Finger Recognition and Gesture Based Augmented Keyboard and Related Applications

Fathimath Suhara M A
MES20MCA-2018
Product Owner: Nowshad C V

TABLE OF CONTENTS

INTRODUCTION

* In this project, we have tried to reduce the gap between the real world and the augmented environment to produce a mixed reality system.

* For that purpose, we created a virtually controllable keyboard system ,pong game,virtual painter which is created and implemented using OpenCV libraries and python3.2.1.

* To provide an easy immersive augmented experience which is also gesture enabled, we employ a web camera which is integrated with OpenCV libraries through a compiler.

* Using the concept of gesture recognition, it is possible to point a finger at the computer screen or web camera so that the keypad will be pressed accordingly to form meaningful sentences or words.

* Gesture recognition is an active research field in Human-Computer Interaction technology. It has many applications in virtual environment control and sign language translation, robot control, or music creation.

* In this machine learning project on Hand Gesture Recognition, we are going to make a real-time Hand Gesture Recognizer using the MediaPipe framework.

* OpenCV is a real-time Computer vision and image-processing framework built on C/C++. But we'll use it on python via the OpenCV-python package.

* Depending on the type of the input data or source, the approach for interpreting a fingertip could be done in different ways,most of the techniques rely on key pointers represented in a 3D coordinate system.

* Based on the relative motion of these, the gesture can be detected with a high accuracy, depending on the quality of the input and the algorithm's approach.

AI VIRTUAL KEYBOARD

Using our system, users can control a virtual keyboard using their finger movements and finger tips. Further, users can communicate with people who are viewing the screen, the user selects an alphabet with their fingertip and can move the keyboard with the help of hand gesture.

This describes the way of implementing a virtual keyboard without any additional hardware but by using the webcam available in the system. The webcam simply captures the consecutive frames and compares them to recognize it as a click if there is a difference in the contour.

Various on-screen virtual keyboards are available but it is difficult to accommodate a full-sized keyboard on the screen as it creates hindrance to see the documents being typed.

PONG GAME

Pong is one of the earliest arcade video games. It is a table-tennis inspired game featuring simple two-dimensional graphics. In it, the player controls the paddle by moving it vertically across the left or the right side of the screen. The objective is to reach 21 points before the opponent; each player earns points when the opponent fails to return the ball. The aim of this project is to create a python-based application for Pong using the Pygame library. It also involves creating a few AI models which the user can play against.Here we can control the game by using the finger movements.

AI VIRTUAL PAINTER

Drawing or Sketching using hand is everyone's wish. Some or the other time we imagine writing in air using our hand. So, here came the project from this concept where we create a canvas and pick the colours required using our hand and draw the required design or write anything you wish.

Virtual Sketch is in where we can draw by just capturing the motion of a colored marker with a camera. One colored object at the tip of the finger is mainly used as the marker.

We are here now, using the techniques of computer vision in OpenCV to build this project. The required language for this project is python due to its more exhaustive libraries and easy to make use of the syntax and but understanding the basics as well as it can be implemented in any OpenCV supported languages The colour tracking and detection processes are used to achieve the goal of this project.

Given the real time webcam data, this paint-like python application uses OpenCV library to track an object-of-interest (a bottle cap in this case) and allows the user to draw by moving the object, which makes it both awesome and challenging to draw simple things.

DEVELOPING ENVIRONMENT

Hardware specification:

- Processor : Intel Pentium Core i5 and above
- Primary Memory : 4 GB RAM and above
- Storage : 500 GB hard disk and above
- Display : VGA Colour Monitor
- Key Board : Windows compatible
- Mouse : Windows compatible

Software specification:

- Front end : Python
- Operating system : windows 7 and above
- IDE : PyCharm

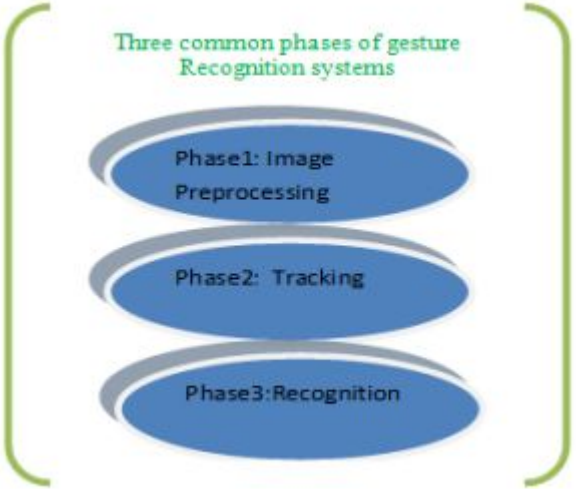
METHODOLOGY

Gesture recognition is the process by which the gestures made by the user are recognized by the receiver. Gestures are expressive, meaningful body motions involving physical movements of the fingers, hands, arms, head, face, or body with the intent of:

1) conveying meaningful information or

2) interacting with the environment.

