A
Major Project
on

FINGER COUNTING AND VIRTUAL MOUSE USING CVLEARN

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

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

E.VAMSHI YADAV	**(187R1A0577)**
ANDREW DOMINIC FERNANDEZ	**(187R1A0578)**
P KEERTHANA	**(197R5A0505)**

Under the Guidance of

V NARESH KUMAR
(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC,NBA,Permanently Affiliated to JNTUH, Approved by AICTE, NewDelhi)

Recognized Under Section 2(f) & 12(B) of the UGCAct.1956,

Kandlakoya (V), Medchal Road, Hyderabad-501401.

2018-2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled **"FINGER COUNTING AND VIRTUAL MOUSE USING CVLEARN"** being submitted by **E VAMSHI YADAV(187R1A0577), ANDREW DOMINIC FERNANDEZ(187R1A0578), P KEERTHANA(197R5A0505)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by our team under our guidance and supervision during the year 2021-22.

The results embodied in this project have not been submitted to any other University or Institute for the award of any degree or diploma.

V NARESH KUMAR	**DR.A.RAJI REDDY**
(Assistant Professor)	**DIRECTOR**
INTERNAL GUIDE	

DR.K.SRUJAN RAJU	**EXTERNAL EXAMINER**
HOD	

Submitted for viva voice Examination held on ____________________

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to my guide **Mr V.Naresh Kumar** , Assistant Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark. We also take this opportunity to express a deep sense of gratitude to Project Review Committee (PRC) **Mr. A. Uday Kiran**, **Mr. J. Narsimharao, Dr. T. S. Mastan Rao, Mrs. G. Latha, Mr. A. Kiran Kumar**, for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srujan Raju**, Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully

We are obliged to **Dr. A. Raji Reddy**, Director for being cooperative throughout the course of this project. We also express our sincere gratitude to **Sri. Ch. Gopal Reddy**, Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

E VAMSHI YADAV	**(187R1A0577)**
ANDREW DOMINIC FERNANDEZ	**(187R1A0578)**
P KEERTHANA	**(197R1A0505)**

ABSTRACT

The finger tracking system is focused on user-data interaction, where the user interacts with virtual data, by handling through the fingers the volumetric of a 3D object that we want to represent. This system was born based on the human-computer interaction problem. The objective is to allow the communication between them and the use of gestures and hand movements to be more intuitive, Finger tracking systems have been created. These systems track in real time the position in 3D and 2D of the orientation of the fingers of each marker and use the intuitive hand movements and gestures to interact.

This project promotes an approach for the Human Computer Interaction (HCI) where cursor movement can be controlled using a real-time camera, it is an alternative to the current methods including manual input of buttons or changing the positions of a physical computer mouse. Instead, it utilizes a camera and computer vision technology to control various mouse events and is capable of performing every task that the physical computer mouse can. The Virtual Mouse color recognition program will constantly be acquiring real-time images where the images will be undergone a series of filtration and conversion. Whenever the process is complete, the program will apply the image processing technique to obtain the coordinates of the targeted colors position from the converted frames. After that, it will proceed to compare the existing colors within the frames with a list of color combinations, where different combinations consist of different mouse functions. If the current colors combination found a match, the program will execute the mouse function, which will be translated into an actual mouse function to the users' machine.

LIST OF FIGURES/TABLES

LIST OF SCREENSHOTS

TABLE OF CONTENTS

1. INTRODUCTION

1. INTRODUCTION

1.1 PROJECT SCOPE

This project is titled as "Hand tracking,finger tracking,virtual mouse using open cv” The main objective of the proposed virtual mouse system is to develop an alternative to the regular and traditional mouse system to perform and control the mouse functions, and this can be achieved with the help of a web camera that captures the hand gestures and hand tip and then processes these frames to perform the particular mouse function such as left click, right click, and scrolling function.

1.2 PROJECT PURPOSE

The main goal of the project is to manage computers and other devices with gestures other than pointing and clicking a mouse or touch display directly.This project avoids the physical requirement of mouse and The application has been designed to be cost effective and uses low cost input tools like webcam for capturing hand as input.

1.3 PROJECT FEATURES

The functions of mouse like controlling of movement of virtual object have been replaced by hand gestures.The main feature of the project is to identify the hand gestures and with help of them we can drag,drop,click and scroll the mouse This project is designed to operate with help of the webcam .It provides an efficient interface for a user to interact with the computer and access the various applications effortlessly.

2. SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

System analysis is the important phase in this process. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, "what must be done to solve the problem?" The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst has a firm understanding of what is to be done.

2.1 PROBLEM DEFINITION

The problems in this project are:To design motion tracking mouse which detect finger movements gestures instead of physical mouse.• To design an application (.py file) with user friendly user interface which provides feature for accessing motion tracking mouse feature. The camera should detect all the motions of hand and performs the operation of mouse.

2.2 EXISTING SYSTEM

There are three types of existing system:

1. Track ball.
2. Mechanical Mouse
3. Optical Mouse.

2.2.1 DISADVANTAGES OF EXISTING SYSTEM

1. Lasers are more accurate than optical
2. Slightly expensive

.

2.3 PROPOSED SYSTEM

As the technology increase everything becomes virtualised**.** Any application should either make human life more comfortable,more productive.Using the proposed system even-though there are a number of quick access methods available for the hand and mouse gesture for the laptops, using our project we could make use of the laptop and web-cam and by recognizing the hand gesture we could control mouse and perform basic operations like mouse pointer controlling, select and deselect using left click, and a quick access feature for file transfer between the systems.

2.3.1 ADVANTAGES OF PROPOSED SYSTEM

- Great flexibility then existing system.
- Easy to modify an adapt.
- Less prone to physical damaged due to absence of physical device.
- Avoid the mouse related wrist damage like **RSI &CTS**

2.4 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensurethat the proposed system is not a burden to the company. Three key considerations involved in the feasibility analysis are Economic,Technical and social Feasibility

2.4.1 ECONOMIC FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization.The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

2.4.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.4.3 BEHAVIOURAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently.The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system

2.5 HARDWARE & SOFTWARE REQUIREMENTS

HARDWARE REQUIREMENTS:

Hardware interfaces specifies the logical characteristics of each interface between the software product and the hardware components of the system. The following are hardware requirements.

System	: Windows 8 and above
RAM	: 8GB And higher
Hard Disk	: 50GB(Minimum)

SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system.The following are software requirements.

Operating System	:Windows 8 and above
Programming Language	: Python3
Tool	:Pycharm

Following are the Python Libraries Used :

- Import math
- Import Pyautogui
- Import cv2
- Import mediapipe as mp

2.6 MODULE DESCRPTION

- **USER MODULE:**

  User Module:In this user module the user faces the web camera and performs some hand gestures

- **SYSTEM MODULE:**

  System module: In this module it recognizes the gesture moments and perform several operations such as drag and drop,finger counting,right click and left clicks of a mouse operation.

3. ARCHITECTURE

3. ARCHITECTURE

3.1 PROJECT ARCHITECTURE

The project architecture given a brief description about how our system will function.We used built-in python libraries for the whole user-computer interaction.The user first sits in front of the camera and performs various gestures, the gestures are recorded by the camera and sends it to the system for processing.Then the results are displayed on the screen.

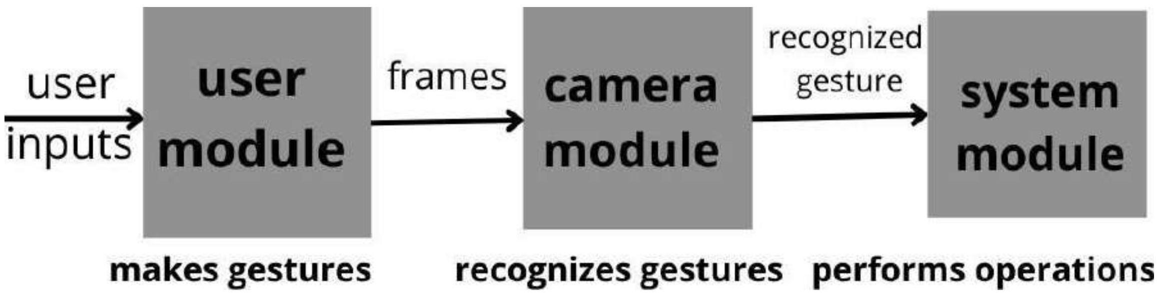


Figure 3.1 Project Architecture for Finger Counting and Virtual Mouse using CV Learn

3.2 USE CASE DIAGRAM

A use case diagram can summarize the details of our system's users (also known as actors) and their interactions with the system. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally.

