AIR CANVAS

PROF.MUSILEK, Professor of CSE, St.Peter's Engineering College, Maisammaguda, Hyderabad, TS,India.

Abstract: In today's world online education is the new challenge and to make teaching much easier and effective there must be some handy tools for the work to go by ease. Air canvas is one such tool which can be used to make e-learning effective. It is a hands-free digital drawing canvas that utilizes a Camera and OpenCV to recognize and map hand gestures onto a screen. The user's brush can be modified in size and color by using built-in buttons. The direction of the brush is controlled completely using open source OpenCV software and modified to map the pointer finger onto the screen following a calibration screen to measure and record the color of the user's hand. The idea for Air Canvas is a result of developing handy tools in digital drawing and smart photo recognition software. To create a simple prototype for a drawing tool that uses hand gesture recognition software to paint on a screen. Core objectives includes OpenCV to recognize the pointer finger. Mapping coordinates extracted from hand recognition software to produce a drawing. Implementing additional features such as color change, size change on-screen.
Keywords: Air canvas, OpenCV, Computer Vision, Media Pipe

1. INTRODUCTION

1.1 Motivation

The initial motivation was a need for a dustless class room for the students to study in. There are many ways like touch screens and more but what about the schools which can't afford to buy such huge, large screens and teach on them like a T.V. So, A thought why not a finger be tracked, but that too at an initial level without deep learning. In the era of digital world, traditional art of writing is being replaced by digital board. Social development is not a people's will, but the needs of human life are the main driving force anyway. The same situation happens in education. In the present circumstances, digital canvas and traditional methods are inclusive of the symbiotic state, so this needs to systematically understand the basic knowledge of the form between digital canvas and traditional methods. The traditional way includes pen and paper, chalk and board method of writing. The essential aim of digital canvas is of building hand gesture recognition system to write digitally. Smart air canvas includes many ways of writing like by using keyboard, touch-screen surface, digital pen, using electronic hand gloves, etc. But in this system, it uses hand gesture recognition with the use of machine learning algorithm by using python programming, which creates natural interaction between man and machine. Air canvas helps to draw on a screen just by waving a finger fitted with a colorful point or a simple colored cap. Using the computer vision techniques of OpenCV to build this paper. The preferred language is python due to its exhaustive libraries and easy to use syntax but understanding the basics it can be implemented in any OpenCV supported language. Hand detection and gesture recognition has a broad range of potential applications, including in-car gestures, sign language recognition, virtual reality and so on. Through gestures, users can control or interact with devices without touching them.

1.2 Problem Definition

To develop An AI based tool using techniques of OpenCV which can draw anything on any surface by just capturing the motion of a hand's position with a camera. Here a finger's tip is used as the marker. Ever thought, waiving the finger into the air can draw on a real canvas. How this air canvas in Computer Vision Projects works. The pandemic has resulted in schools shut all across the world. As a result, education has changed dramatically, with the distinctive rise of e-learning, whereby teaching is undertaken remotely and on digital platforms. Since it has drastically changed towards the digital platform initially the teachers faced many challenges to delegate their lectures, by either placing their boards, books or by sharing the screen which was not very effective and nor did the students understand the concepts clearly as reported by a survey conducted.

1.3 Objective

The main objective of this paper is to simply create a prototype for a drawing tool that uses hand gesture recognition software to draw on a screen. Core objectives of this project includes:

- To detect hands and track fingers
- Functionalities to draw and erase
- Include different colors and shapes

1.4 Limitations

- Whatever has been written on the Air canvas cannot be saved and prepared into a document for future reference.
- OCR cannot take images that include textual elements and attempts to recognize that text.
- The experiment couldn't be done with different interpolation methods such as PyGame includes a line drawing method (PyGame draw.line()) that could prove useful in producing smoother and cleaner lines.

The organization of this paper is in Section1 has introduction., where it consists of motivation, problem definition, objectives and limitations. Section 2 has literature survey. It consists of existing system, disadvantages of existing system and proposed system. Section 3 architecture design., it represents the workflow of the project and also explains about the major algorithms involved. Section 4 has Module Design. It consists of all the modules that are there in the process. Here, palm detection model and hand landmark detection are elaborated. Section 5 has implementation and results. Here, key functions and methods of implementation are elaborated. Section 6 is about testing and validation. Testing represents about the accuracy. Validation gives us idea about the results and variation between proposed system and existing systems. Chapter 7 demonstrates the conclusions and future enhancements.