Normally gesture recognition systems are divided into three stages namely image processing followed by tracking succeeded finally by the recognition stage as depicted in the figure. Generally, there exist many-to-one mappings from concepts to gestures and vice versa. Hence, gestures are ambiguous and incompletely specified.



Gesture recognition is a topic pursued with the goal of interpreting human motions via mathematical algorithms. Gesture recognition can be seen as a way for a computer to begin to understand human body language, thus building a richer bridge between machines and humans than primitive text interface or even graphical user interfaces (GUI), which

still limit the majority of input to keyboard and mouse. Hence, gesture recognition is a process by which the system will know, what is going to be performed by the gesturer. Gesture recognition can be conducted with techniques from computer vision and image processing. Gesture recognition enables humans to interface with the machine and interact naturally without any mechanical devices. Using the concept of gesture recognition, it is possible to point a finger at the computer screen so that the cursor will move accordingly. This could potentially make conventional input devices such as mouse, keyboards and even touch-screens redundant.

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

target representation and localization,

filtering and data association.

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

FINGERTIP DETECTION ALGORITHM

HAND MOVEMENT

WEB CAM

REAL TIME INPUT HAND IMAGE

PRE-PROCESSING

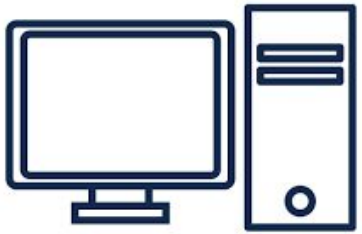
HAND DETECTION AND LANDMARKING

TRACKING OF MOVING HAND REGION

FINGERTIP DETECTION

RECOGNIZED MEANINGFUL HAND GESTURE

CORRESPONDING SYSTEM COMMANDS

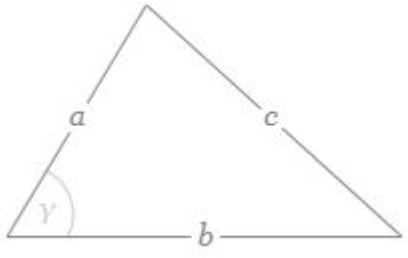


SYSTEM TO BE CONTROLLED

Mathematics used

Cosine Theorem

In trigonometry, the law of cosines relates the lengths of the sides of a triangle to the cosine of one of its angles. Using notation as in Figure the law of cosines states where γ denotes the angle contained between sides of lengths a and b and opposite the side of length c.



Formula used

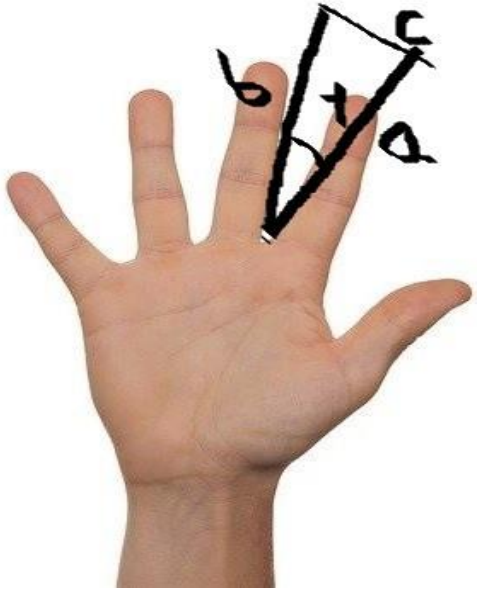
$$c = \sqrt{a^2 + b^2 - 2ab\cos\gamma}$$

By seeing this formula now we understand that if we have; *a,b* and *gama* then we also find *c* as well as if we have; *a,b,c* then we also find *gamma (vice-versa)*

For finding gamma this formula is used:

$$\gamma = \cos^{-1}\left(\frac{a^2 + b^2 - c^2}{2ab}\right)$$

Using Cosine theorem to recognize fingers



draw a Side: *a,b,c* and angle: *gamma.* Now this gamma is always less than 90 degree, So we can say: If *gamma* is less than 90 degree or *pi/2* we consider it as a finger

What is MediaPipe?

MediaPipe is a customizable machine learning solutions framework developed by Google. It is an open-source and cross-platform framework, and it is very lightweight. MediaPipe comes with some pre-trained ML solutions such as face detection, pose estimation, hand recognition, object detection, etc.

Mediapipe is a Framework for building machine learning pipelines for processing time-series data like video, audio, etc. This cross-platform Framework works in Desktop/Server, Android, iOS, and embedded devices like Raspberry Pi and Jetson Nano.

