functions mainly for computer vision.

VIII

**Table of Contents**

**ChapterNo** **Particulars**

**ABSTRACT**

**LIST OF FIGURES**

**1** **INTRODUCTION**

1.1 General

1.2 Objectives of the research

|  |  |
| --- | --- |
| **2** | **LITERATURE REVIEW** |
| **3** | **EXISTING SYSTEM** |
| **4** | **PROBLEM STATEMENT** |
| **5** | **PROPOSED SYSTEM** |
| **6** | **RESULT AND DISCUSSION** |
| **7** | **CONCLUSION AND FUTURE SCOPE** |
|  | **REFERENCE** |
|  | **APPENDIX** |

**Page No**

1

2

3

5

6

7

10

12

14

18

IX

**List of Figures**

**Figure No** **Figure Name** **Page No**

|  |  |  |
| --- | --- | --- |
| 3.1 | Hardware implementation | 5 |
| 3.2 | System architecture | 7 |
| 3.3 | Block diagram | 8 |
| 4.0 | Modules | 9 |

X

**Acronyms/List of Abbreviations**

**Acronym**

**Abbreviations**

ASIC

Applicatiosn-Specific Integrated Circuit

CCD

Charge-Coupled Device

CFA

Color Filter Array

FIFO

FPGA

FPS

HAL

HDL

IP

LE

MMU

PLL

RISC

SDRAM

SRAM

VGA

First-In First-Out

Field-Programmable Gate Array

Frames Per Second

Hardware Abstraction Layer

Hardware Descriptive Language

Intellectual Property

Logic Element

Memory Management Unit

Phase-Locked Loop

Reduced Instruction-Set Computing

Synchronous Dynamic Random Access Memory

Static Random Access Memory

Video Graphics Array

XI

**INTRODUCTION**

**CHAPTER 1**

**INTRODUCTION**

**1.1 GENERAL**

Virtual AI painter using OpenCV and Mediapipe is an application that tracks the movement of an object. Using this tracking feature the user can draw on the screen by moving the object (which in our project is the human hand) in the air, in front of the webcam. This real time webcam data generated by tracking the movement of the object helps the user to draw simple things which are both interesting and challenging. OpenCV ( OpenCV (Open Source Computer Vision) - is a programming language library consisting of different types of functions mainly for computer vision. To explain in a simple language or in general way it is a library used for Image Processing. It is used mainly to do all the operations which are related to Images.

1

**1.2 OBJECTIVES OF THE RESEARCH**

OpenCV is a library for images. It roughly supports all main programming languages. Commonly used in python and C++. OpenCv can be used to read or write an image, for image modification. Convert coloured to gray, binary etc. OPENCV is also an OPEN SOURCE.

Some of it’s notable applications are as follows:

1. Face detection
2. Multi-hand tracking
3. Detecting and tracking an object.
4. AutoFlip - pipeline to crop videos automatically.

2

**LITERATURE REVIEW**

**CHAPTER 2**

**LITERATURE REVIEW**

Many methods are used for hand gesture recognition in real-time. Sayem Mohammad Siam, Jahidul Adnan Sakel, and Md. Hasanul Kabir has proposed a new method of HCI (Human-Computer Interaction), that uses marker detection and tracking technique. Instead of having a mouse or touchpad, two colored markers are worn on the tips of the fingers to generate eight hand movements to provide instructions to a desktop or laptop computer with a consumer-grade camera . They have also used the "Template matching" algorithm for the detection of markers and Kalman Filter for tracking. In the developed system uses a data glove-based approach to recognize real-time dynamic hand gestures. The data glove has ten soft sensors integrated in it that measure the joint angles of five fingers and are used to collect gesture data. Real-time gestures are recognized using techniques such as gesture spotting, gesture sequence simplification, and gesture recognition.