2. LITERATURE SURVEY AND RESEARCH BACK GROUND

2.1 Introduction

MediaPipe approach provides high-fidelity hand and finger tracking by employing machine learning (ML) to infer 21 3D key points of a hand from just a single frame. To detect initial hand locations, it employs a single-shot detector model called Blaze Palm, optimized for mobile real-time uses in a manner similar to Blaze Face, which is also available in MediaPipe. After the palm detection over the whole image a subsequent hand landmark model performs precise key point localization of 21 3D hand-knuckle coordinates inside the detected hand regions via regression, that is direct coordinate prediction. The model learns a consistent internal hand pose representation and is robust even to partially visible hands and self-occlusions.

MediaPipe Hands utilizes an ML pipeline consisting of multiple models working together: A palm detection model that operates on the full image and returns an oriented hand bounding box. A hand landmark model that operates on the cropped image region defined by the palm detector and returns high-fidelity 3D hand key points. This strategy is similar to that employed in MediaPipe Face Mesh solution, which uses a face detector together with a face landmark model.

OpenCV-Python is a library of Python bindings designed to solve computer vision problems. OpenCV is the huge open-space library for 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 is 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 it uses vector space and perform mathematical operations on these features.

OpenCV is the huge open-space 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 12 structure for analysis. To Identify image pattern and its various features it uses vector space and perform mathematical operations on these features.

NumPy enriches the programming language Python with powerful data structures, implementing multi-dimensional arrays and matrices. These data structures guarantee efficient calculations with matrices and arrays. The implementation is even aiming at huge matrices and arrays, better known under the heading of "big data". Besides, the module supplies a large library of high-level mathematical functions to operate on these matrices and arrays.

Python bindings of the widely used computer vision library OpenCV utilize NumPy arrays to store and operate on data. Since images with multiple channels are simply represented as three-dimensional arrays, indexing, slicing or masking with other arrays are very efficient ways to access specific pixels of an image. The NumPy array as a universal data structure in OpenCV for images, extracted feature points, filter kernels and many more vastly simplifies the programming workflow and debugging.

2.2 Existing System

In the era of the digital world, traditional art of writing is being replaced by digital art. Digital art refers to forms of expression and transmission of art form with digital form. Relying on modern science and technology is the distinctive characteristics of the digital manifestation.

Traditional art refers to the art form which is created before digital art. Digital art and traditional art are interrelated and interdependent. Social development is not a people's will, but the needs of human life are the main driving force anyway. The traditional way includes pen and paper, chalk and a board method of writing. But that doesn't support in the growing world.

So, digital art has been introduced. Digital art includes many ways of writing like by using keyboard, touch-screen surface, digital pen, stylus, using electronic hand gloves, etc. All the available options have its own merits and de-merits. And here, this project is concentrating on digital pen. Contour detection and object detection are the existing systems already.

In contour detection, queue is implemented to trace all the points to draw on canvas. In object detection, tracking mechanism is implemented to draw on canvas. But in this system, it uses hand gesture recognition with python programming, to write on screen.

2.3 Disadvantages of Existing System

- Traditional teaching methods cannot be used in online teaching.
- Digital art equipment's like writing pads are expensive.
- It becomes difficult to explain diagrams using the mouse or the keypad.

2.4 Proposed System

The essential aim of the proposed system is to build a hand gesture recognition system to write digitally. Digital art includes many ways of writing like by using keyboard, touch-screen surface, digital pen, stylus, using electronic hand gloves, etc. But in this system, it uses hand gesture recognition with the use of python programming, which creates natural interaction between man and machine. With the advancement in technology, the need for development of natural human – computer interaction systems to replace traditional systems is increasing rapidly. This tool allows the user to track the movement of any colored object of his choice. The user can even choose the colors of his choice to be displayed. By running the application, the camera is activated thus enabling the user to draw in the air just by waving the tracker object. The drawing is also simultaneously visible on the white window. He/She can choose any color of his/her choice displayed to draw and also can clear the screen when needed. It will be using the computer vision techniques of OpenCV to build this application.

3. ANALYSIS

The system starts by taking a video input from a webcam. The identified image is then processed by using media pipe and landmarks are mapped to a hand which further helps in tracking the exact position of all the fingers by coordinating all the points in the landmarks. There are two major modes defined in this model named selection mode and drawing mode. Selection mode helps to select different colors and shapes available on the canvas screen and drawing mode helps to draw or write content on the screen. A black canvas mask has also been included which works as a board. To perform all these actions some of the software requirements specifications are mentioned below.