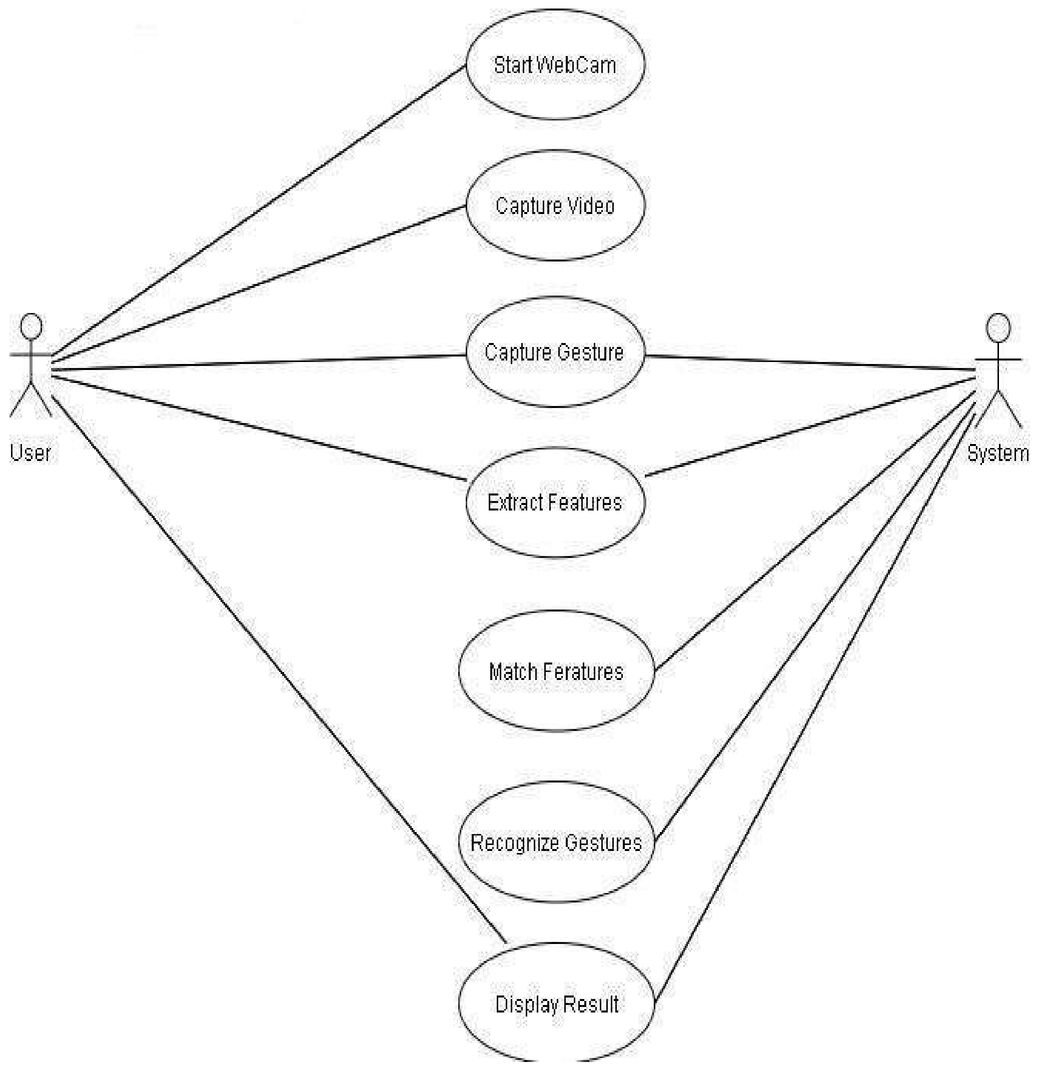


Figure 3.2 Use Case Diagram for Finger Counting and Virtual Mouse using CV Learn

3.3 CLASS DIAGRAM

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

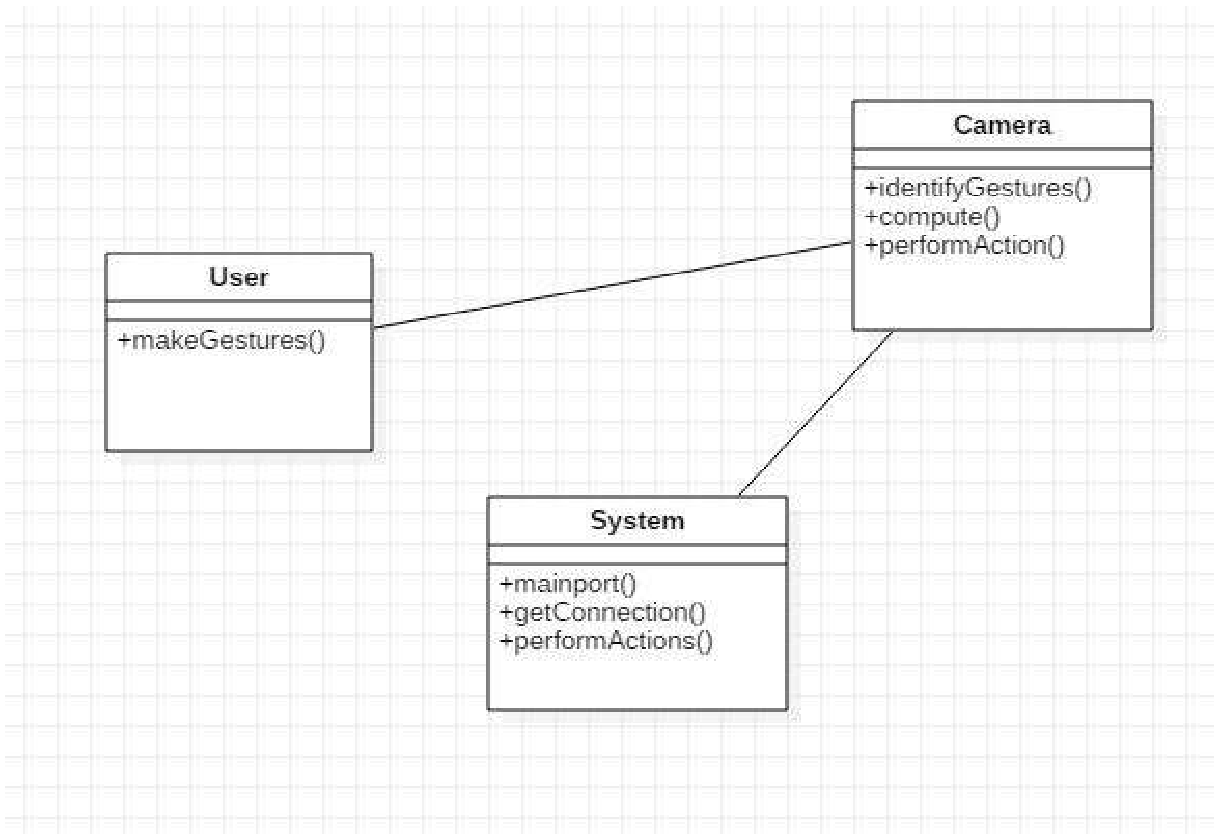


Figure 3.3 Class Diagram for Finger Counting and Virtual Mouse using CV Learn

3.4 SEQUENCE DIAGRAM

A sequence diagram is a Unified Modeling Language (UML) diagram that illustrates the sequence of messages between objects in an interaction. A sequence diagram consists of a group of objects that are represented by lifelines, and the messages that they exchange over time during the interaction.A sequence diagram shows the sequence of messages passed between objects. Sequence diagrams can also show the control structures between objects.

.

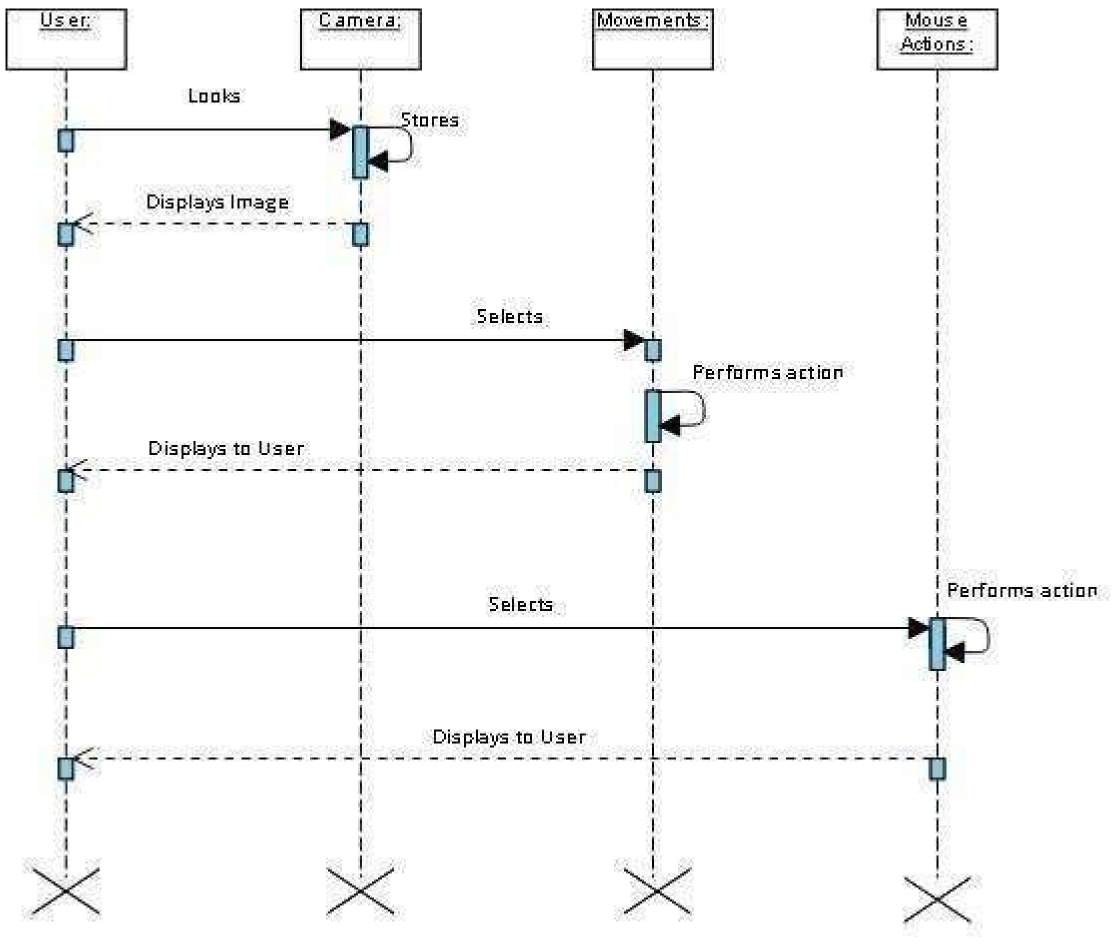


Figure 3.4 Sequence Diagram for Finger Counting and Virtual Mouse using CV Learn

3.5 ACTIVITY DIAGRAM

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system.Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc

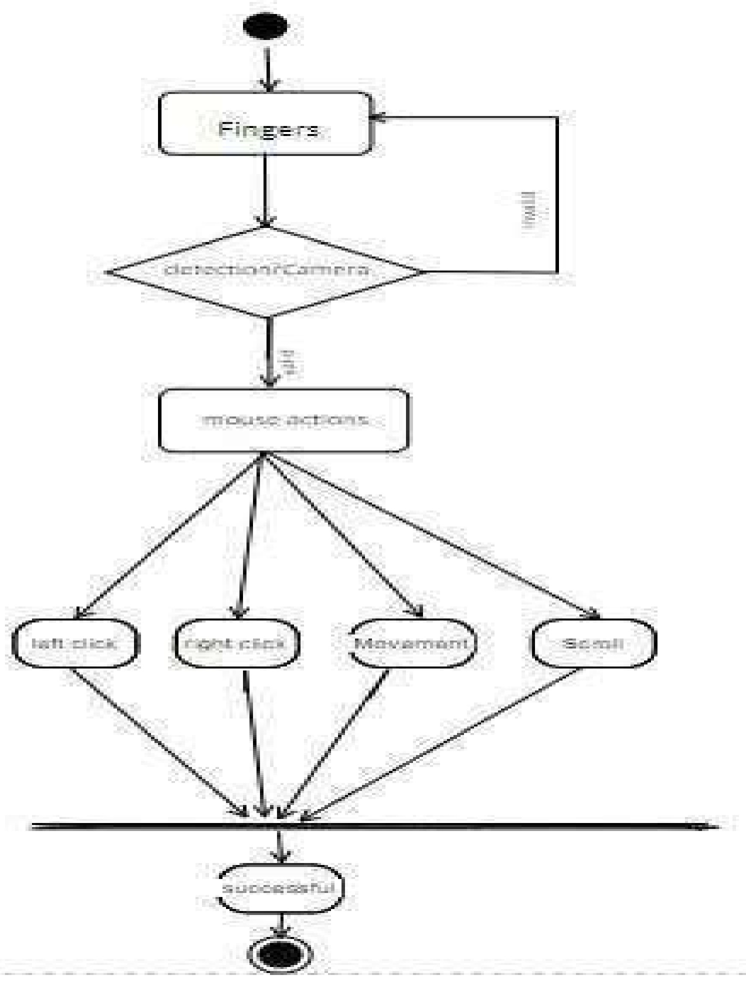


Figure 3.5 Activity Diagram for Finger Counting and Virtual Mouse using CV Learn

4. IMPLEMENTATION

4. IMPLEMENTATION

4.1 HANDTRACKING:

```
from cvlearn import HandTrackingModule as handTracker

import cv2
cap = cv2.VideoCapture(0)
detector = handTracker.handDetector()while True:
    ret, img = cap.read()
    img = detector.findHands(img)

    cv2.imshow("Result", img)
    cv2.waitKey(1)
```