MediaPipe offers cross-platform, customizable ML solutions for live and streaming media.

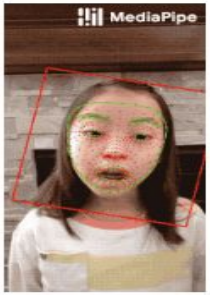
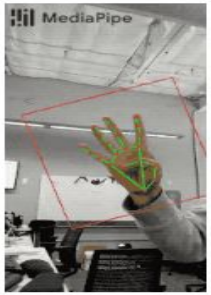
End-to-End acceleration: *Built-in fast ML inference and processing accelerated even on common hardware*

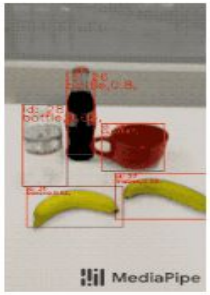
Build once, deploy anywhere: *Unified solution works across Android, iOS, desktop/cloud, web and IoT*

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*

ML solutions in MediaPipe

Face Detection	Face Mesh	Iris	Hands	Pose	Holistic
					

Hair Segmentation	Object Detection	Box Tracking	Instant Motion Tracking	Objectron	KNIFT
					

Uses of MediaPipe

Every Youtube video we watch is processed with machine learning models using MediaPipe. Google has not hired thousands of employees to watch every video people upload, because thousands of people are not enough to look after and check each published video, the amount of data Google gets daily is not easy for humans to check.

Machine Learning models are developed to make our life easier, so tasks that are hard for us to complete, machine learning and deep learning models help us to do in less amount of time, on the other hand, we can save money by not hiring employees.

Yes, Google has machine learning/deep learning models to see if the videos match their policies and the content is not having copy-right issues.

Basically, MediaPipe is a framework for Computer Vision and Deep Learning that builds perception pipelines. For now, you just need to know, perception pipelines are some sort of audio, video, or time-series data that catch the process in pipelining zone.

Some of the major applications of MediaPipe:

- Multi-hand Tracking
- Face Detection
- Objective: 3D Object Detection and Tracking
- AutoFlip: Automatic video cropping pipeline etc.

Algorithm used in MediaPipe

Google researchers have unveiled a new real-time hand tracking algorithm that could be a new breakthrough for people communicating via sign language.

Their algorithm uses machine learning to compute 3D keypoints of a hand from a video frame. This research is implemented in MediaPipe which is an open-source cross-platform framework for building multimodal (eg. video, audio, any time series data) applied ML pipelines.

What is interesting is that the 3D hand perception can be viewed in real-time on a mobile phone human computer

How real-time hand perception and gesture recognition works with MediaPipe?

The algorithm is built using the MediaPipe framework. Within this framework, the pipeline is built as a directed graph of modular components.

The pipeline employs three different models:

- a palm detector model,
- a handmark detector model
- gesture recognizer.

Palm detector model operates on full images and outputs an oriented bounding box. They employ a single-shot detector model called BlazePalm, They achieve an average precision of 95.7% in palm detection.

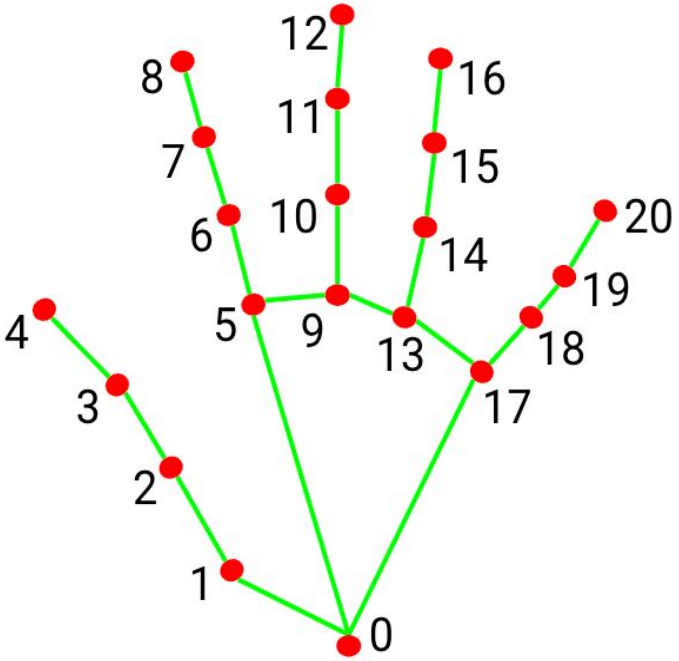
Handmark detector model takes the cropped image defined by the palm detector and returns 3D hand keypoints. For detecting key points on the palm images, researchers manually annotated around 30K real-world images with 21 coordinates. They also generated a synthetic dataset to improve the robustness of the hand landmark detection model.