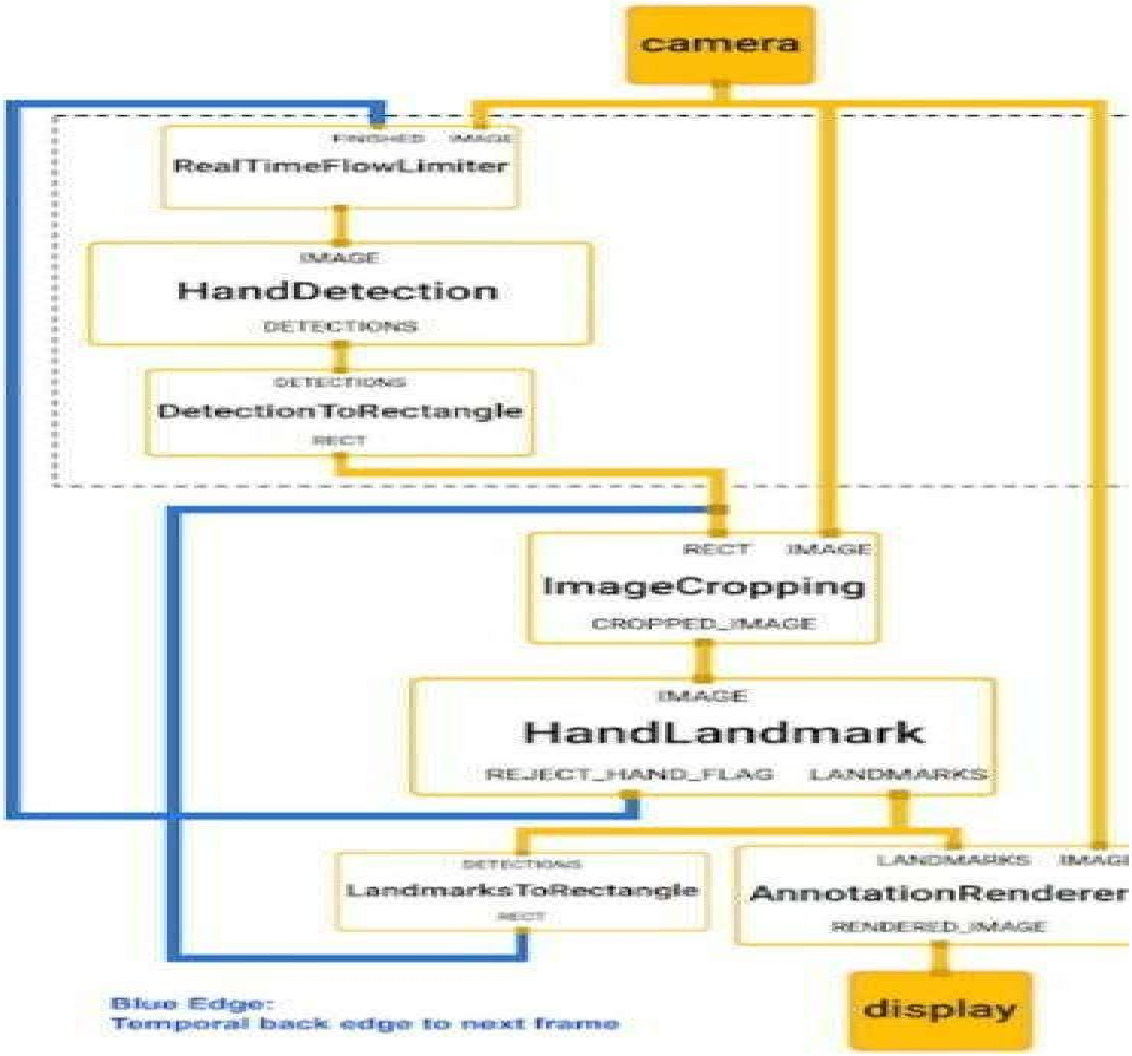

Fig 1. Architecture Diagram of Project

This is the most exciting part of a system. Writing involves a lot of functionalities. So, the number of gestures used for controlling the system is equal to these number of actions involved. The basic functionalities included in a system are:

1. Writing Mode - In this state, the system will trace the fingertip coordinates and stores.
2. Color Mode - The user can change the color of the text among the various available colors.
3. Circle and Square - If the user wants to draw circle and square, select circle or square.