4.2 FINGERCOUNTING:

```
from cvlearn import FingerCounter as fc
import cvlearn.HandTrackingModule as handTracker
import cv2

cap = cv2.VideoCapture(0)

detector = handTracker.handDetector(maxHands=1)

counter = fc.FingerCounter()while True:
    ret, frame = cap.read()
    frame =cv2.flip(frame, 180)

    frame = detector.findHands(frame)
    lmList, bbox = detector.findPosition(frame)

    if lmList:
        frame1 = counter.drawCountedFingers(frame, lmList, bbox)

    cv2.imshow("res", frame)
    key = cv2.waitKey(1)
    if key == 27:
        break
cv2.destroyAllWindows()
```

4.3 VIRTUAL MOUSE:

```
import cv2
import mediapipe as mp
import pyautogui
import math
from enum import IntEnum
from ctypes import cast, POINTER
from comtypes import CLSCTX_ALL
from pycaw.pycaw import AudioUtilities, IAudioEndpointVolume
from google.protobuf.json_format import MessageToDict
import screen_brightness_control as sbcontrol

pyautogui.FAILSAFE = False
mp_drawing = mp.solutions.drawing_utils
mp_hands = mp.solutions.hands


 # Gesture Encodings
class Gest(IntEnum):
     # Binary Encoded
    FIST = 0
    PINKY = 1
    RING = 2
    MID = 4
    LAST3 = 7
    INDEX = 8
    FIRST2 = 12
    LAST4 = 15
    THUMB = 16
    PALM = 31

    # Extra Mappings
    V_GEST = 33
    TWO_FINGER_CLOSED = 34
    PINCH_MAJOR =35
    PINCH_MINOR =36


# Multi-handedness Labels
class HLabel(IntEnum):
    MINOR = 0
    MAJOR = 1


# Convert Mediapipe Landmarks to recognizable Gestures
class HandRecog:

    def _init_(self, hand_label):
        self.finger = 0
        self.ori_gesture = Gest.PALM
        self.prev_gesture = Gest.PALM
        self.frame_count = 0
        self.hand_result = None
        self.hand_label = hand_label

    def update_hand_result(self, hand_result):
        self.hand_result = hand_result

    def get_signed_dist(self, point):
        sign = -1
        if self.hand_result.landmark[point[0]].y < self.hand_result.landmark[point[1]].y:
            sign = 1
        dist = (self.hand_result.landmark[point[0]].x - self.hand_result.landmark[point[1]].x) ** 2
        dist += (self.hand_result.landmark[point[0]].y - self.hand_result.landmark[point[1]].y) ** 2
        dist = math.sqrt(dist)
        return dist * sign

    def get_dist(self, point):
        dist = (self.hand_result.landmark[point[0]].x - self.hand_result.landmark[point[1]].x) ** 2
        dist += (self.hand_result.landmark[point[0]].y - self.hand_result.landmark[point[1]].y) ** 2
        dist = math.sqrt(dist)
        return dist

    def get_dz(self, point):
        return abs(self.hand_result.landmark[point[0]].z - self.hand_result.landmark[point[1]].z)

    # Function to find Gesture Encoding using current finger_state.
    # Finger_state: 1 if finger is open, else 0
    def set_finger_state(self):
        if self.hand_result == None:
            return

        points = [[8, 5, 0], [12, 9, 0], [16, 13, 0], [20, 17, 0]]
        self.finger = 0
        self.finger = self.finger | 0 # thumb
        for idx, point in enumerate(points):

            dist = self.get_signed_dist(point[:2])
            dist2 = self.get_signed_dist(point[1:])

            try:
                ratio = round(dist / dist2, 1)
            except:
                ratio = round(dist1 / 0.01, 1)

            self.finger = self.finger << 1
            if ratio > 0.5:
                self.finger = self.finger | 1

    # Handling Fluctations due to noise
    def get_gesture(self):
        if self.hand_result == None:
            return Gest.PALM

        current_gesture = Gest.PALM
        if self.finger in [Gest.LAST3, Gest.LAST4] and self.get_dist([8, 4]) < 0.05:
            if self.hand_label == HLabel.MINOR:
                current_gesture = Gest.PINCH_MINOR
            else:
                current_gesture = Gest.PINCH_MAJOR

        elif Gest.FIRST2 == self.finger:
            point = [[8, 12], [5, 9]]
            dist1 = self.get_dist(point[0])
            dist2 = self.get_dist(point[1])
            ratio = dist1 / dist2
            if ratio > 1.7:
                current_gesture = Gest.V_GEST
            else:
                if self.get_dz([8, 12]) < 0.1:
                    current_gesture = Gest.TWO_FINGER_CLOSED
                else:
                    current_gesture = Gest.MID

        else:
            current_gesture = self.finger

        if current_gesture == self.prev_gesture:
            self.frame_count += 1
        else:
            self.frame_count = 0

        self.prev_gesture =current_gesture

        if self.frame_count > 4:
            self.ori_gesture =current_gesture
        return self.ori_gesture


# Executes commands according to detected gestures
class Controller:
    tx_old = 0
    ty_old = 0
    trial = True
    flag = False
    grabflag = False
    pinchmajorflag = False
    pinchminorflag = False
    pinchstartxcoord = None
    pinchstartycoord = None
    pinchdirectionflag = None
```

```
prevpinchlv = 0
pinchlv = 0
framecount = 0
prev_hand = None
pinch_threshold = 0.3

def getpinchylv(hand_result):
    dist = round((Controller.pinchstartycoord - hand_result.landmark[8].y) * 10, 1)
    return dist

def getpinchxlv(hand_result):
    dist = round((hand_result.landmark[8].x - Controller.pinchstartxcoord) * 10, 1)
    return dist

def changesystembrightness(self):
    currentBrightnessLv = sbcontrol.get_brightness() / 100.0
    currentBrightnessLv += Controller.pinchlv / 50.0
    if currentBrightnessLv > 1.0:
        currentBrightnessLv = 1.0
    elif currentBrightnessLv < 0.0:
        currentBrightnessLv = 0.0
    sbcontrol.fade_brightness(int(100 * currentBrightnessLv), start=sbcontrol.get_brightness())

def changesystemvolume(self):
    devices = AudioUtilities.GetSpeakers()
    interface = devices.Activate(IAudioEndpointVolume.iid, CLSCTX_ALL, None)
    volume = cast(interface, POINTER(IAudioEndpointVolume))
    currentVolumeLv = volume.GetMasterVolumeLevelScalar()
    currentVolumeLv += Controller.pinchlv / 50.0
    if currentVolumeLv > 1.0:
        currentVolumeLv = 1.0
    elif currentVolumeLv < 0.0:
        currentVolumeLv = 0.0
    volume.SetMasterVolumeLevelScalar(currentVolumeLv, None)

def scrollVertical(self):
    pyautogui.scroll(120 if Controller.pinchlv > 0.0 else -120)

def scrollHorizontal(self):
    pyautogui.keyDown('shift')
    pyautogui.keyDown('ctrl')
    pyautogui.scroll(-120 if Controller.pinchlv > 0.0 else 120)
    pyautogui.keyUp('ctrl')
    pyautogui.keyUp('shift')

# Locate Hand to get Cursor Position
# Stabilize cursor by Dampening
def get_position(hand_result):
    point = 9
    position = [hand_result.landmark[point].x, hand_result.landmark[point].y]
    sx, sy = pyautogui.size()
    x_old, y_old = pyautogui.position()
    x = int(position[0] * sx)
    y = int(position[1] * sy)
    if Controller.prev_hand is None:
        Controller.prev_hand = x, y
    delta_x = x - Controller.prev_hand[0]
    delta_y = y - Controller.prev_hand[1]

    distsq = delta_x * 2 + delta_y * 2
    ratio = 1
    Controller.prev_hand = [x, y]

    if distsq <= 25:
        ratio = 0
    elif distsq <= 900:
        ratio = 0.07 * (distsq ** (1 / 2))
    else:
        ratio = 2.1
    x, y = x_old + delta_x * ratio, y_old + delta_y * ratio
    return (x, y)

def pinch_control_init(hand_result):
    Controller.pinchstartxcoord = hand_result.landmark[8].x
    Controller.pinchstartycoord = hand_result.landmark[8].y
    Controller.pinchlv = 0
    Controller.prevpinchlv = 0
    Controller.framecount = 0

# Hold final position for 5 frames to change status
def pinch_control(hand_result, controlHorizontal, controlVertical):
    if Controller.framecount == 5:
        Controller.framecount = 0
        Controller.pinchlv = Controller.prevpinchlv

        if Controller.pinchdirectionflag == True:
            controlHorizontal() # x

        elif Controller.pinchdirectionflag == False:
            controlVertical() # y

    lvx = Controller.getpinchxlv(hand_result)
    lvy = Controller.getpinchylv(hand_result)

    if abs(lvy) > abs(lvx) and abs(lvy) > Controller.pinch_threshold:
        Controller.pinchdirectionflag = False
        if abs(Controller.prevpinchlv - lvy) < Controller.pinch_threshold:
            Controller.framecount += 1
        else:
            Controller.prevpinchlv = lvy
            Controller.framecount = 0
    elif abs(lvx) > Controller.pinch_threshold:
        Controller.pinchdirectionflag = True
        if abs(Controller.prevpinchlv - lvx) < Controller.pinch_threshold:
            Controller.framecount += 1
        else:
            Controller.prevpinchlv = lvx
            Controller.framecount = 0

def handle_controls(gesture, hand_result):x,
    y = None, None
    if gesture != Gest.PALM:
        x, y = Controller.get_position(hand_result)

    # flag reset
    if gesture != Gest.FIST and Controller.grabflag:
        Controller.grabflag = False
        pyautogui.mouseUp(button="left")

    if gesture != Gest.PINCH_MAJOR and Controller.pinchmajorflag:
        Controller.pinchmajorflag = False

    if gesture != Gest.PINCH_MINOR and Controller.pinchminorflag:
        Controller.pinchminorflag = False

    # implementation
    if gesture == Gest.V_GEST:
        Controller.flag = True
        pyautogui.moveTo(x, y, duration=0.1)

    elif gesture == Gest.FIST:
        if not Controller.grabflag:
            Controller.grabflag = True
            pyautogui.mouseDown(button="left")
        pyautogui.moveTo(x, y, duration=0.1)

    elif gesture == Gest.MID and Controller.flag:
        pyautogui.click()
        Controller.flag = False

    elif gesture == Gest.INDEX and Controller.flag:
        pyautogui.click(button='right')
        Controller.flag = False

    elif gesture == Gest.TWO_FINGER_CLOSED and Controller.flag:pyautogui.doubleClick()
        Controller.flag = False

    elif gesture == Gest.PINCH_MINOR:
        if Controller.pinchminorflag == False:
            Controller.pinch_control_init(hand_result)
            Controller.pinchminorflag = True
            Controller.pinch_control(hand_result, Controller.scrollHorizontal, Controller.scrollVertical)

        elif gesture == Gest.PINCH_MAJOR:
            if Controller.pinchmajorflag == False:
                Controller.pinch_control_init(hand_result)
                Controller.pinchmajorflag = True
            Controller.pinch_control(hand_result, Controller.changesystembrightness,
Controller.changesystemvolume)


'''
------------------------------------------------------------- Main Class -------------------------------------------------------------
    Entry point of Gesture Controller
'''


class GestureController:
    gc_mode = 0
    cap = None
    CAM_HEIGHT = None
    CAM_WIDTH = None
    hr_major = None # Right Hand by default
    hr_minor = None # Left hand by default
    dom_hand = True

    def _init_(self):
        GestureController.gc_mode = 1
        GestureController.cap = cv2.VideoCapture(0)
        GestureController.CAM_HEIGHT =
GestureController.cap.get(cv2.CAP_PROP_FRAME_HEIGHT)
        GestureController.CAM_WIDTH =
GestureController.cap.get(cv2.CAP_PROP_FRAME_WIDTH)

    def classify_hands(results):
        left, right = None, None
        try:
            handedness_dict = MessageToDict(results.multi_handedness[0])
            if handedness_dict['classification'][0]['label'] == 'Right':
                right =results.multi_hand_landmarks[0]
            else:
                left =results.multi_hand_landmarks[0]
        except:
            pass

        try:
            handedness_dict = MessageToDict(results.multi_handedness[1])
            if handedness_dict['classification'][0]['label'] == 'Right':
                right =results.multi_hand_landmarks[1]
            else:
                left =results.multi_hand_landmarks[1]
        except:
            pass

        if GestureController.dom_hand == True:
            GestureController.hr_major = right
            GestureController.hr_minor = left
        else:
            GestureController.hr_major = left
            GestureController.hr_minor = right

    def start(self):

        handmajor = HandRecog(HLabel.MAJOR)
        handminor = HandRecog(HLabel.MINOR)

        with mp_hands.Hands(max_num_hands=2, min_detection_confidence=0.5,
min_tracking_confidence=0.5) as hands:
            while GestureController.cap.isOpened() and GestureController.gc_mode:
                success, image = GestureController.cap.read()

                if not success:
                    print("Ignoring empty camera frame.")
                    continue

                image = cv2.cvtColor(cv2.flip(image, 1), cv2.COLOR_BGR2RGB)
                image.flags.writeable = False
                results = hands.process(image)

                image.flags.writeable = True
                image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)

                if results.multi_hand_landmarks:
                    GestureController.classify_hands(results)
                    handmajor.update_hand_result(GestureController.hr_major)
                    handminor.update_hand_result(GestureController.hr_minor)

                    handmajor.set_finger_state()
                    handminor.set_finger_state()
                    gest_name = handminor.get_gesture()

                    if gest_name == Gest.PINCH_MINOR:
                        Controller.handle_controls(gest_name, handminor.hand_result)
                    else:
                        gest_name = handmajor.get_gesture()
                        Controller.handle_controls(gest_name, handmajor.hand_result)

                    for hand_landmarks in results.multi_hand_landmarks:
                        mp_drawing.draw_landmarks(image, hand_landmarks,
mp_hands.HAND_CONNECTIONS)
                else:
                    Controller.prev_hand = None
                cv2.imshow('Gesture Controller', image)
            if cv2.waitKey(5) & 0xFF == 13:
                break
        GestureController.cap.release()
        cv2.destroyAllWindows()


# uncomment to run directly
gc1 = GestureController()
gc1.start()
```