Gesture recognizer then classifies the previously computed keypoint configuration into a discrete set of gestures. The algorithm determines the state of each finger, e.g. bent or straight, by the accumulated angles of joints.

Concept of Hand Tracking

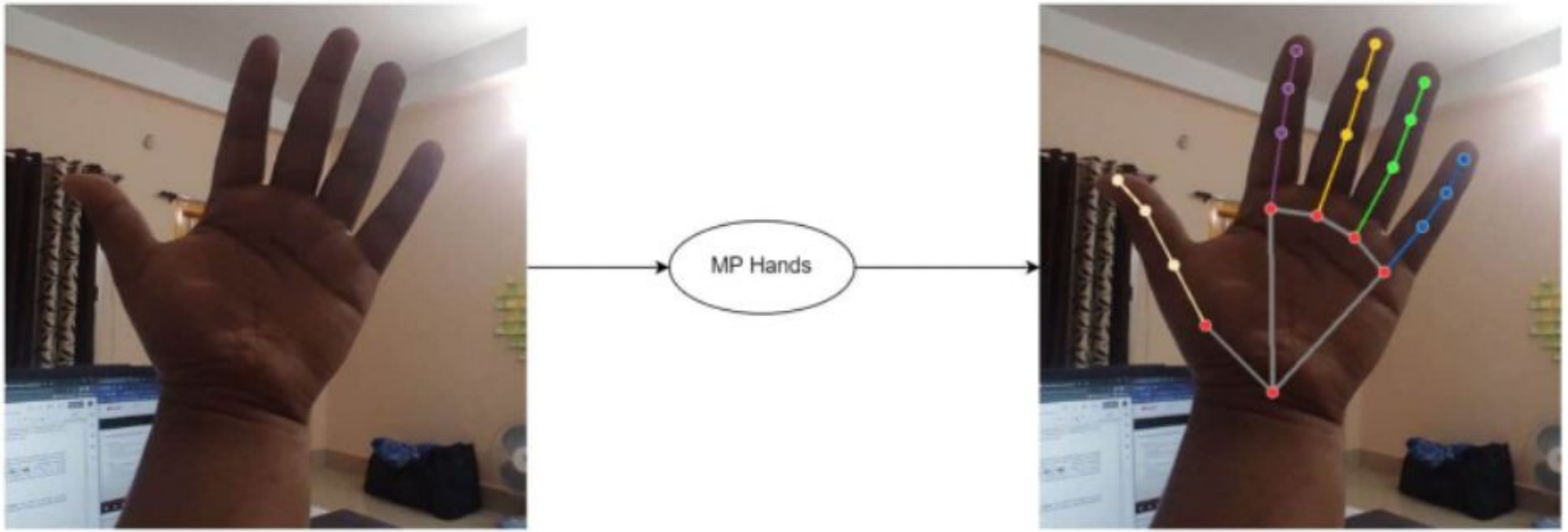
Before we jump into coding, let us discuss how MediaPipe performs hand tracking. Hand tracking using MediaPipe involves two stages:

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

HAND BEFORE AND AFTER



Working of AI Virtual Painter

Features :

1. Can draw on your system screen based on your index finger movement;
2. Can track your hand in real time.

Working :

1. This project is a use case of Hand Tracking Technology.
2. As soon as the user shows up his hand in the camera the application detects it and draws a bounding box around the hand.
3. If user shows only Index finger then he/she is in drawing mode.
4. To select different color or eraser from the top of canvas,user must select it by taking his both Index and Middle finger together at the top of icon.

CONCEPT OF HAND DETECTION

Here we can find the distance between the top point of our index finger and middle finger, if the distance between the two is less than a certain threshold, then we can type the letter on which we are indicating. Once we get the position then we loop through the entire position list. From that list, we find button position and button size and then we plot it on the frame according to a well-defined manner.

Here we can find the distance between the top point of our index finger and middle finger. If the distance between the two is less than a certain threshold, then we can type the letter on which we are indicating.

After that, we need to find the distance between the top point of our index finger and middle finger. In the above image, you can see the top points which we require are point 8 and point 12.

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 and now it plays a major role in real-time operation which is very important in today's systems. 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.

The first OpenCV version was 1.0. OpenCV is released under a BSD license and hence it’s free for both academic and commercial use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. When OpenCV was designed the main focus was real-time applications for computational efficiency. All things are written in optimized C/C++ to take advantage of multi-core processing.