3.4 Algorithms and Flowcharts Flow Chart:

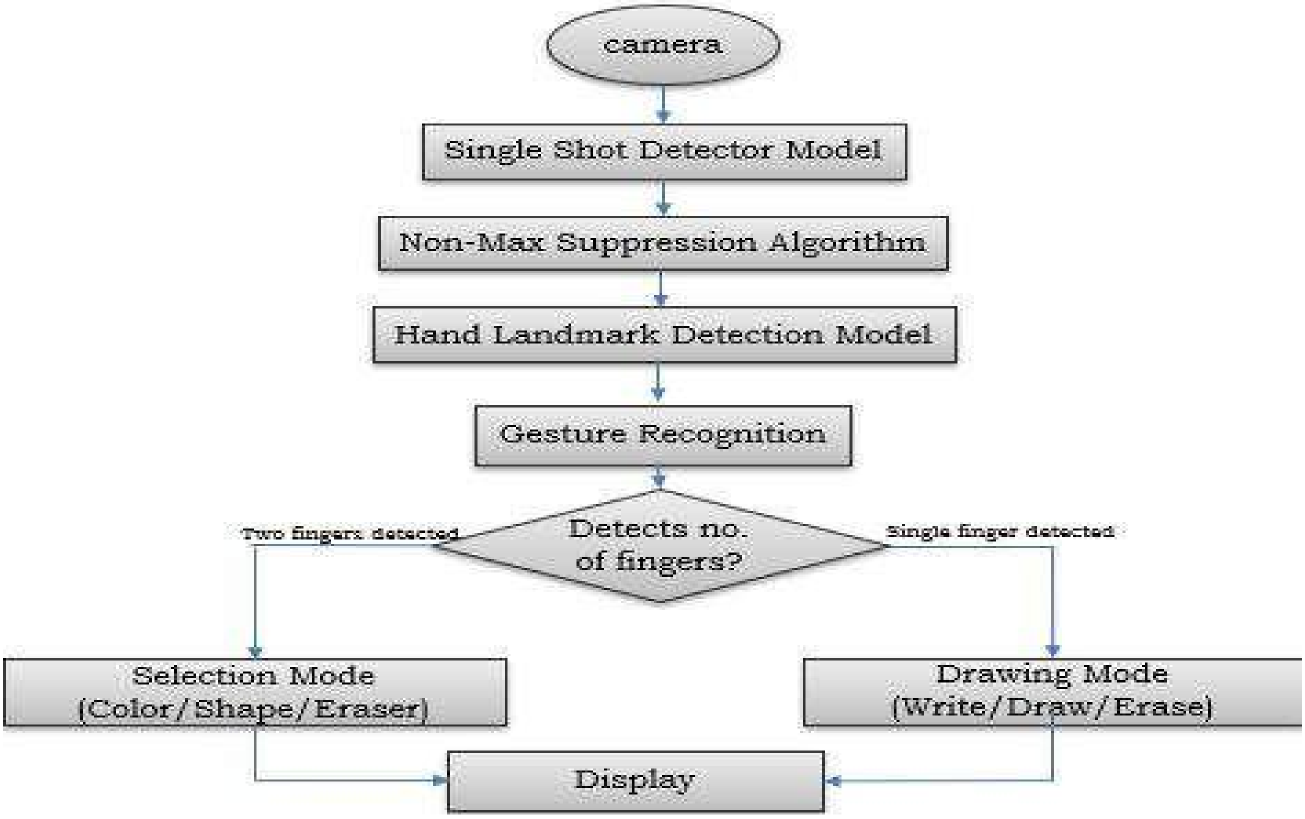

Fig 2. Flow Chart of Air Canvas

4. DESIGN

4.1 Introduction

The basic goal of Air Canvas is to map the coordinates of the user's pointer finger to the screen, where colored circles are drawn and connected to simulate a crude brush stroke. Project by searching for open space hand gesture recognition software that utilized OpenCV in combination with Python. In doing so, a project's design changed as discovered different image processing algorithms.

A primitive implementation sought to use hand gestures to control a cola and size variables. To do so, first set out to create an image mask that would separate the hand from the background. With some trial and error using OpenCV, successfully captured an image, Gaussian blurred it, and applied a binary mask to starkly contrast the hand shape from the background. This is a method obtained from Izane's Finger Detection tutorial1, chosen because of its use of convexity detection; in other words, determining the valleys between the fingers.

Discovered a suitable finger detection algorithm, which first takes a cola histogram of the palm to adapt to changing light conditions. Using a grid of 9 small rectangles, pixel data from each rectangular region is used to create a cola histogram. The histogram is then used in a built-in OpenCV function, cv2.calcBackProject, to separate the features in an image. In this case, the program is separating all pixels containing the appropriate colors from the rest of the frame. The image is then smoothed via thresholding and filtering, resulting in a frame that contains the hand alone. From here, the conta of the image is found. According to the

OpenCV website, contas are curves which join continuous points of the same cola or intensity and are used for shape analysis and detection.

