



GESTURE CONTROLLED VIRTUAL MOUSE

AMRITHA U

MES20MCA-2006

PRODUCT OWNER : Mr. MUHAMMAD JABIR C

TABLE OF CONTENTS

CONTENT	PAGE NO
Introduction	1
Developing Environment	2
Methodology	3- 13
Project Plan	14
Product Backlog	15
Sprint Backlog Plan	16-19
Sprint Backlog Actual	20-21

INTRODUCTION

- ❑ In this project proposes a way to control the position of the cursor with the bare hands without using any electronic device.
- ❑ The operations like clicking and dragging etc, of objects will be performed with different hand gestures.
- ❑ The proposed system will only require a webcam as an input device.
- ❑ The software's that will be required to implement the proposed system are OpenCV, Mediapipe and python.
- ❑ The output of the camera will be displayed on the system's screen so that it can be further changed by the user.

DEVELOPING ENVIRONMENT

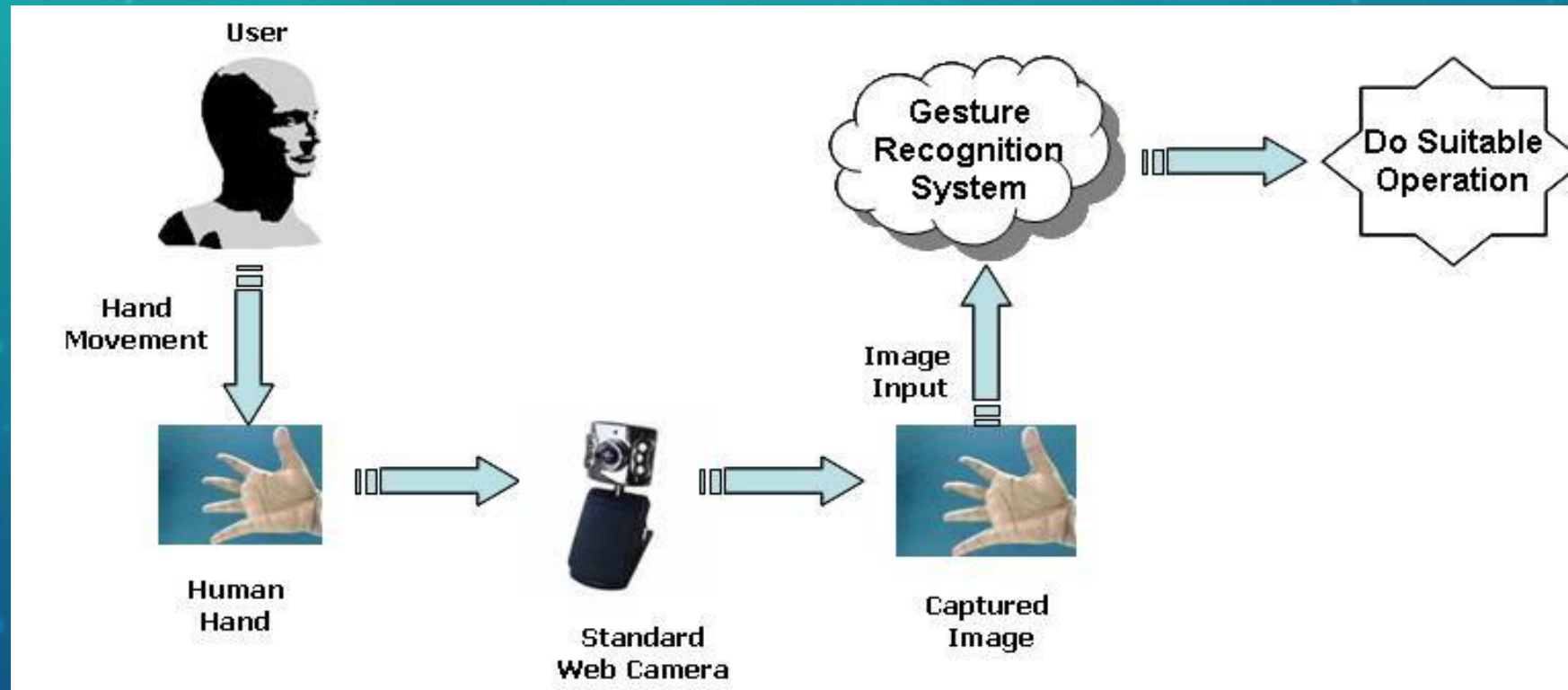
Hardware specification :

- Processor : intel core i3 and above
- Primary memory : 4 GB RAM and above
- Storage : 500GB hard disk and above
- camera

Software specification :

- Front End : python
- Operating system : windows 7 and above
- Back end : pyCharm

METHODOLOGY



In the Methodology, the method used in each component of the system will be explained separately. They are following subsections:

Camera setting

- ❑ The runtime operations are managed by the webcam of the connected laptop or desktop. To capture a video, we need to create a Video Capture object.
- ❑ Device index is just the number to specify which camera. Since we only use a single camera we pass it as '0'.
- ❑ After that can capture frame-by-frame.