5. SCREENSHOTS

5. SCREENSHOTS

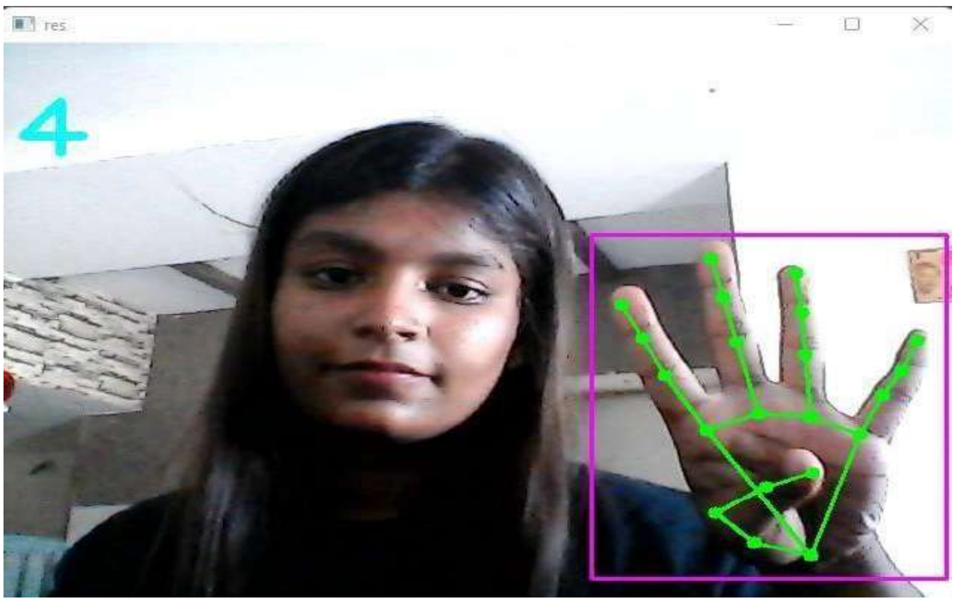


Figure 5.1 Finger Counting

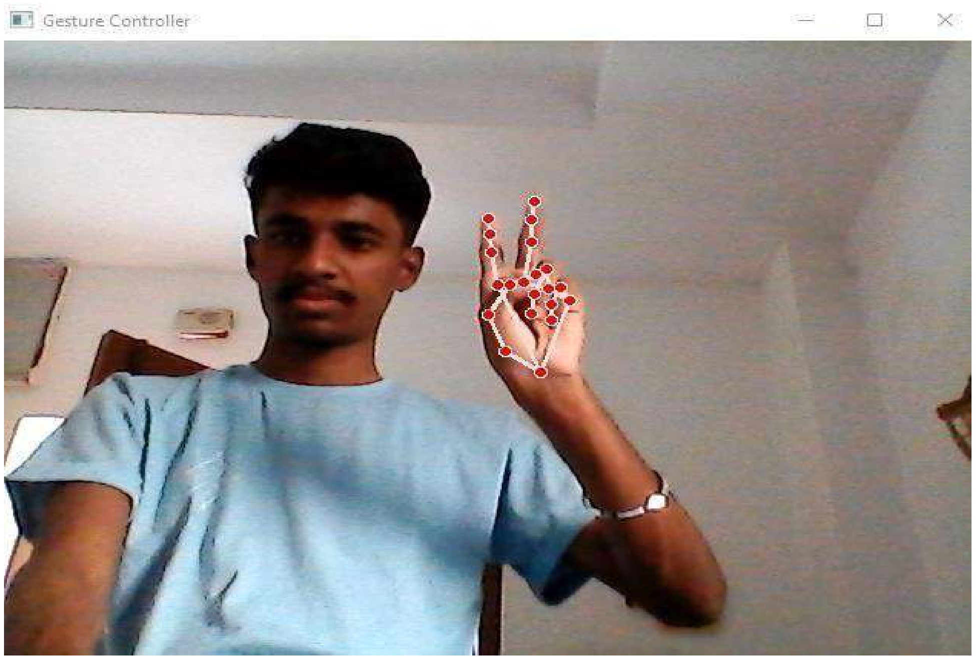


Figure 5.2 Gesture for scrolling Up and Down

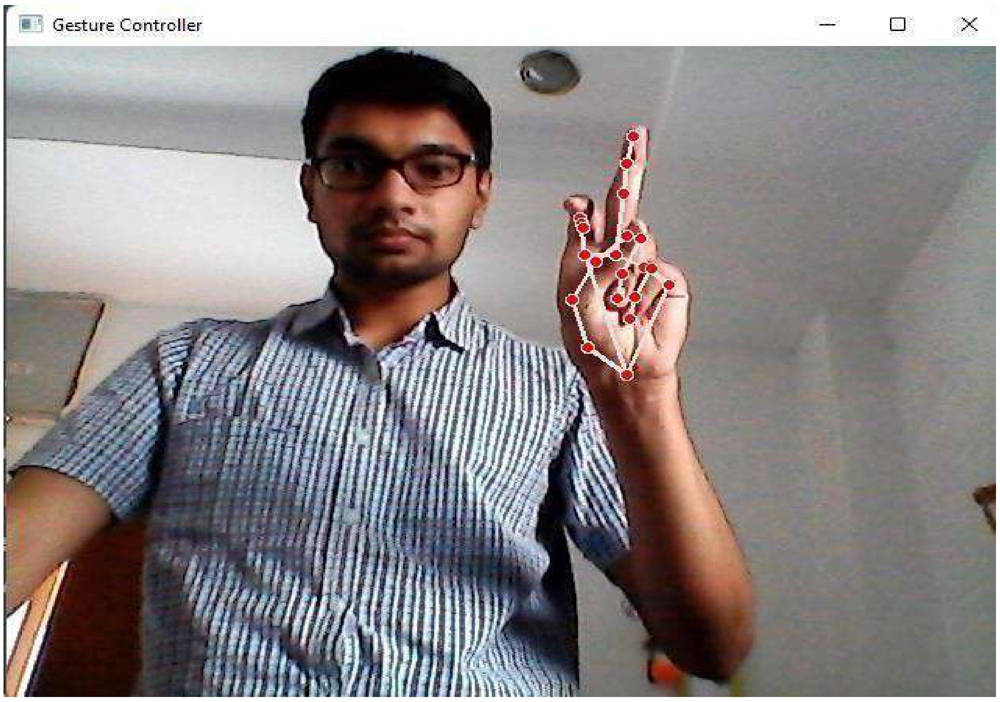


Figure 5.3 Gesture for Left-Click

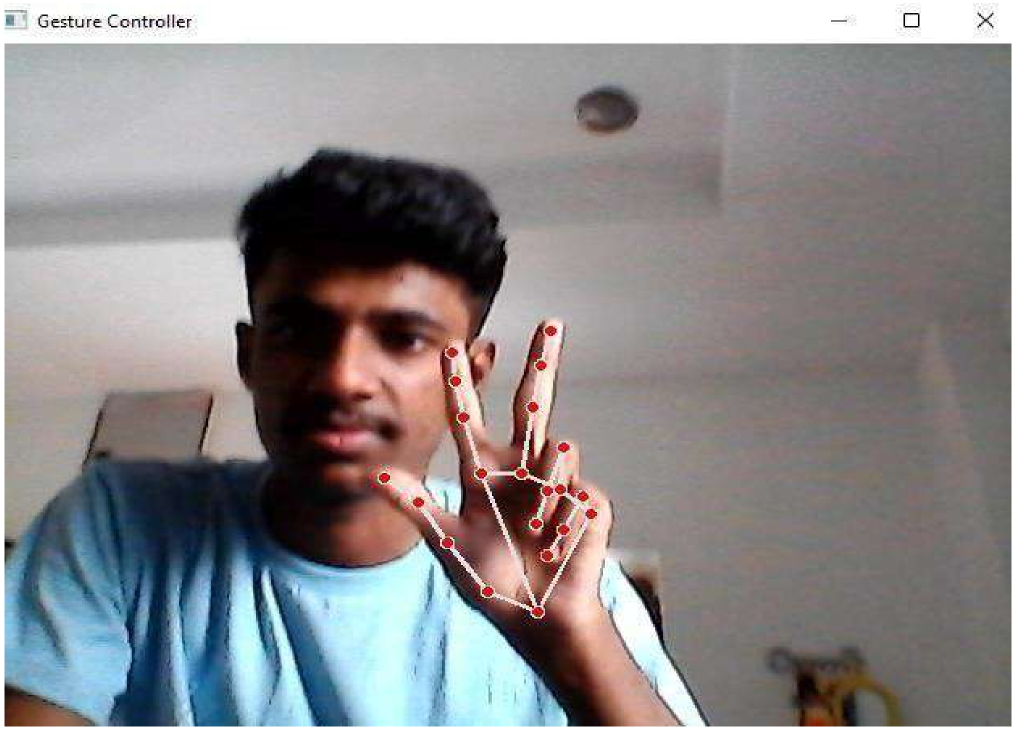


Figure 5.4 Gesture for Cursor Movement

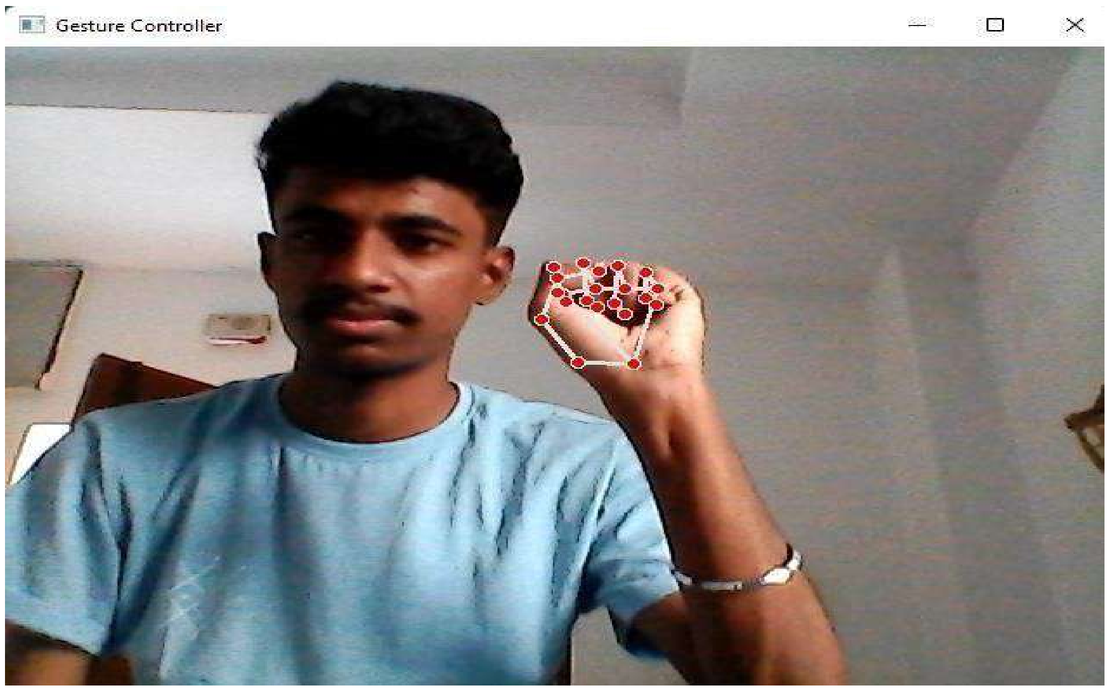


Figure 5.5 Gesture for Drop And Drag

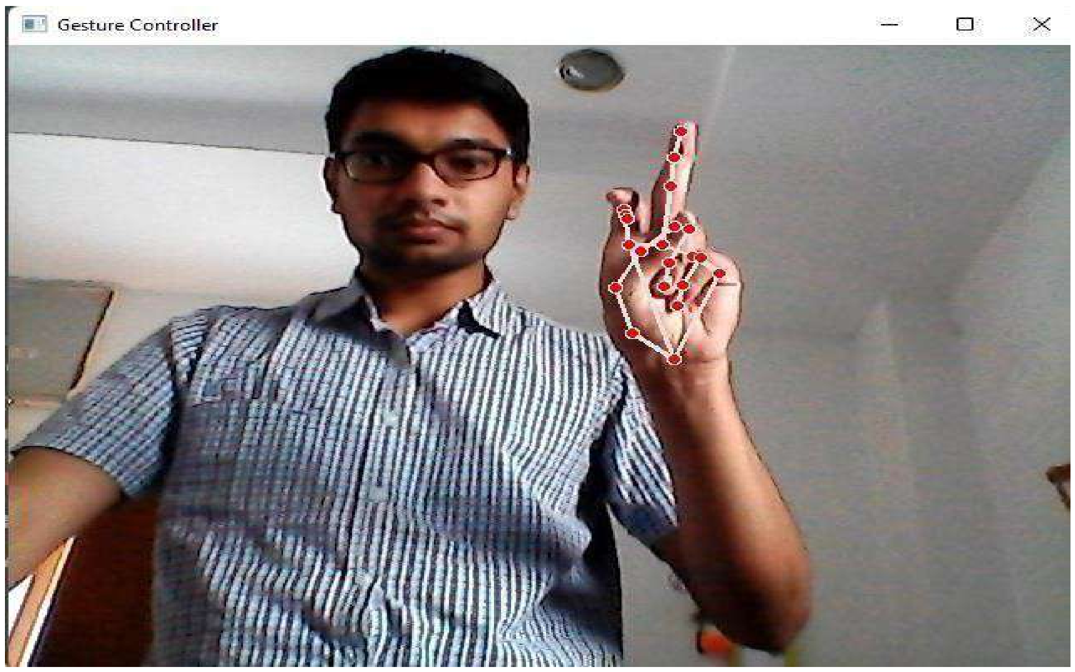


Figure 5.6 Gesture for Right-Click

6. TESTING

6.TESTING

6.1 INTRODUCTION TO TESTING

An estimate says that 50% of whole software development process should be tested.The errors that are occurred may destroy the entire software. Software testing is done while coding by the developers and through testing is conducted by testing experts at various level of code such as module testing, program testing, in-house testing and testing the product at user's end. Early discovery of errors and their remedy is the key to reliable software.

6.2 TYPES OF TESTING

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.

INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