Applications of OpenCV:

There are lots of applications which are solved using OpenCV, some of them are listed below:

- face recognition
- Automated inspection and surveillance
- number of people – count (foot traffic in a mall, etc)
- Vehicle counting on highways along with their speeds
- Interactive art installations

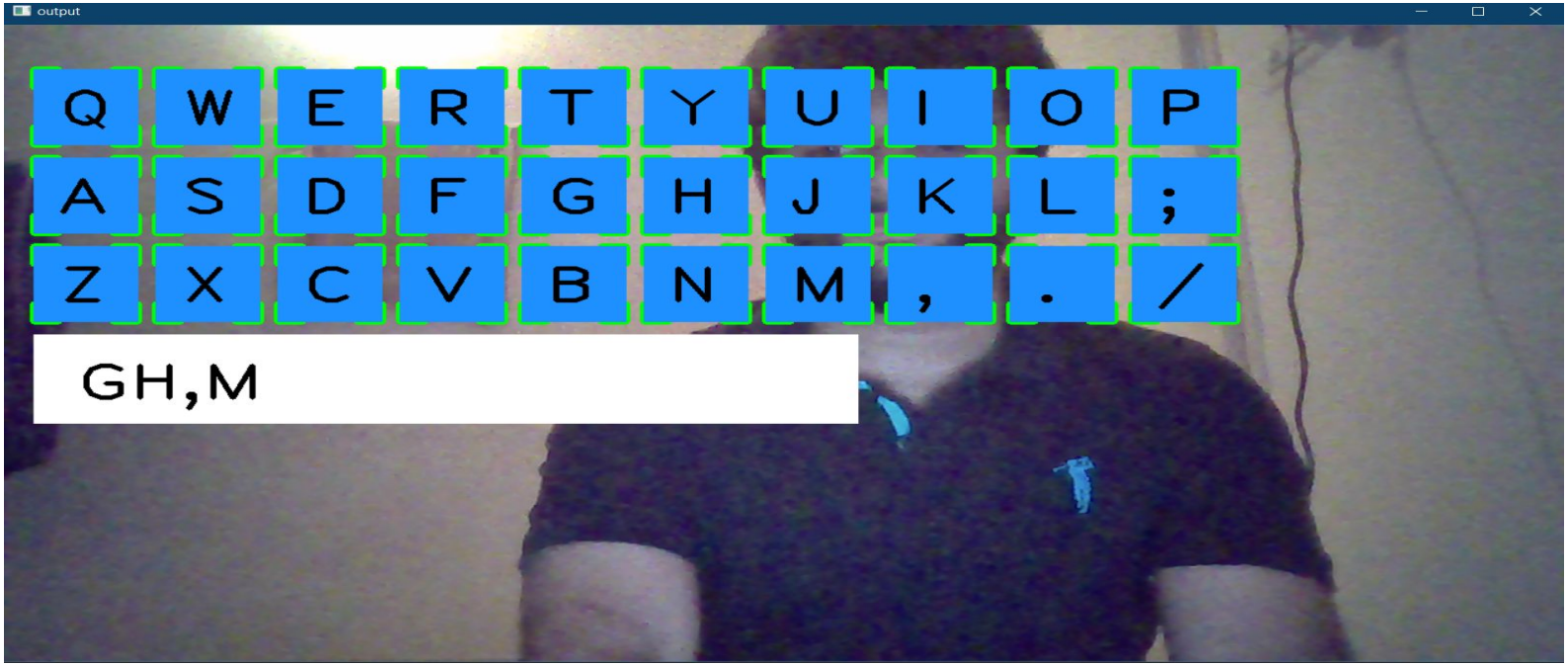
- Anomoly (defect) detection in the manufacturing process (the odd defective products)
- Street view image stitching
- Video/image search and retrieval
- Robot and driver-less car navigation and control
- object recognition
- Medical image analysis
- Movies – 3D structure from motion
- TV Channels advertisement recognition