3

Furthermore, Mohamed Nasir Bin Mohamed Shukor, Lo Hai Hiung and Patrick

Sebastian entitled “Implementation of Colour Filtering on FPGA” proposed presents the construction of a real time hardware image processing system on Field Programmable Gate Array (FPGA). The chosen image processing algorithm is a single colour Filtering algorithm. This work utilizes Altera DE2 development board empowered by the Cyclone II FPGA paired with a 1.3 Mega pixel CMOS camera from Terasic Technologies. Verilog HDL is chosen as the hardware programming language for this system and its compiled using Quartus

1. program. The functionality of the algorithm is first verified in Matlab, simulating the expected output of the system before implementing it onto the FPGA development board. colour Filtering algorithms are successfully implemented on Cyclone II FPGA. The double band-pass filter algorithm is

found to be more effective to capture a wider spectrum of blue colour compared to that of a single band-pass filter algorithm. Currently work is done to quantify a performance metric for the system before implementing and testing the triple and quadruple band-pass filter algorithm to determine if there were further improvement could be achieved.

4

**EXISTING SYSTEM**

**CHAPTER 3**

**EXISTING SYSTEM**

**Writing Hand Pose Detection**

Recognizing the position of the composing hand and recognizing it through other signals is an fundamental step in initializing airborne composing. Not at all like conventional writing, when the write moves down, and the write moves up, composing within the discuss isn't laid out as a writing arrangement. Events. The framework recognizes the position of a piece hand and recognizes it from a non-writing hand by tallying the number of raised fingers.

5

**PROBLEM STATEMENT**

**CHAPTER 4**

**PROBLEM STATEMENT**

Object recognition and tracking with real-time smart camera plays an important role in many modern vision applications such as work force tracking, intelligence surveillance, fire detection and many more. This research will be present a FPGA-based digital control for a Virtual Paint. Virtual Paint project use DE2 board, 5 megapixel 3 camera and a VGA monitor, which is uses the camera to capture hand movements, so you can hand in the air painting and displayed on a monitor, the whole process without using mouse or joystick like device. The main motivation of this project is to extract the gesture detection and color segmentation technique from CMOS camera sensor to perform virtual paint. The control algorithm will be using a Verilog language based on the use of logical state diagram.

6

**PROPOSED SYSTEM**

**CHAPTER 5**

**5.1 PROPOSED SYSTEM**

In the Proposed System we have utilized the advancements, for example,open cv and media pipe. Hand motion acknowledgment can likewise be utilized for applications like modern robot control, communication through signing interpretation, in the restoration gadget for individuals with furthest point actual debilitation and so on. Hand motion acknowledgment finds applications in shifted spaces including virtual conditions, brilliant observation, gesture based communication interpretation, clinical frameworks and so on.

7

**5.2 MODULES**

The modules are

1.Full compilation success

2.Camera not working

3.Remove i Exposure function

4.Pin assignment for camera

5.Image for hidden camera

6.Random dot colour appear on monitor

**5.3 MODULE**

1.FULL COMPILATION SUCCESS

Even successfully compiled and downloaded to the board, monitor running normally, but the camera still does not work.

**2.CAMERA NOT WORKING**

Replacement of the CCD control file, CCD\_Capture.v, I2C\_CCD\_Config.v, I2C\_Controller.v and Reset\_Delay.v in submodule files has been made and CCD camera module still does not work. Recompile the compiler get an error because the main function module refers to original I2C\_Controller.v to define a function.

**3.REMOVE IEXPOSURE FUNCTION**

Downloaded to the board running well, but still no success for camera function. Change on pin assignment for 1.5 megapixel camera pin out have been made and the compilation is successful .

8

**4. PIN ASSIGNMENT FOR CAMERA**

CCD\_DATA bit wide for 1.5 megapixel camera is 12 bits. Modifications and

compilation successfully done. Download and run program, camera still cannot correctly read images, but the images rich in color than before.

**5. IMAGE FOR HIDDEN CAMERA**

If any movements detect over the lens, there will be on the screen moving

images. Figure 4.6 shows an image with hidden camera which image becomes dark.

**6. RANDOM DOT COLOUR APPEAR ON MONITOR**

After programming, the white screen appeared a few scattered dots but the longer run the more points occurs on the screen show that it is still not possible to control brush function. At this point, the image points increased significantly which is Red, Green and Blue dots appear randomly based on three colors corresponding palette.