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, keyfunctions, or special test cases.

6.3 TEST CASES

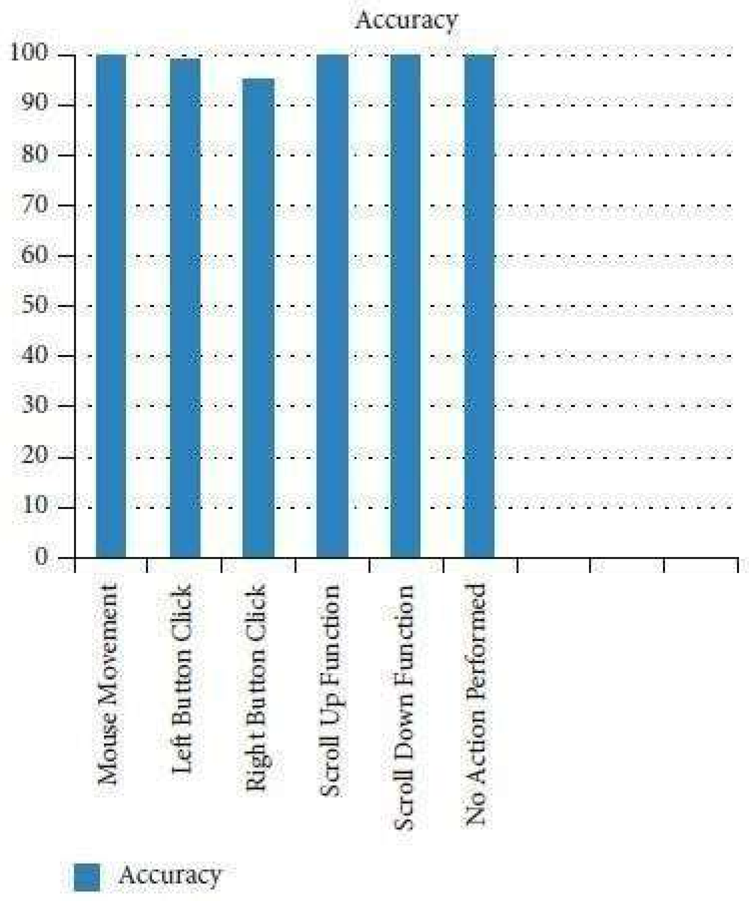


Figure 6.3 Test case status

7.CONCLUSION

7.1 PROJECT CONCLUSION

The main objective of the AI virtual mouse system is to control the mouse cursor functions by using the hand gestures instead of using a physical mouse. The proposed system can be achieved by using a webcam or a built-in camera which detects the hand gestures and hand tip and processes these frames to perform the particular mouse functions.From the results of the model, we can come to a conclusion that the proposed AI virtual mouse system has performed very well and has a greater accuracy compared to the existing models and also the model overcomes most of the limitations of the existing systems. Since the proposed model has greater accuracy, the AI virtual mouse can be used for real-world applications, and also, it can be used to reduce the spread of COVID-19, since the proposed mouse system can be used virtually using hand gestures without using the traditional physical mouse.