4.2 Module Design and Organization

Module 1: Palm Detection Model

To detect initial hand locations, designed a single-shot detector model optimized for mobile real-time uses in a manner similar to the face detection model in MediaPipe Face Mesh. Detecting hands is a decidedly complex task: a model has to work across a variety of hand sizes with a largescale span (~20x) relative to the image frame and be able to detect occluded and self-occluded hands.

Whereas faces have high contrast patterns, e.g., in the eye and mouth region, the lack of such features in hands makes it comparatively difficult to detect them reliably from their visual features alone. Instead, providing additional context, like arm, body, or person features, aids accurate hand localization.

A method addresses the above challenges using different strategies. First, train a palm detector instead of a hand detector, since estimating bounding boxes of rigid objects like palms and fists is significantly simpler than detecting hands with articulated fingers. In addition, as palms are smaller objects, the non-maximum suppression algorithm works well even for two-hand self-occlusion cases, like handshakes. Moreover, palms can be modelled using square bounding boxes (anchors in ML terminology) ignoring other aspect ratios, and therefore reducing the number of anchors by a factor of 3-5. Second, an encoder-decoder feature extractor is used for bigger scene context awareness even for small objects.

Lastly, minimize the focal loss during training to support a large number of anchors resulting from the high scale variance. With the above techniques, it achieves an average precision of 95.7% in palm detection. Using a regular cross entropy loss and no decoder gives a baseline of just 86.22%.

Fig 3. Finding extreme points in contours with OpenCV and MediaPipe

Module 2: Hand Landmark Model

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

To obtain ground truth data, manually annotated ~30K real-world images with 21 3D coordinates, as shown below (take Z-value from image depth map, if it exists per corresponding coordinate). To better cover the possible hand poses and provide additional supervision on the nature of hand geometry, also render a high-quality synthetic hand model over various backgrounds and map it to the corresponding 3D coordinates.

ISSN NO: 2249-3034

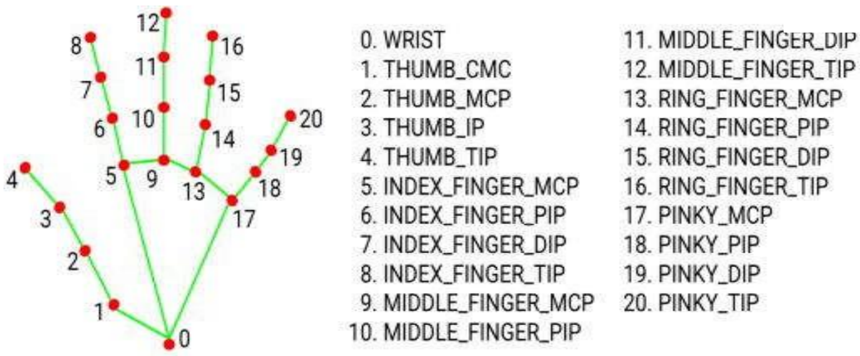

Fig 4. 21 Hand Landmarks

Module 3: Virtual Paint

As soon as the user shows up his hand in the camera the application detects it & draws a bounding box around the hand.

- If User shows only Index Finger than he/she is in drawing mode.
- To Select different cola 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.
- Moment of the hand is tracked by find position function, then user can write whatever they want by waving hand in the air.

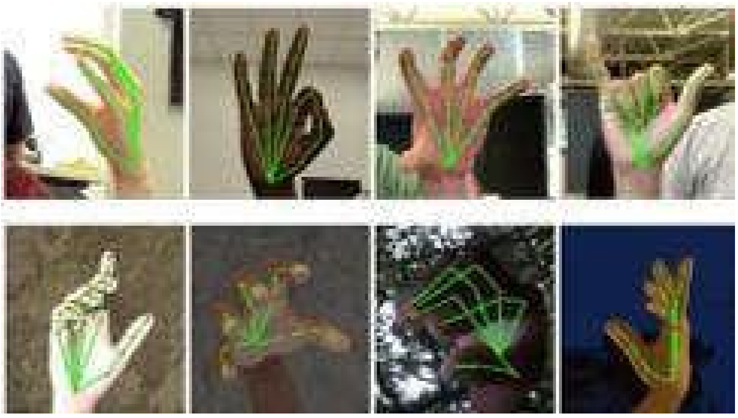

Fig 5. Virtual Canvas

Module 4: Gesture Recognition

- On the predicted hand skeleton, the state of each finger, e.g., bent or straight, is determined by the accumulated angles of joints. Then the set of finger states are mapped to a set of pre-defined gestures.
- This straight forward yet effective technique allows us to estimate basic static gestures with reasonable quality

Fig 6. Types of Gestures

5. IMPLEMENTATION AND RESULTS

Smart air canvas plays an important role in e-learning where the mode of explanation becomes easier and the students can understand the concepts in a better way. Writing in air has been one of the most fascinating and challenging research areas in field of image processing and pattern recognition in the recent years. It contributes immensely to the advancement of an automation process and can improve the interface between man and machine in numerous applications. Several research works have been focusing on new techniques [3] and met holds that would reduce the processing time while providing higher recognition accuracy.