9

**RESULT AND DISCUSSION**

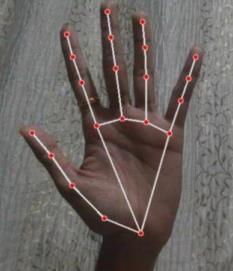
**CHAPTER 6**

**RESULT AND DISCUSSION**

The project has been created based on Verilog language which is run using the

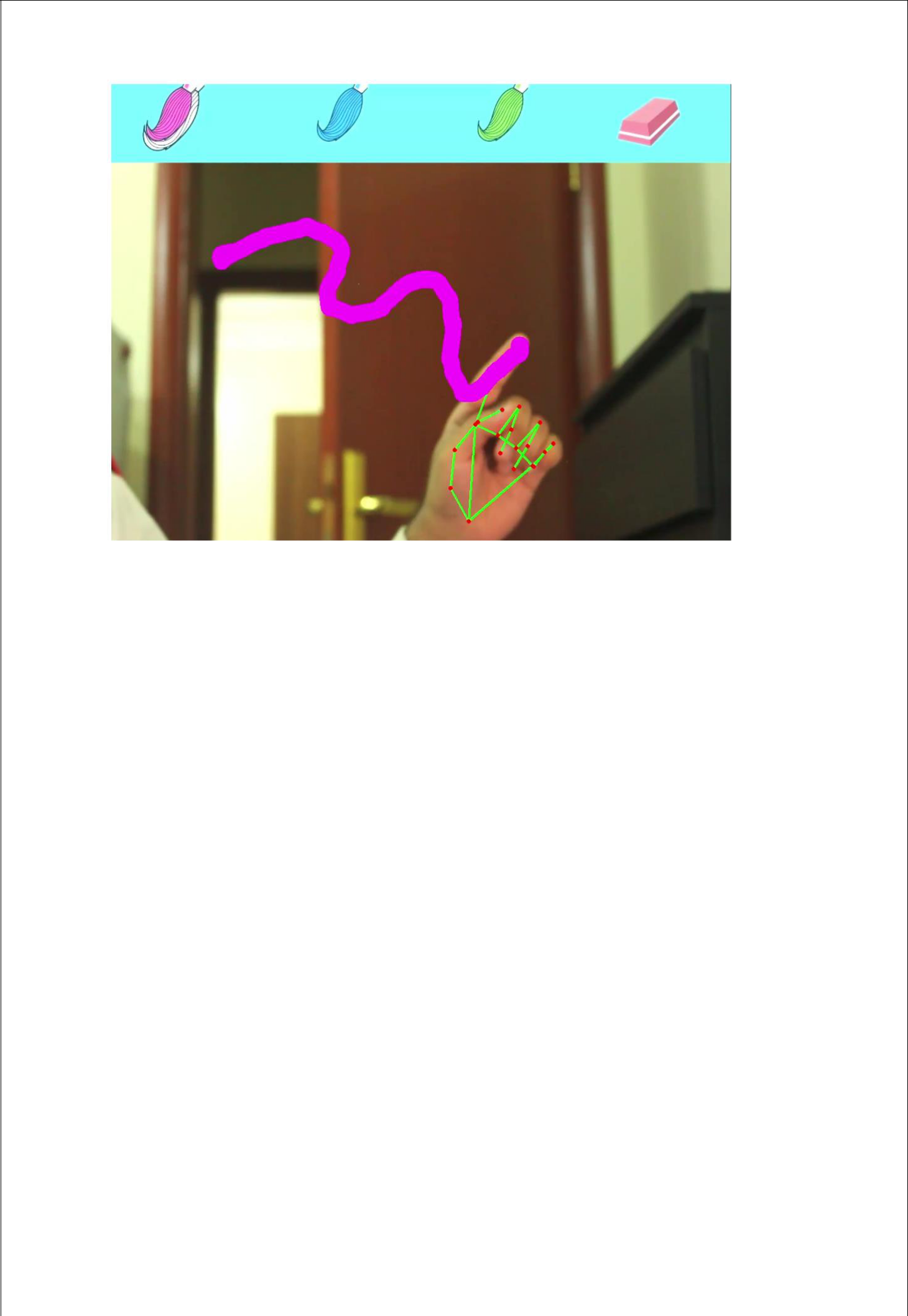
Quartus 8.1 software. Collected information provided on the website and research paper about procedures and software operation is carefully analyzed. Based on collecting information, the main challenge is to set up cursor movement that not response to the specified target color. With a different interface for 1.5 megapixel camera properties, reallocation of the pin assignment for 1.5 megapixel camera have been made.