OpenCV Functionality

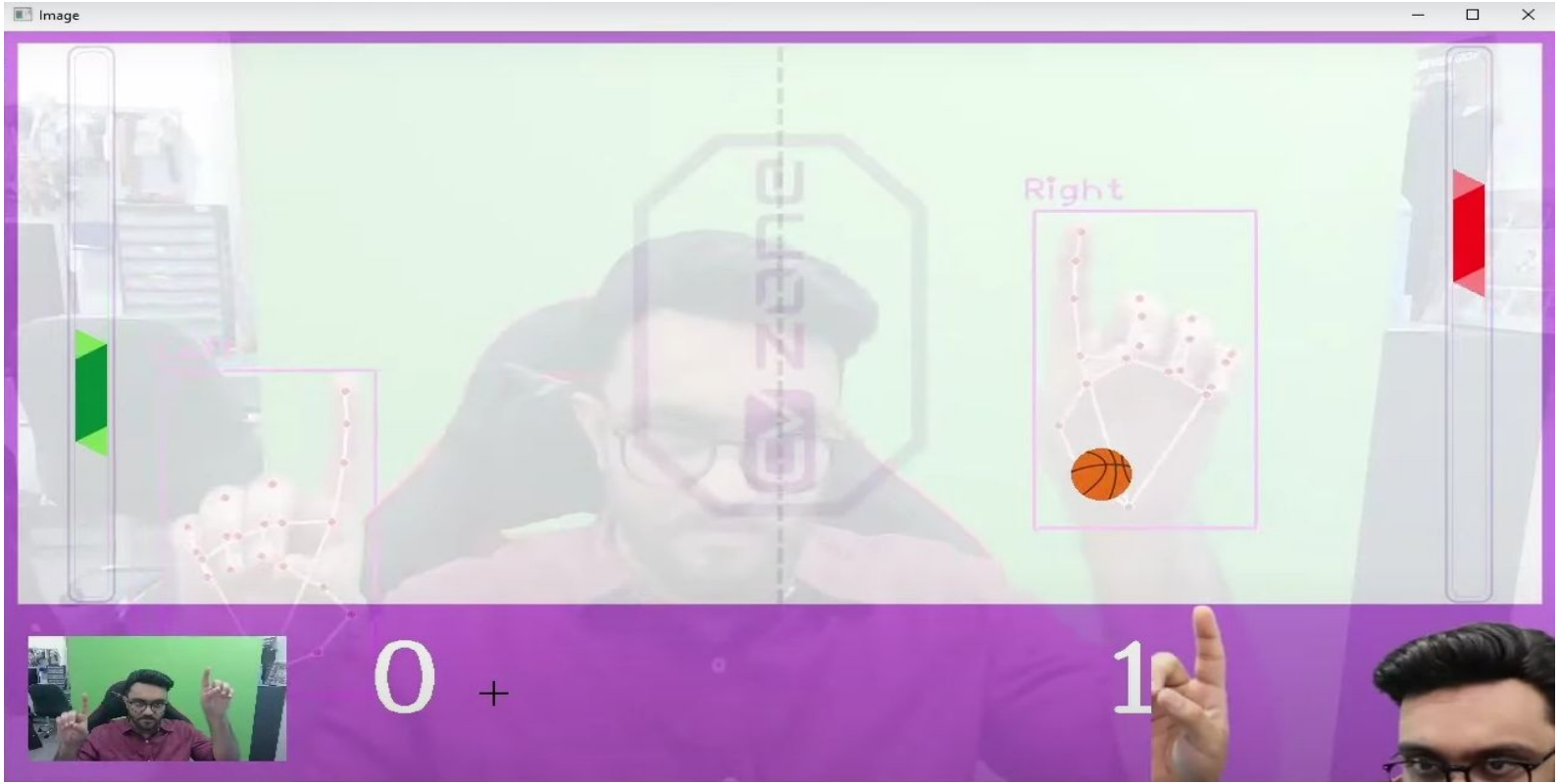
- Image/video I/O, processing, display (core, imgproc, highgui)
- Object/feature detection (objdetect, features2d, nonfree)
- Geometry-based monocular or stereo computer vision (calib3d, stitching, videostab)
- Computational photography (photo, video, superres)
- Machine learning & clustering (ml, flann)
- CUDA acceleration (gpu)