Traditional training emphasizes individual student work and projects and is poor preparation for a student's future endeavors, which are likely to include working on teams and collaborating with colleagues. Under this training model, students receive few opportunities to practice group dynamics, teamwork and lacks the motivation where as in smart air canvas is a new and interactive way where it captivates the attention of multiple users irrespective of their age and field.

5.1 Technologies used are:

- **OpenCV:** OpenCV (Open Space Computer Vision Library) is an open space computer vision and machine learning software library.
- **MediaPipe:** MediaPipe Hands is a high-fidelity hand and finger tracking solution. It employs machine learning (ML) to infer 21 3D landmarks of a hand from just a single frame.
- **NumPy:** NumPy can be used to perform a wide variety of mathematical operations on arrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices.

5.2 Hand Detection and Tracking

MediaPipe Hands is a high-fidelity hand and finger tracking solution. It employs machine learning (ML) to infer 21 3D landmarks of a hand from just a single frame. There are 3 main phases in this.

Find hands: For this, input is taken from the webcam and processed by using MediaPipe. And it produces cropped hand image as output.

Find Position: It is to find hand position. This takes a cropped hand image as input and generates all the coordinates of landmarks of the complete hand. By this can know what landmark of hand is situated in which position in the whole image.

FingersUp: This tells us which fingers are open and which are closed based on the coordinates generated by the "FindPositon" method. For thumb, it sees x coordinates positions of landmark 4 and 3. If the x coordinate of 3 is greater than x coordinate of 4, then the thumb is open otherwise close. For fingers other than the thumb, y coordinate is compared with its tip and one landmark below the tip and decides whether the finger is open or closed. 24

Selection mode: If index and middle finger both are up, then it is selection mode.

Drawing mode: If only the index finger is up, then it is drawing mode.

5.3 Writing on Canvas

1. Importing Libraries
2. Define a color palette. Going to use red and blue colors in a palette to be used by a marker to paint in space.
   import numpy as np
   import cv2
   import mediapipe as mp

3. Deques (doubly ended queues) are used to store the points of movement left in trail by a marker. Since using red and blue colors in a palette, shall use two such deques for the two colors.
   # Blue Red Green Yellow White
   color_list = [(255, 0, 0), (0, 0, 255), (0, 255, 0), (0, 255, 255), (255, 255, 255)] # Blue Red
   color_palette_list = [(255, 0, 0), (0, 0, 255)] # index for the colors in a palette
   idx = 0
4. It can adjust the length of deque depending on how long, want the trail of the marker to be.
5. Reading the camera video feed *camera = cv2.VideoCapture(0)* 25
   If using older versions of OpenCV, then the problem might occur here regarding unpacking of values. Build some components so the captured screen looks an actual canvas. So, it shall create tabs for switching the colors and for clearing previous drawings.
6. After this, draw a circle to specify the position of the marker detected by an application. To read more on 'moments' and 'contas' in OpenCV. Functionality is brought to the tabs created earlier.
7. According to the position of the marker, an application should switch the colors or clear the screen.
8. When no colors are detected, append the next list (which is obviously a deque here). This helps when a marker is not present on or hidden from the screen.
9. Finally, to paint on the canvas, run the deque values to trace the stored points. Now, the starting color of the pencil is blue. If want to change it, it can be done by changing the order of colors in the color palette list.