Capturing frames

- ❑ The infinite loop is used so that the web camera captures the frames in every instance and is open during the entire course of the program.
- ❑ capture the live feed stream, frame by frame.

Display the frame

- ☐ A window will pop up on the screen of the user displaying the hands of the user and the subordinates lines controlling the cursor the output can be shown.

Mouse movement

- ☐ Calculate the difference between the fingers and the specific assigned value meet the the difference value then the mouse movement started.

Clicking

- ☐ The operation is performed by clicking the object and dragging it.

MEDIA PIPE

- ❑ MediaPipe is a framework which is used for applying in a machine learning pipeline, and it is an opensource framework of Google.
- ❑ The MediaPipe framework is useful for cross platform development.
- ❑ The MediaPipe framework is multimodal, where this framework can be applied to various audios and videos .
- ❑ The MediaPipe framework is used by the developer for building and analyzing the systems through graphs, and it also been used for developing the systems for the application purpose.

[MediaPipe](#) offers cross-platform, customizable ML solutions for live and streaming media.



End-to-End acceleration: Built-in fast ML inference and processing accelerated even on common hardware



Build once, deploy anywhere: Unified solution works across Android, iOS, desktop/cloud, web and IoT



Ready-to-use solutions: Cutting-edge ML solutions demonstrating full power of the framework

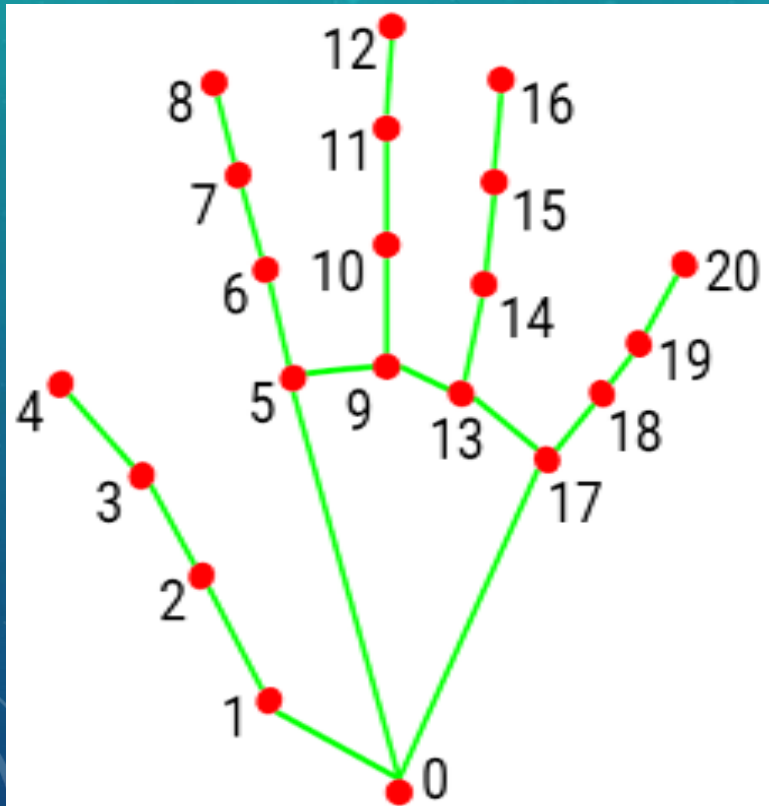


Free and open source: Framework and solutions both under Apache 2.0, fully extensible and customizable

CONCEPT OF HAND TRACKING

Hand tracking using MediaPipe involves two stages :

- ❑ **Palm detection** - MediaPipe works on the complete input image and provides a cropped image of the hand.
- ❑ **Hand landmarks identification** - MediaPipe finds the 21 hand landmarks on the cropped image of the hand.