**SCREENSHOTS**



**6.1 SCREENSHOT OF DETECTION**

10



**6.2 SCREENSHOT OF PAINTING**

11

**CONCLUSION**

**AND**

**FUTURE SCOPE**

**CHAPTER 7**

**CONCLUSION**

This application is capable of tough traditional writing strategies. It gets rid of the want to hold a cell telephone in hand to take notes, offering a easy manner to do the identical. The last intention is to create a computer imaginative and prescient device studying application that promotes human-computer interaction (HCI) moreover referred to as Human-laptop interaction (MMI)] which refers to the connection between humans and the PC or particularly the gadget. This undertaking allows the client to have interactive surroundings in which the customer can draw some thing he goals by the usage of choosing the preferred colors from the displayed collaboration.

12

**FUTURE SCOPE**

This work can be further improved by experimenting with different interpolation methods such as PyGame which includes a line drawing method that could help produce smoother and cleaner lines. In the same vein, a variety of brush shapes and textures can be implemented to make this application more robust.

13

**REFERENCES**

**REFERENCES**

1. M. Petouris, A. Kalantzopoulos and E. Zigouris (2005), “An FPGA-based Digital Camera System Controlled from an LCD Touch Panel”, Electronics Laboratory, Electronics and Computers Div., Department of Physics, University of Patras, GR-265 00 Patras, GREECE.

1. Suh Ho Lee, Seon Wook Kim and Suki Kim (2004), “Implementation Of A Low Power Motion Detection Camera Processor Using A Cmos Image Sensor”,

ISCAS 2004, Department of Electronics Engineering, Korea University, Seoul,

Korea.

3. Mohamed Nasir Shukor, Lo H. Liong and P. Sebastian (2007), “Implementation of Color Filtering on FPGA”, Electrical and Electronic Engineering **,** Universitiy Teknologi PETRONAS, International Conference on Intelligent and Advanced Systems.

4. Nai-Jian Wang, Sheng-Chieh Chang, Pei-Jung Chou (2012), “A Real-time Multiface Detection System, Implemented on FPGA” *IEEE International* Symposium on Intelligent Signal Processing and Communication Systems *(ISPACS).*

5. C. T. Johnston, K.T. Gribbon, and D. G. Bailey (2004), "Implementing Image

Processing Algorithms on FPGAs", Proceedings of the Eleventh ElectronicsNew Zealand Conference, ENZCon’04.

1. A.Downton and D. Crookes (1998), "Parallel architectures for image processing," Electronics& Communication Engineering Journal , 1998.
2. Zoltan Pele, Tatj ana Samardzic, Neboj sa Maluckov and Nikola Teslic

14

(2003),"One Application of FPGA Integrated Circuits CMOS Camera and LCD

Display Controller Design", 6th International Conference on 42

Telecommunications in Modem Satellite, Cable and Broadcasting Services,

TELSIKS 2003, Nis, Serbia & Modenegro.

1. G. L. Foresti (1999), “Object recognition and tracking for remote video surveillance,” Circuits and Systems for Video Technology, IEEE Transactions on.
2. G. L. Foresti, C. S. Regazzoni (1994), “A change detection method for multiple object localization in real scenes”, Industrial Electronics, Control and

Instrumentation, IECON '94, 20 th International Conference on.

1. Jacky Baltes (2010), "Efficient Image Processing for Increased Resolution and Color Correctness of CMOS Image Sensors", 5th Robot World Cup Soccer

Games and Conferences.

1. Yakun Wu, Xiaohui Duan, Zhouhui Lian, Tiantian Shen, Shihu Zhao and Lei Van (2008), "The Design of Storage System for Digital Airborne Camera Based SOPC", Internation al Congress on Image and Signal Processing, CISP 2008, Sanya, Hainan, China .
2. S.Y. Chien, S. Y. Ma, and L. G. Chen (2002), “Efficient moving object

segmentation algorithm using background registration technique, ” *IEEE* Transactions on Circuits and Systems for Video Technology, Vol. 12, NO. 7,pp*.* 577-586.