5.3 Method of Implementation

5.3.1 Output Screens

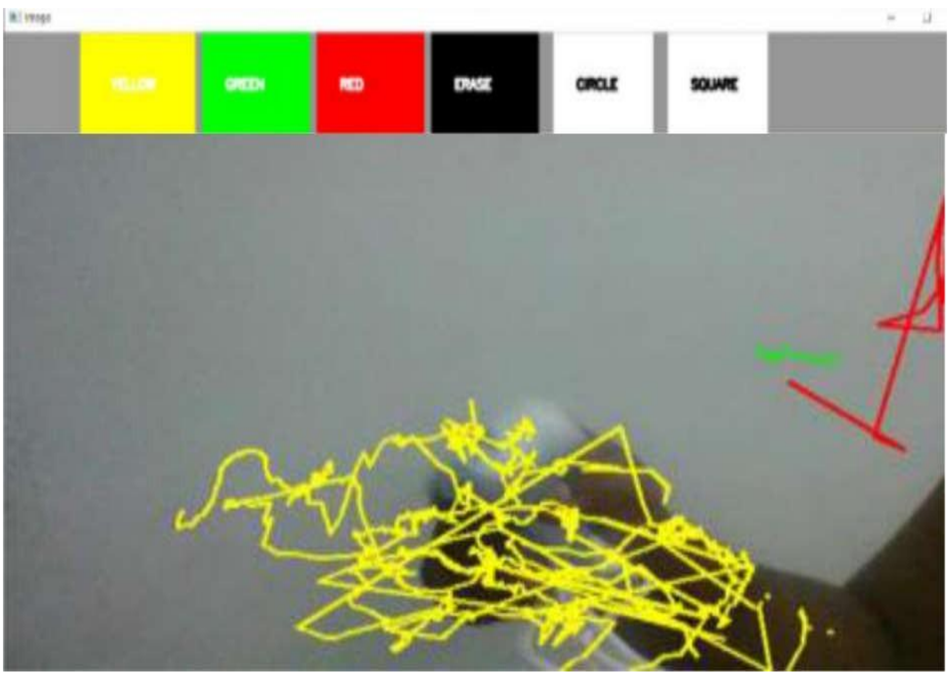

Fig 7. Tracking of continuous lines

ISSN NO: 2249-3034

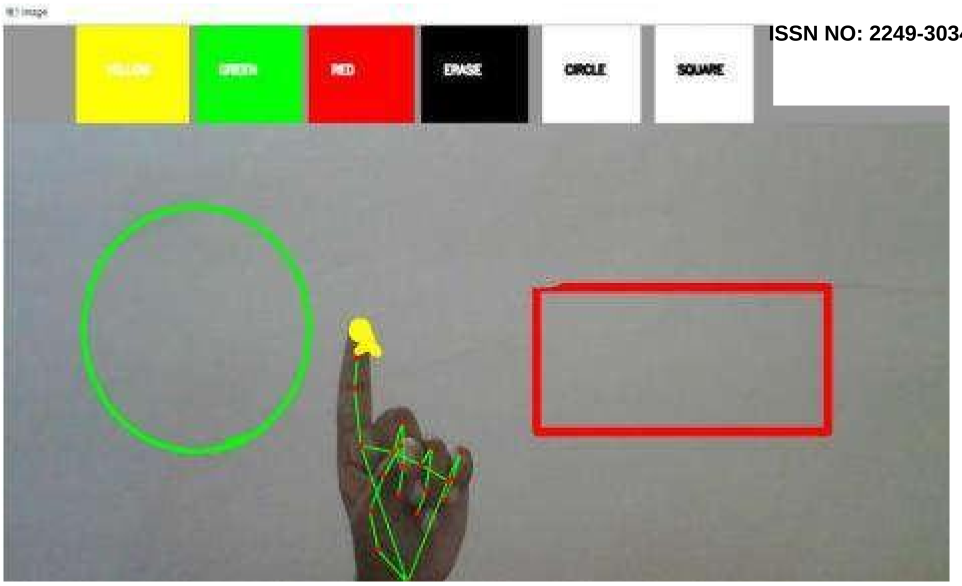

Fig 8. Selection of font colors

Fig 9. Selection of shapes

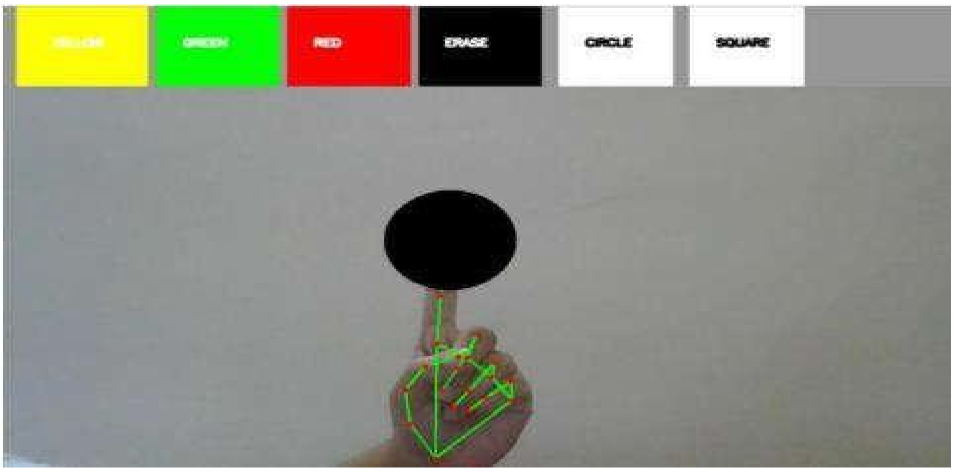

Fig 10. Selection of eraser

5.3.2 Result Analysis

This new model works perfectly fine for any number of objects before the camera. Different shapes and colors can be used as well. The drawbacks were rectified using MediaPipe.

MediaPipe Hands utilizes an ML pipeline consisting of multiple models working together: A palm detection model that operates on the full image and returns an oriented hand bounding box. A hand landmark model that operates on the cropped image region defined by the palm detector and returns high-fidelity 3D hand key points. This strategy is similar to that employed in a Media Pipe Face Mesh solution, which uses a face detector together with a face landmark model. When the hand is placed in front of the camera the hand is detected and the landmarks are mapped onto the palm. Later the fingers and the position of the fingers are detected.

When the index and middle finger are up it is in the selection mode, where the user is allowed to choose the font color or the shape required and is allowed to select the eraser if required. When only the index finger is up it is in the drawing mode and the contents are displayed as the user moves their fingers. Providing the accurately cropped hand image to the hand landmark model drastically reduces the need for data augmentation (e.g., rotations, translation and scale) and instead allows the network to dedicate most of its capacity towards coordinate prediction accuracy.

In addition, in a pipeline the crops can also be generated based on the hand landmarks identified in the previous frame, and only when the landmark model could no longer identify hand presence is palm detection invoked to relocalize the hand.

6. Testing and Validation

In previous chapters, we implemented our air canvas. Simply designing and implementing a canvas does not guarantee that it will function properly and that we will be able to use it. To use our canvas, we must first test it and validate it with our expected results, then modify it as needed and test until our expected output is obtained.

6.1 Design of test cases and scenarios

Some of the scenarios where we can use our air canvas to test are:

1. Free Handwriting with Desired color.
2. Writing Math Formulas.
3. Drawing images with freehand.

6.2 Validation

Fig 11. Free hand writing with desired color

Fig 12. Writing Math Formulas

Fig 13. Drawing images with Freehand