RESULTS

AI VIRTUAL KEYBOARD



PONG GAME



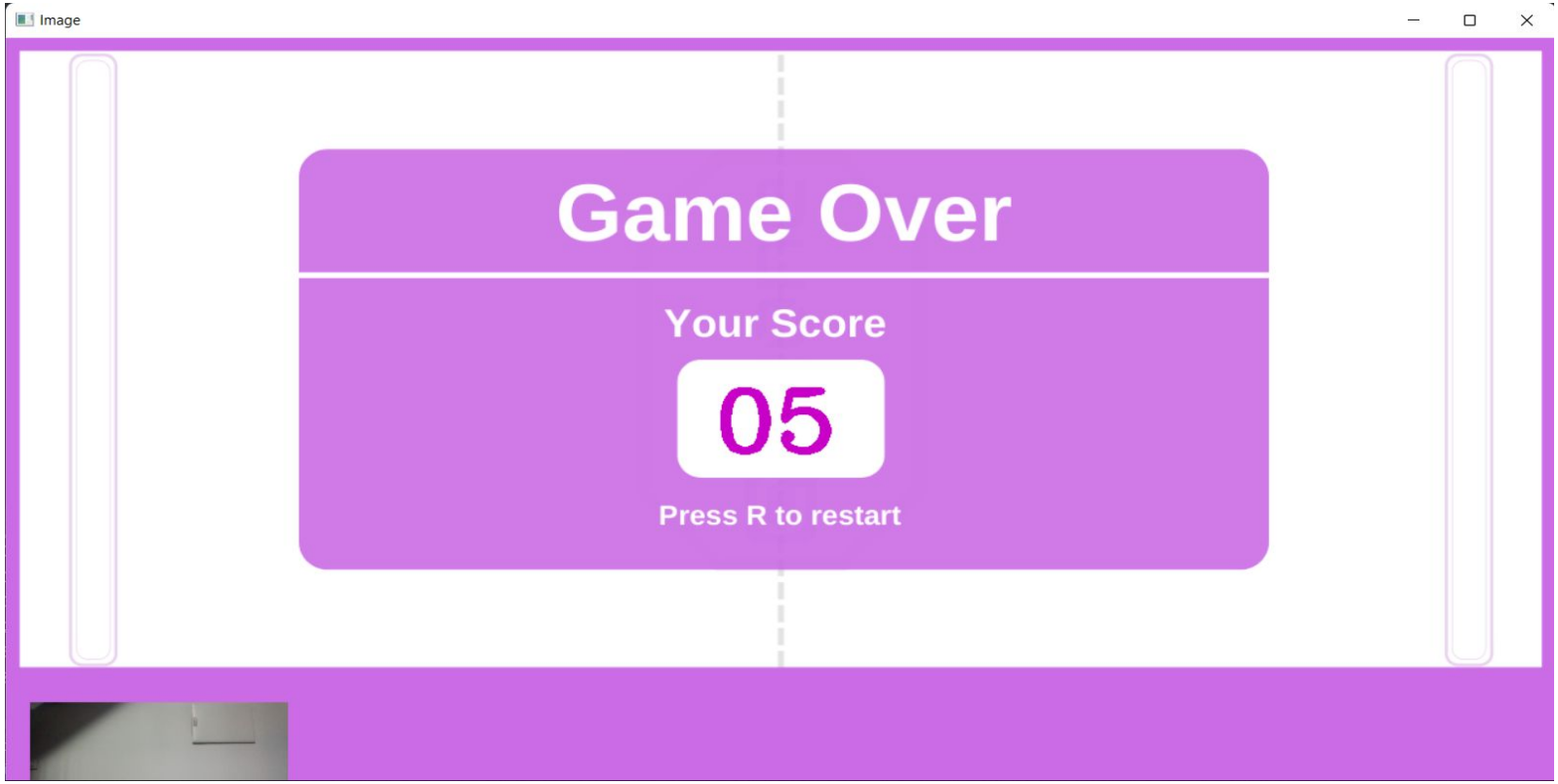
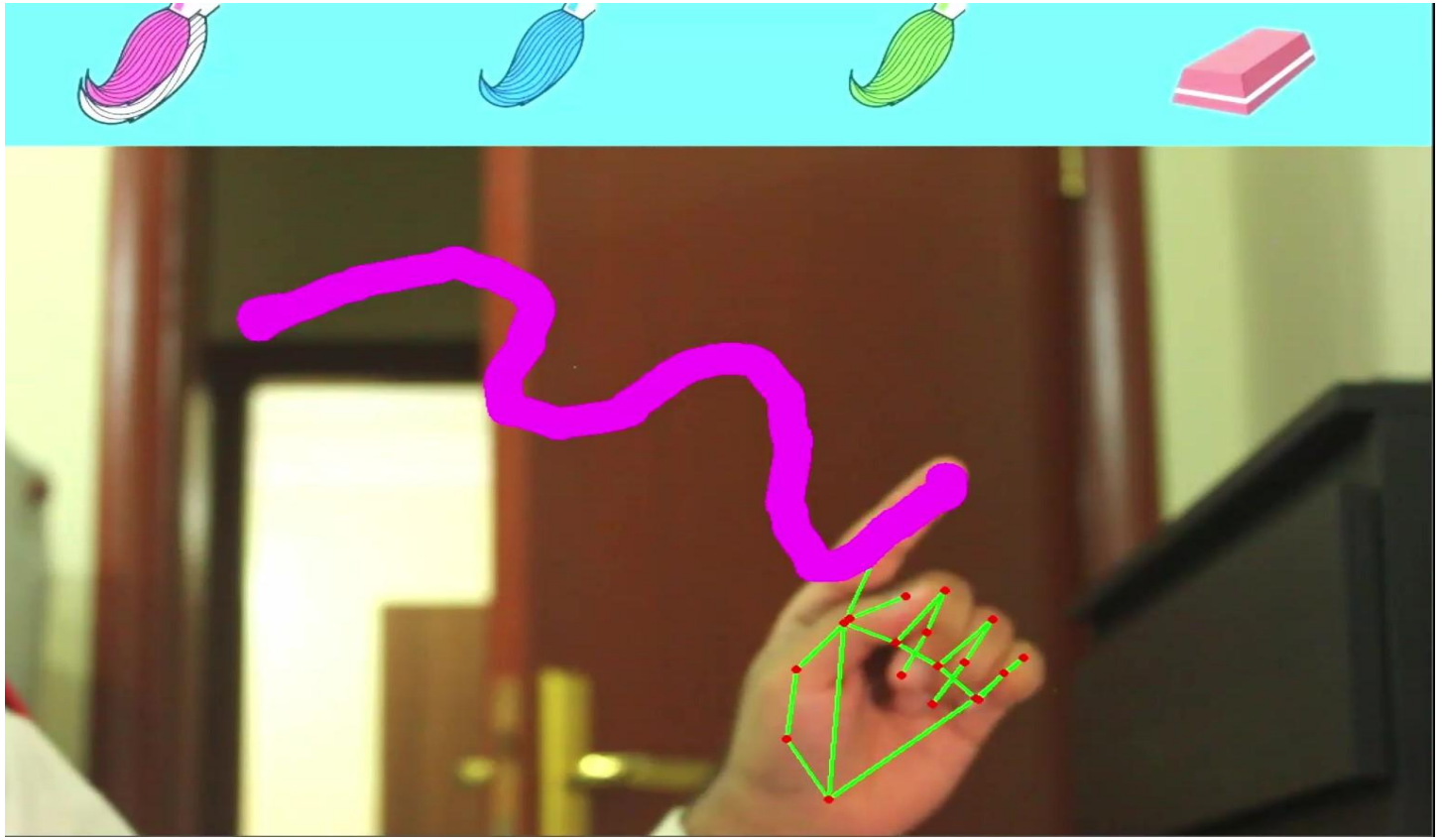