1. Y. C. Chung, J. M. Wang, and S. W. Chen (2002), “Progressive Background

15

Image Generation, ” in Proc. of 15th IPPR Conf. on Computer Vision, Graphics

and Image Processing, pp. 858-865.

1. L. Maddalena and A. Petrosino (2008), “A Self-Organizing Approach to Background Subtraction for Visual Surveillance Applications,” *IEEE*

Transactions on Image Processing, Vol. 17, NO. 7, pp. 1168 - 1177.43

1. S. S. Ghidary, Y. Nakata, T. Takamori, and M. Hattori (2000), “Human Detection and Localization at Indoor Environment by Home Robot,” *in Proc. of*

IEEE International Conference of Systems, Man, and Cybernetics, Vol. 2,

pp.1360–1365.

16. K. K. Ken, S. H. Cho, H. J. Kim, and J. Y. Lee (2005), “Detecting and tracking moving object using an active camera, ” *in Proc. of 7th International*

Conference of Advanced Communication Technology, ICACT , Vol. 2,pp. 817–

820.

17. D. Murray and A. Basu (1994), “Motion tracking with an active camera”, IEEE Trans. On Pattem Analysis and Machine Intelligence, 16(5), pp. 449-459.

18. M. Irani, R. Roousso, and S. Peleg (1994), “Recovery of ego-motion using

image stabilization”, En Proc. Of the IEEE Computer Vision and Pattern

Recognition, pp. 454-460.

1. B. Jung and G. S. Sukhatme (2010), “Detecting moving objects using a single camera on a mobile robot in an outdoor environment”, In the 8’Conf. on

Intelligent Autonomous Systems,pp. 980-987.

20. Ronald J. Tocci, Neal S. Widmer (2001), “Digital Systems: Principles and

16

Application”, 8th Edition, Prentice Hall.

21. Rafikha Aliana A. Raof, Norina Idris, Phaklen Eh Kan, Mohammad Nazri Md. Noor (2006), “Digital Electronics Design”, Prentice Hall.

1. Altera (2006), "DE2 Education and Development Board User Manual",

Version 1.4.

1. Terasic (2007), "TRDB-LTM 4.3 Inch Digital Touch Panel Development

*Kit".*

1. Altera (2008), "Creating Low Cost Intelligent Display Modules With an FPGA and Embedded Processor", White Paper.

17

**APPENDIX**

import cv2

import numpy as np

import os

import track\_hands as TH

class VideoCamera():

def \_\_init\_\_(self,overlay\_image=[],draw\_color =(255,200,100)): self.cap = cv2.VideoCapture(0,cv2.CAP\_DSHOW) self.cap.set(3, 1280)

self.cap.set(4,720)

self.xp =0

self.yp =0

self.x1 = 0

self.y1 = 0

self.x2= 0

self.y2 = 0

self.brush\_thickness =15

self.eraser\_thickness =100

self.overlay\_image = overlay\_image

self.draw\_color = draw\_color

self.detector = TH.handDetector(min\_detection\_confidence=0.85) self.image\_canvas = np.zeros((720,1280,3), np.uint8) self.default\_overlay = overlay\_image[0]

def \_\_del\_\_(self):

self.cap.release()

18

def set\_overlay(self,frame, overlay\_image): self.default\_overlay = overlay\_image[0] frame[0:125,0:1280] = self.default\_overlay return frame

def get\_frame(self, overlay\_image):

frame = self.cap.read()

frame = cv2.flip(frame, 1)

frame[0:125,0:1280] = self.default\_overlay

frame = self.detector.findHands(frame, draw=True)

landmark\_list = self.detector.findPosition(frame, draw =False)

if(len(landmark\_list)!=0):

self.x1, self.y1 =(landmark\_list[8][1:]) #index self.x2, self.y2 = landmark\_list[12][1:] #middle

my\_fingers = self.detector.fingerStatus() if (my\_fingers[1]and my\_fingers[2]): self.xp, self.yp = 0,0 if (self.y1<125):

if(200<self.x1<340):

self.default\_overlay = overlay\_image[0]

frame[0:125,0:1280] = self.default\_overlay

self.draw\_color = (255,0,0)

elif (340<self.x1<500):

self.default\_overlay = overlay\_image[1]