The model has some limitations such as small decrease in accuracy in right click mouse function and some difficulties in clicking and dragging to select the text. Hence, we will work next to overcome these limitations by improving the finger tip detection algorithm to produce more accurate results.

7.2 FUTURE SCOPE

The model proposed has drawbacks like accuracy of left and right click These limitations will over come in the future work.Further more this can be added with a virtual keyboard as well.

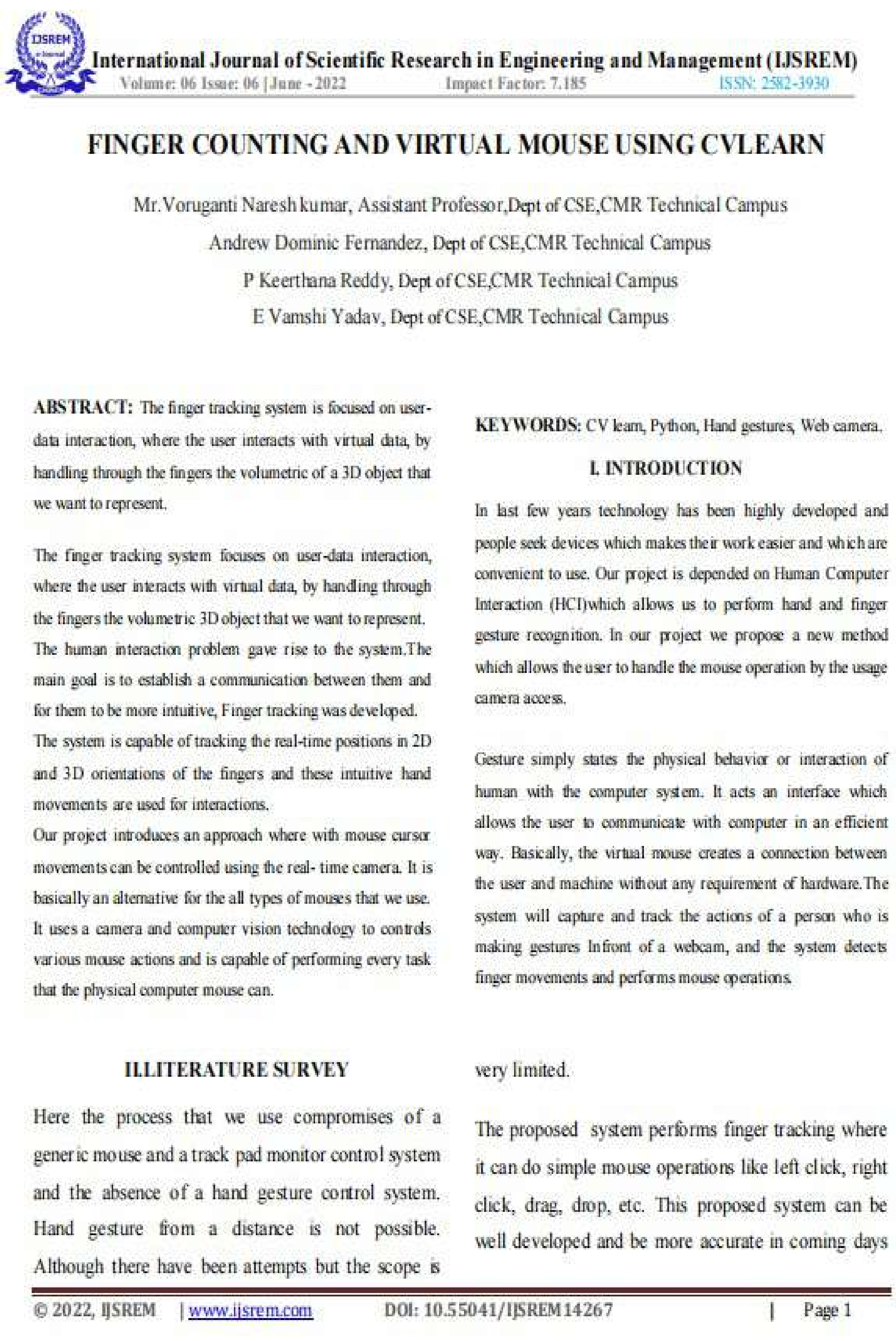
8.BIBILOGRAPHY

8.1 REFERENCES

1. **Computer Vision : Models, Learning and Interface, 2012.**
2. **Computer Vision : Algorithms and Applications 2010.**
3. **Practical Deep Learning for Cloud, Mobile & Edge.**
4. **Learning OpenCV 4 Computer Vision with Python 3.**
5. **https://xd.adobe.com/ideas/principles/emerging-technology/what-is-computer-vision-how-does-it-work/**
6. **https://bdtechtalks.com/2019/01/14/what-is-computer-vision/**

8.2 GITHUB LINK

https://github.com/vamshi3145/Virtual_Mouse.py



International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 06 | June - 2022 Impact Factor: 7.185 ISSN: 2582-3930

FINGER COUNTING AND VIRTUAL MOUSE USING CVLEARN

Mr.Voruganti Naresh kumar, Assistant Professor,Dept of CSE,CMR Technical Campus
Andrew Dominic Fernandez, Dept of CSE,CMR Technical Campus
P Keerthana Reddy, Dept of CSE,CMR Technical Campus
E Vamshi Yadav, Dept of CSE,CMR Technical Campus

ABSTRACT: The finger tracking system is focused on user-data interaction, where the user interacts with virtual data, by handling through the fingers the volumetric of a 3D object that we want to represent.

The finger tracking system focuses on user-data interaction, where the user interacts with virtual data, by handling through the fingers the volumetric 3D object that we want to represent. The human interaction problem gave rise to the system.The main goal is to establish a communication between them and for them to be more intuitive, Finger tracking was developed. The system is capable of tracking the real-time positions in 2D and 3D orientations of the fingers and these intuitive hand movements are used for interactions.
Our project introduces an approach where with mouse cursor movements can be controlled using the real- time camera. It is basically an alternative for the all types of mouses that we use. It uses a camera and computer vision technology to controls various mouse actions and is capable of performing every task that the physical computer mouse can.

KEYWORDS: CV learn, Python, Hand gestures, Web camera.

I. INTRODUCTION

In last few years technology has been highly developed and people seek devices which makes their work easier and which are convenient to use. Our project is depended on Human Computer Interaction (HCI)which allows us to perform hand and finger gesture recognition. In our project we propose a new method which allows the user to handle the mouse operation by the usage camera access.

Gesture simply states the physical behavior or interaction of human with the computer system. It acts an interface which allows the user to communicate with computer in an efficient way. Basically, the virtual mouse creates a connection between the user and machine without any requirement of hardware.The system will capture and track the actions of a person who is making gestures Infront of a webcam, and the system detects finger movements and performs mouse operations.

II.LITERATURE SURVEY

Here the process that we use compromises of a generic mouse and a track pad monitor control system and the absence of a hand gesture control system. Hand gesture from a distance is not possible. Although there have been attempts but the scope is very limited.

The proposed system performs finger tracking where it can do simple mouse operations like left click, right click, drag, drop, etc. This proposed system can be well developed and be more accurate in coming days

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 06 | June - 2022 Impact Factor: 7.185 ISSN: 2582-3930

by using advanced technolgy .

Below is the technique that we used:

Finger gestures: A sensor(webcam) will identify the gestures of finger movements. These certain gestures need to be performed in order to accomplish the desired mouse action

III.PROPOSED METHODOLOGY

Here we have three modules: user module, camera module and system module.

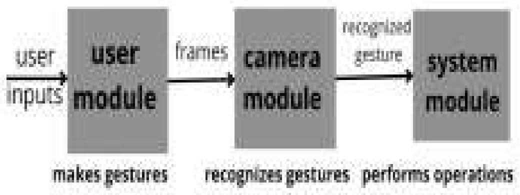


Figure 3.1: Proposed architecture

USER MODULE: In the user module, the user makes hand gestures or finger gestures

sitting Infront of the real time camera and these gestures will be converted into 2D or 3D images

CAMERA MODULE: Here we are going to use an inbuilt camera by using which we capture the gestures made by the user and process those images into frames for recognizing different sorts of gestures

SYSTEM MODULE: This module is responsible for calibrating the detected hand gestures to their perspective actions. All windows are opened automatically the gestures are taken as inputs from the user and accordingly PyAutoGui gets into work and perform various operations

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 06 | June - 2022 Impact Factor: 7.185 ISSN: 2582-3930

V.RESULTS

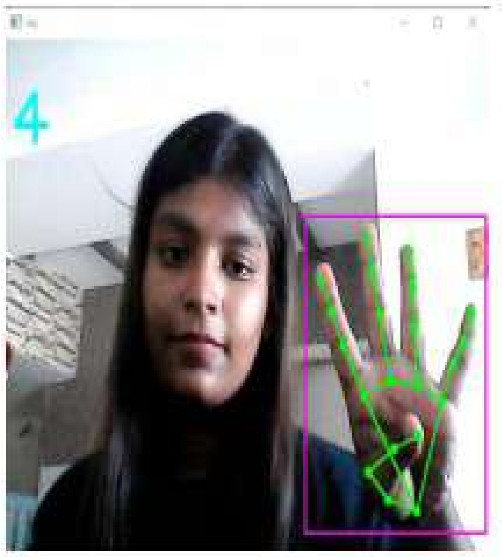


Figure 5.1 : Finger Counting

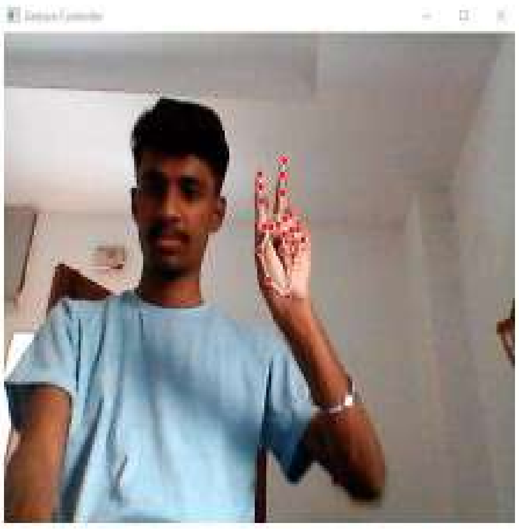


Figure 5.2: Gesture For ScrollingScreen Up

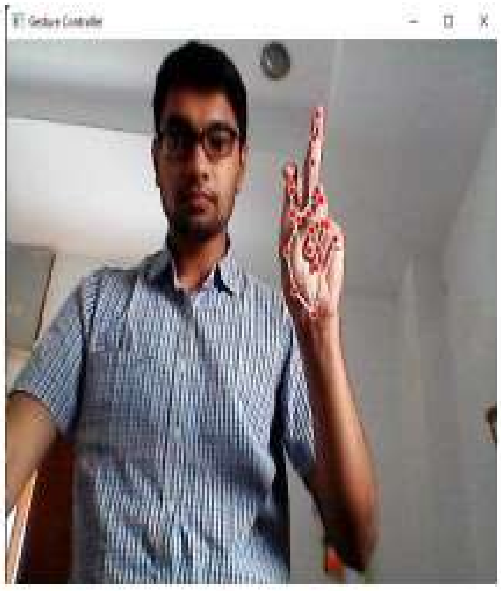


Figure 5.3:GestureFor Left Click Operation

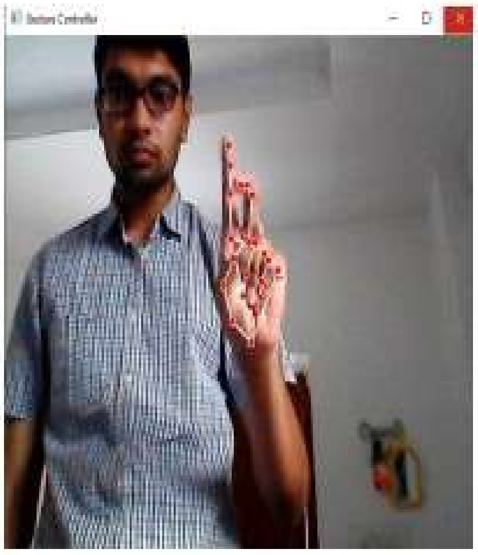


Figure 5.4: Gesture For Right Click Operation

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 06 | June - 2022 Impact Factor: 7.185 ISSN: 2582-3930