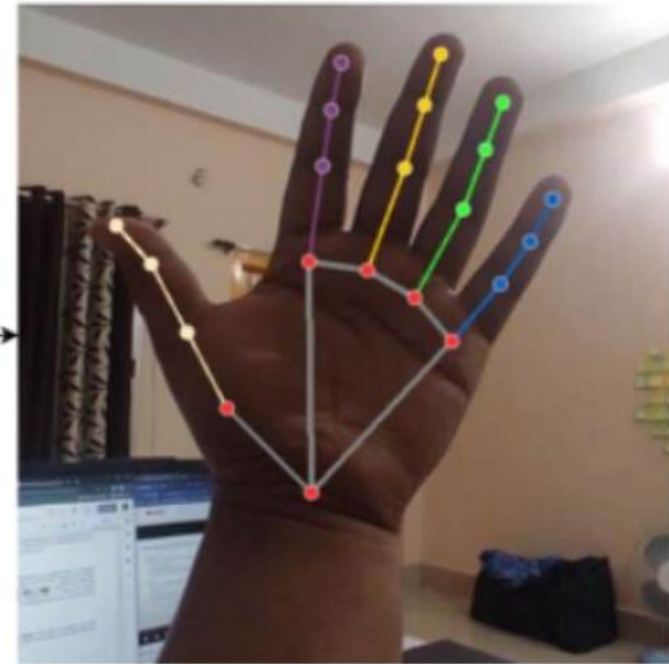


- | | |
|-----------------------|-----------------------|
| 0. WRIST | 11. MIDDLE_FINGER_DIP |
| 1. THUMB_CMC | 12. MIDDLE_FINGER_TIP |
| 2. THUMB_MCP | 13. RING_FINGER_MCP |
| 3. THUMB_IP | 14. RING_FINGER_PIP |
| 4. THUMB_TIP | 15. RING_FINGER_DIP |
| 5. INDEX_FINGER_MCP | 16. RING_FINGER_TIP |
| 6. INDEX_FINGER_PIP | 17. PINKY_MCP |
| 7. INDEX_FINGER_DIP | 18. PINKY_PIP |
| 8. INDEX_FINGER_TIP | 19. PINKY_DIP |
| 9. MIDDLE_FINGER_MCP | 20. PINKY_TIP |
| 10. MIDDLE_FINGER_PIP | |

HAND BEFORE AND AFTER



MP Hands



OPENCV

COMPUTER VISION

- ❑ Computer vision is a process by which we can understand the images and videos how they are stored and how we can manipulate and retrieve data from them.
- ❑ Computer Vision is the base or mostly used for Artificial Intelligence. Computer-Vision is playing a major role in self-driving cars, robotics as well as in photo correction apps

OPENCV

- ❑ OpenCV is the huge open-source library for the computer vision, machine learning, and image processing .
- ❑ .By using it, one can process images and videos to identify objects, faces, or even handwriting of a human.
- ❑ When it integrated with various libraries, such as ***NumPy, python*** is capable of processing the OpenCV array structure for analysis.
- ❑ To Identify image pattern and its various features we use vector space and perform mathematical operations on these features.

Steps to capture a video:

- Use `cv2.VideoCapture()` to get a video capture object for the camera.
- Set up an infinite while loop and use the `read()` method to read the frames using the above created object.
- Use `cv2.imshow()` method to show the frames in the video.
- Breaks the loop when the user clicks a specific key
- Frame Capture A frame is an image that forms a single instance of a video. A video consists of a lot of frames running per second .

Frame Capture

- import cv2 include opencv library functions in python.
- Create an object to hold reference to camera video capturing.
- `vidcap = cv2. VideoCapture(0)` • check if connection with camera is successfully.
- if `vidcap. isOpened()`:
 - `ret, frame = vidcap. read()` capture a frame from live video.
 - If the frame was successfully captured, and display the captured frame by using `cv2.imshow(windname, frame)`. The captured frame will be displayed in a new window

PROJECT PLAN

ID	Task Name	Start Date	End Date	Days	Status
1	Sprint 1	16/04/2022	28/04/2022	13	completed
2	Sprint 2	03/05/2022	15/05/2022	13	completed
3	Sprint 3	17/05/2022	30/05/2022	14	completed

PRODUCT BACKLOG

User story ID	Priority <High/Medium/Low>	Size (Hours)	Sprint <#>	Status <planned/in progress/ completed>	Release Date	Release Goal
1	Medium	13	1	Completed	30/4/2022	Camera configuration
2	High	15	2	Completed	16/5/2022	Hand recognition
3	High	20	3	Completed	30/5/2022	Gesture identification