Image
Game Over
Your Score
05
Press R to restart

AI VIRTUAL PAINTER



FUTURE ENHANCEMENT

1. AI VIRTUAL KEYBOARD

For the future work, swipe keypads which detect the gestures in air view can also be implemented. This would improve the results when the typing speed is quick. The same key-pad can be made multi-lingual by just changing the keys on the keypad. Also, this technique can be further improvised to be used on a smart TV.

Covid-19 create chaos in society. So there was restriction in touching different gadgets in shops. So implementing such initiative will protect people contact to infection.

Since the project is a small prototype, this is able to done in great way. Since many concepts are being used by big companies like lenskart, reliance trends to satisfy consumers.

2. PONG GAME

This simple game later can be used to build exciting game.

This game further improved and have wide range of possibility

This include adding

ˆ Mobile Application

ˆ Data base

ˆ Online players ..etc

3. AI VIRTUAL PAINTER

- To ensure that, the interface is very simple and easily understandable by the user.
- The user should be able to draw what he wishes to draw without any interruptions.
- In future, this is useful for making kids to learn drawing in schools in an interactive way
- Teacher in school will be able to teach student in more interactive and exciting manner.

PROJECT PLAN

User story ID	Task Name	Start Date	End Date	Days	Status
1	Sprint 1	09/04/2022	25/04/2022	12	Completed
2	Sprint 2	28/04/2022	19/05/2022	14	Completed
3	Sprint 3	21/05/2022	04/06/2022	7	Completed

PRODUCT BACKLOG

User story ID	Priority <High/Medium/Low>	Size (Hours)	Sprint <#>	Status <Planned/In progress/Completed>	Release Date	Release Goal
1	Medium	12	1	Completed	25/04/2022	AI Virtual Keyboard
2	Medium	10	2	Completed	19/05/2022	Pong Game
3	High	14	3	Completed	04/06/2022	AI Virtual Painter

SPRINT BACKLOG PLAN

Backlog items	status & completion	Original Estimate in hours	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
Camera configuration	25/4/2022	6	1	0	3	2	0	0	0	0	0	0	0	0	0	0
Coding	28/4/2022	12	0	0	0	0	3	2	0	3	0	1	3	0	0	0
Testing	4/6/2022	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

Backlog items	status & completion	Original Estimate in hours	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14	Completed <Y/N>
Camera configuration		6	1	0	3	2	0	0	0	0	0	0	0	0	0	0	Y
Coding		12	0	0	0	0	3	2	0	3	0	1	3	0	0	0	Y
Testing		4	0	0	0	0	0	0	0	0	0	0	0	2	2	0	Y
Total		22	1	0	3	2	3	2	0	3	0	1	3	2	2	0	Y

THANK YOU