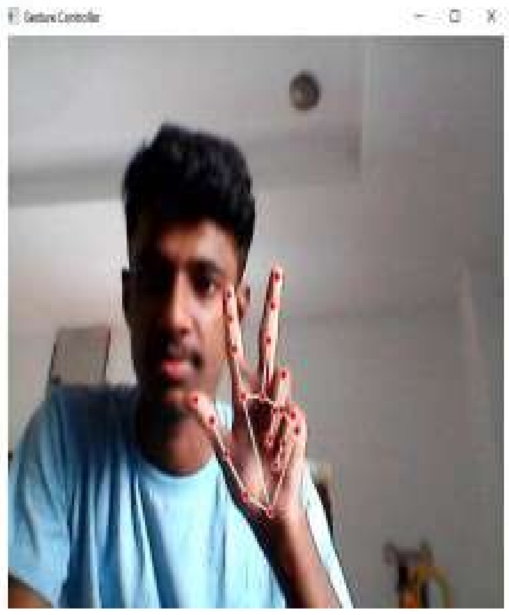


Figure 5.5:Gesture For Virtual Mouse Movement

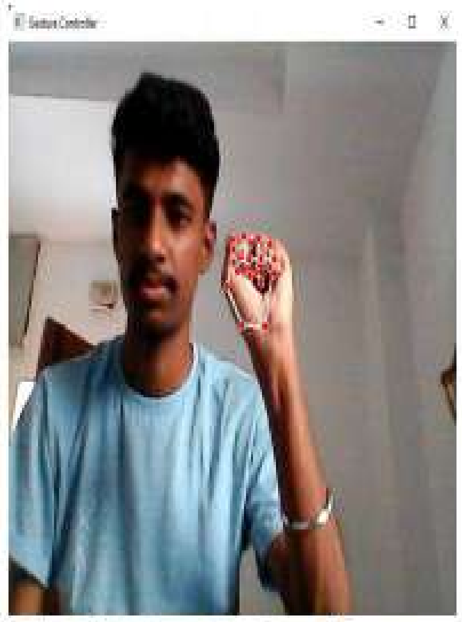
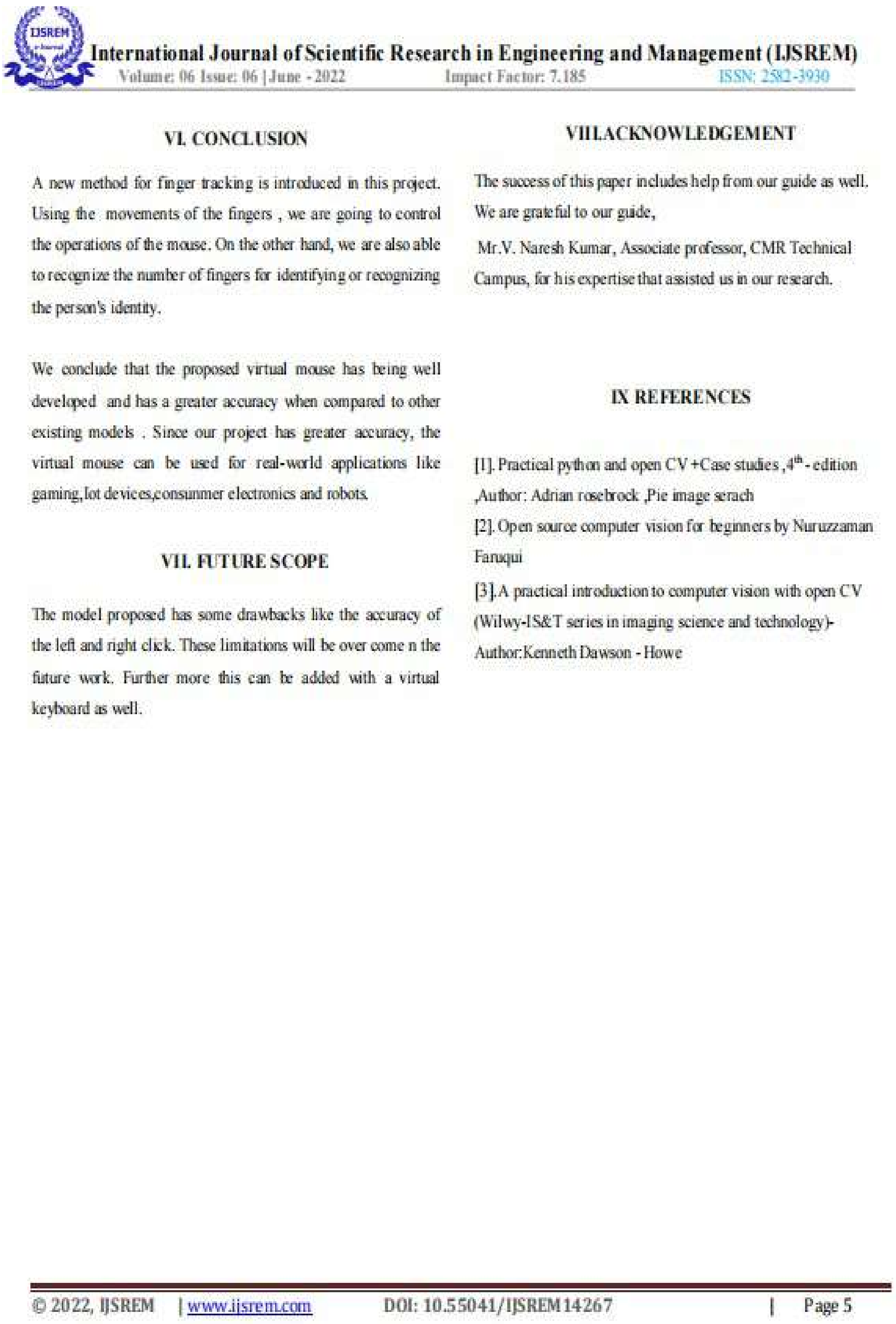


Figure 5.6: Gesture For Drag and Drop



International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 06 | June - 2022 Impact Factor: 7.185 ISSN: 2582-3930

VI. CONCLUSION

A new method for finger tracking is introduced in this project. Using the movements of the fingers , we are going to control the operations of the mouse. On the other hand, we are also able to recognize the number of fingers for identifying or recognizing the person's identity.

We conclude that the proposed virtual mouse has being well developed and has a greater accuracy when compared to other existing models . Since our project has greater accuracy, the virtual mouse can be used for real-world applications like gaming,Iot devices,consunmer electronics and robots.

VII. FUTURE SCOPE

The model proposed has some drawbacks like the accuracy of the left and right click. These limitations will be over come n the future work. Further more this can be added with a virtual keyboard as well.

VIII.ACKNOWLEDGEMENT

The success of this paper includes help from our guide as well. We are grateful to our guide,

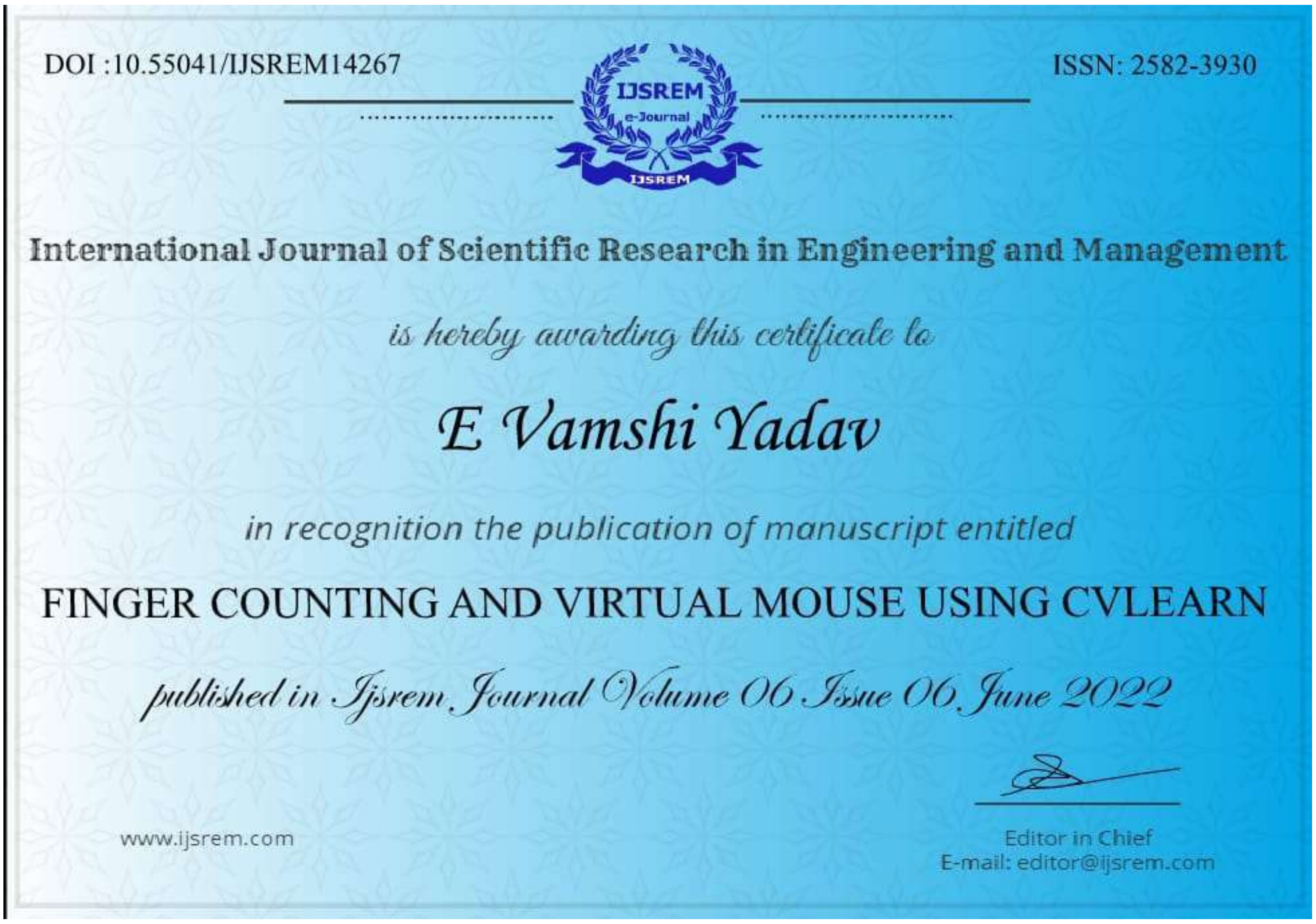
Mr.V. Naresh Kumar, Associate professor, CMR Technical Campus, for his expertise that assisted us in our research.