19

self.draw\_color = (47,225,245) frame[0:125,0:1280] = self.default\_overlay elif (500<self.x1<640):

self.default\_overlay = overlay\_image[2] self.draw\_color = (197,47,245) frame[0:125,0:1280] = self.default\_overlay elif (640<self.x1<780):

self.default\_overlay = overlay\_image[3] self.draw\_color = (53,245,47) frame[0:125,0:1280] = self.default\_overlay elif (1100<self.x1<1280):

self.default\_overlay = overlay\_image[4]

self.draw\_color = (0,0,0)

frame[0:125,0:1280] = self.default\_overl

Ccv2.putText(frame, 'Color Selector Mode',

(900,680)fontFace=cv2.FONT\_HERSHEY\_COMPLEX, color=

(0,255,255), thickness=2, fontScale=1) cv2.line(frame, (self.x1,self.y1), (self.x2,self.y2), color=self.draw\_color, thickness

if (my\_fingers[1] and not my\_fingers[2]):

cv2.putText(frame, "Writing Mode", (900,680), fontFace=

cv2.FONT\_HERSHEY\_COMPLEX, color= (255,255,0), thickness=2, fontScale=1)

cv2.circle(frame, (self.x1,self.y1),15, self.draw\_color, thickness=-1

if self.xp ==0 and self.yp ==0:

self.xp =self.x1

self.yp =self.y1

if self.draw\_color == (0,0,0):

cv2.line(frame, (self.xp,self.yp),(self.x1,self.y1),color=

20

self.draw\_color,thickness=self.eraser\_thickness

cv2.line(self.image\_canvas, (self.xp,self.yp),(self.x1,self.y1),color=self.draw\_color, thickness=self.eraser\_thickness)

else:

cv2.line(frame, (self.xp,self.yp),(self.x1,self.y1),color= self.draw\_color, thickness=self.brush\_thickness)

cv2.line(self.image\_canvas, (self.xp,self.yp),(self.x1,self.y1),color= self.draw\_color, thickness=self.brush\_thickness)

self.xp , self.yp = self.x1, self.y1 frame[0:125,0:1280] = self.default\_overlay

img\_gray = cv2.cvtColor(self.image\_canvas, cv2.COLOR\_BGR2GRAY)

imginv= cv2.threshold(img\_gray, 50, 255, cv2.THRESH\_BINARY\_INV)

imginv = cv2.cvtColor(imginv, cv2.COLOR\_GRAY2BGR)

frame = cv2.bitwise\_and(frame, imginv)

frame =cv2.bitwise\_or(frame, self.image\_canvas)

\_,jpeg =cv2.imencode('.jpg', frame)

return jpeg.tobytes()

def maiN

overlay\_image=[]

header\_img = "Images"

header\_img\_list = os.listdir(header\_img)

for i in header\_img\_list:

image = cv2.imread(f'{header\_img}/{i}')

overlay\_image.append(image)

cam1 = VideoCamera(overlay\_image=overlay\_image)

21

while True:

ret, input\_img = cam1.cap.read()

input\_img = cv2.flip(input\_img,1)

my\_frame = cam1.get\_frame(frame=input\_img, overlay\_image=

overlay\_image)

cv2.imshow('out', my\_frame)

cv2.waitKey(1)

if \_\_name\_\_ =="\_\_main\_\_" :

main()

from flask import Flask, render\_template, Response

import cam

import os

import cv2

app = Flask(\_\_name\_\_,template\_folder='templates')

overlay\_image=[]

header\_img = "Images"

header\_img\_list = os.listdir(header\_img)

for i in header\_img\_list:

image = cv2.imread(f'{header\_img}/{i}')

overlay\_image.append(image)

@app.route('/')

def index():

return render\_template('index.html')

def gen():

22

cam1 = cam.VideoCamera(overlay\_image= overlay\_image)

while True:

frame = cam1.get\_frame(overlay\_image=overlay\_image)

yield (b'--frame\r\n'

b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n\r\n')

@app.route('/video\_feed')

def video\_feed():

return Response(gen(),

mimetype='multipart/x-mixed-replace; boundary=frame')

if \_\_name\_\_ == '\_\_main\_\_':

app.run(host='0.0.0.0', debug=False)

#192.168.0.105

23