SPRINT BACKLOG PLAN

SPRINT 1

Back log item	Statu s & comp letio n	Orig inal esti ate in hou rs	Day1 16/0 4	Day 2 17/0 4	Day 3 18/0 4	Day 4 19/0 4	Day 5 20/ 04	Day 6 21/0 4	Day 7 22/0 4	Day 8 23/0 4	Day 9 24/0 4	Day 10 25/0 4	Day 11 26/0 4	Day 12 27/0 4	Day 13 28/0 4	Day1 4 29/0 4
Cam era confi gurat ion	16/0 4/20 22	6	1	0	3	2	0	0	0	0	0	0	0	0	0	0
codi ng	21/0 4/20 22	12	0	0	0	0	3	2	0	3	0	1	3	0	0	0
testi ng	28/0 4/20 22	4	0	0	0	0	0	0	0	0	0	0	0	2	2	0
total		22	1	0	3	2	3	2	0	3	0	1	3	2	2	0

SPRINT BACKLOG PLAN

SPRINT 2

Bac klo g ite m	Statu s & comp letio n	Origi nal esti mate in hour s	Day 1 03/ 05	Day 2 04/ 05	Day 3 05/ 05	Day 4 06/ 05	Day 5 07/ 05	Day 6 08/ 05	Day 7 09/ 05	Day 8 10/ 05	Day 9 11/ 05	Day 10 12/ 05	Day 11 13/ 05	Day 12 14/ 05	Day 13 15/ 05	Day 14 16/ 05
Cod ing	3/5/2 022	10	0	0	0	0	2	1	0	3	1	1	2	0	0	0
Test ing	15/5/ 2022	4	0	0	0	0	0	0	0	0	0	0	0	2	2	0
Tot al		14	0	0	0	0	2	1	0	3	1	1	2	2	2	0

SPRINT BACKLOG PLAN

SPRINT 3

Backlog item	Status & completion	Original estimate in hour	Day 1 17/05	Day 2 18/05	Day 3 19/05	Day 4 20/05	Day 5 21/05	Day 6 22/05	Day 7 23/05	Day 8 24/05	Day 9 25/05	Day 10 26/05	Day 11 27/05	Day 12 28/05	Day 13 29/05	Day 14 30/05
coding	17/5/2022	15	2	3	0	1	1	3	0	3	1	1	2	0	0	0
Testing	30/5/2022	5	0	0	0	0	0	0	0	0	0	0	0	2	2	1
total		20	2	3	0	1	1	3	0	3	1	1	2	2	2	1

SPRINT BACKLOG ACTUAL

SPRINT 1

Back log item	Stat us & completi on	Origi nal estia te in hour s	Day1 16/0 4	Day 2 17/0 4	Day 3 18/0 4	Day 4 19/0 4	Day 5 20/0 4	Day 6 21/0 4	Day 7 22/0 4	Day 8 23/0 4	Day 9 24/0 4	Day 10 25/0 4	Day 11 26/0 4	Day 12 27/0 4	Day 13 28/0 4	Day1 4 29/0 4
Cam era confi gurat ion	16/0 4/20 22	6	1	0	3	2	0	0	0	0	0	0	0	0	0	0
codi ng	21/0 4/20 22	12	0	0	0	0	3	2	0	3	0	1	3	0	0	0
testi ng	28/0 4/20 22	4	0	0	0	0	0	0	0	0	0	0	0	2	2	0
total		22	1	0	3	2	3	2	0	3	0	1	3	2	2	0

SPRINT BACKLOG ACTUAL

SPRINT 2

Back log item	Stat us & com pleti on	Orig inal esti mat e in hou rs	Day 1 03/ 05	Day 2 04/ 05	Day 3 05/ 05	Day 4 06/ 05	Day 5 07/ 05	Day 6 08/ 05	Day 7 09/ 05	Day 8 10/ 05	Day 9 11/ 05	Day 10 12/ 05	Day 11 13/ 05	Day 12 14/ 05	Day 13 15/ 05	Day 14 16/ 05
Codi ng	3/5/ 202 2	10	0	0	0	0	2	1	0	3	1	1	2	0	0	0
Testi ng	15/ 5/2 022	4	0	0	0	0	0	0	0	0	0	0	0	2	2	0
Total		14	0	0	0	0	2	1	0	3	1	1	2	2	2	0

SPRINT BACKLOG ACTUAL

SPRINT 3

Back log item	Stat us & completi on	Original esti mat e in hour	Day 1 17/05	Day 2 18/05	Day 3 19/05	Day 4 20/05	Day 5 21/05	Day 6 22/05	Day 7 23/05	Day 8 24/05	Day 9 25/05	Day 10 26/05	Day 11 27/05	Day 12 28/05	Day 13 29/05	Day 14 30/05
codi ng	17/5 /202 2	15	2	3	0	1	1	3	0	3	1	1	2	0	0	0
Testi ng	30/5 /202 2	5	0	0	0	0	0	0	0	0	0	0	0	2	2	1
total		20	2	3	0	1	1	3	0	3	1	1	2	2	2	1



THANK YOU