IX REFERENCES

[1]. Practical python and open CV +Case studies ,4th - edition ,Author: Adrian rosebrock ,Pie image serach

[2]. Open source computer vision for beginners by Nuruzzaman Faruqui

[3].A practical introduction to computer vision with open CV (Wilwy-IS&T series in imaging science and technology)- Author:Kenneth Dawson - Howe



DOI :10.55041/IJSREM14267

ISSN: 2582-3930

International Journal of Scientific Research in Engineering and Management

is hereby awarding this certificate to

E Vamshi Yadav

in recognition the publication of manuscript entitled

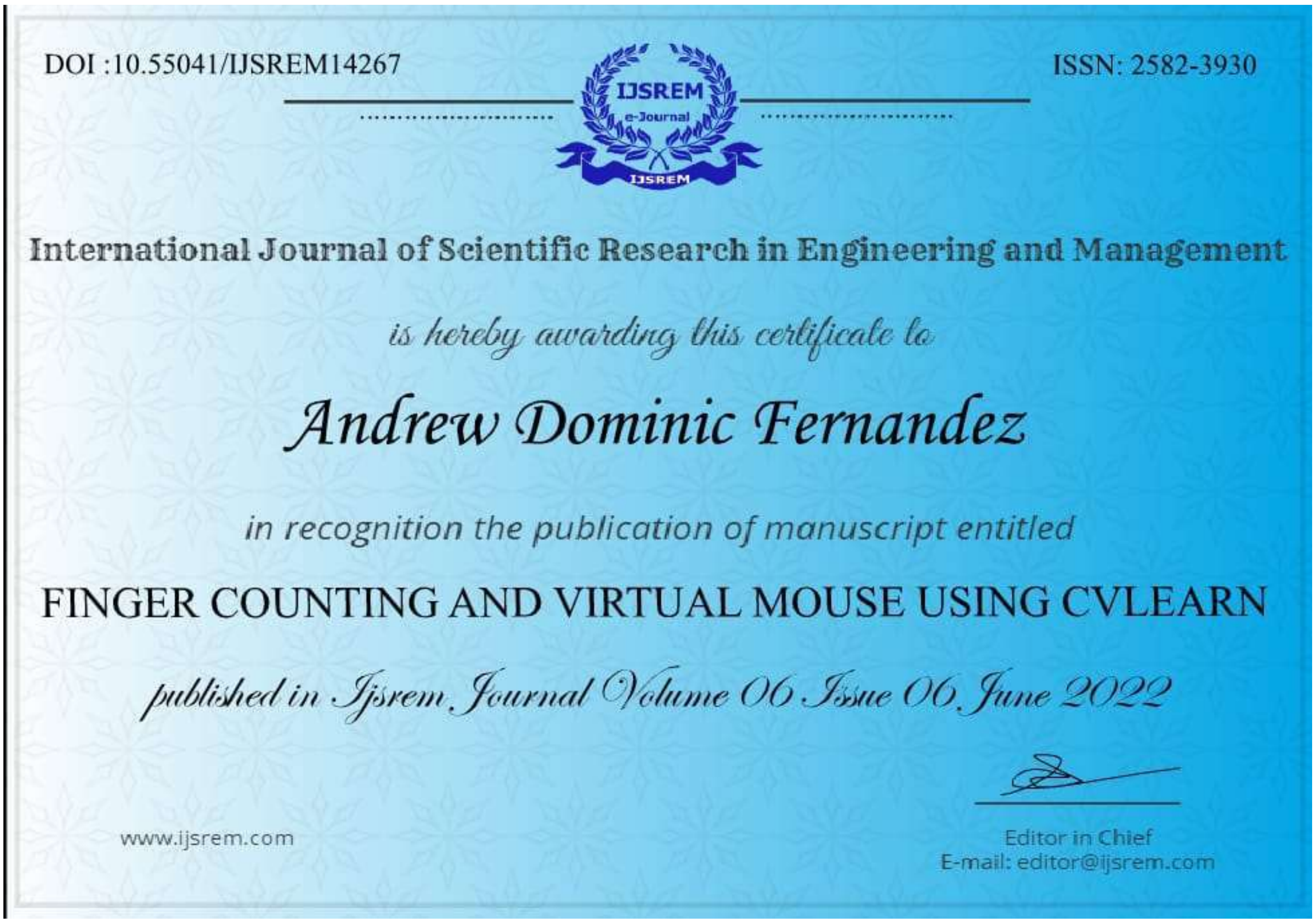
FINGER COUNTING AND VIRTUAL MOUSE USING CVLEARN

published in Ijsrem Journal Volume 06 Issue 06 June 2022

[illegible]

www.ijsrem.com

Editor in Chief
E-mail: editor@ijsrem.com



DOI :10.55041/IJSREM14267

ISSN: 2582-3930

IJSREM e-Journal IJSREM

International Journal of Scientific Research in Engineering and Management

is hereby awarding this certificate to

Andrew Dominic Fernandez

in recognition the publication of manuscript entitled

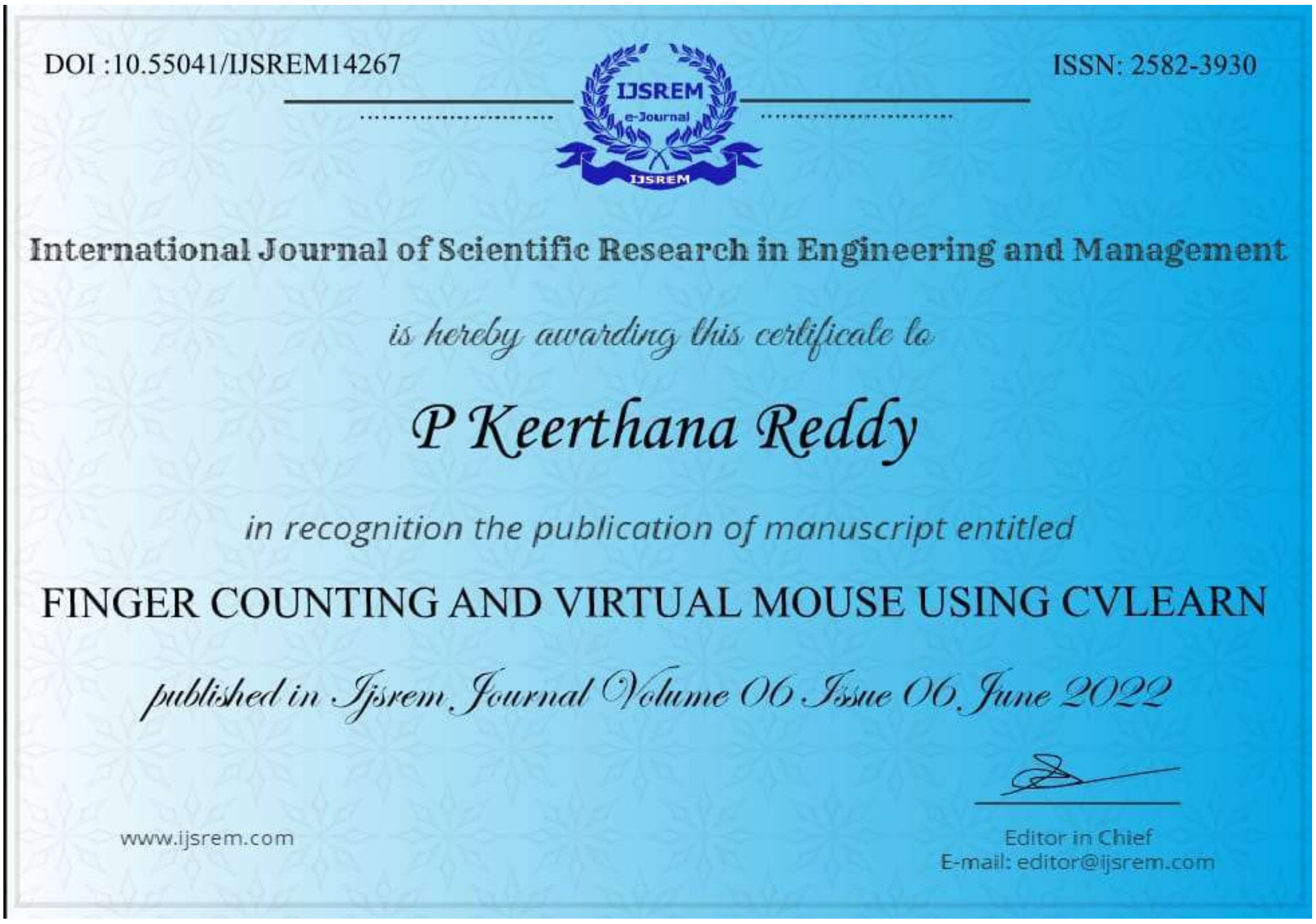
FINGER COUNTING AND VIRTUAL MOUSE USING CVLEARN

published in Ijsrem Journal Volume 06 Issue 06, June 2022

[illegible]

www.ijsrem.com

Editor in Chief
E-mail: editor@ijsrem.com



DOI :10.55041/IJSREM14267

ISSN: 2582-3930

International Journal of Scientific Research in Engineering and Management

is hereby awarding this certificate to

P Keerthana Reddy

in recognition the publication of manuscript entitled

FINGER COUNTING AND VIRTUAL MOUSE USING CVLEARN

published in Ijsrem Journal Volume 06 Issue 06, June 2022

www.ijsrem.com

[illegible]

Editor in Chief
E-mail: editor@ijsrem.com