7. CONCLUSION & FUTURE WORK

Conclusion:

The system has the potential to challenge traditional writing methods. It eradicates the need to carry a mobile phone in hand to jot down notes, providing a simple on-the-go way to do the same. It will also serve a great purpose in helping especially disabled people communicate easily. Even senior citizens or people who find it difficult to use keyboards will able to use system effortlessly. Drawing in the air can also be made possible.

Air canvas can make the interaction between the teacher and the students much more interesting where the complex topics can be easily represented using the functionalities available. There is no need for the teacher to adjust the camera according to the external surroundings, as the user can simply share the screen while teaching and can be properly viewed by all.

Future Enhancements:

Much more accuracy can be brought by including hand landmarks recognition, and multicore module.

- Whatever has been written on the Air canvas can be saved and prepared into a document for reference.
- OCR takes images that include textual elements and attempts to recognize that text. The output is a text string, and accuracy is measured as the degree of similarity between the recognized text and the text a human would be able to read from the image.
- OCR is used to recognize printed text in paper documents, handwritten characters, and text elements in the physical environment, such as license plate numbers, street signs, and street numbers. Traditional OCR algorithms are based on pattern matching, pattern recognition, or image correlation. These techniques, in a standard use case such as a document scanner, can recognize words and sentences with a very high level of accuracy.
- To enhance hand gesture tracking, development more into OpenCV. Furthermore, experiment could be done with different interpolation methods such as PyGame includes a line drawing method (pygame draw.line()) that could prove useful in producing smoother and cleaner lines.
- On the same vein, implementing a variety of brush shapes, textures, and even an eraser would make this tool more robust as a drawing application.

REFERENCES

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

2. EshedOhn-Bar, Mohan Manubhai Trivedi, "Hand Gesture Recognition In Real-Time For Automotive Interfaces," IEEE Transactions on Intelligent Transportation Systems, VOL. 15, NO. 6, December 2014, pp 2368-2377.
3. Y. Huang, X. Liu, X. Zhang, and L. Jin, "A Pointing Gesture Based Egocentric Interaction System: Dataset, Approach, and Application," 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Las Vegas, NV, pp. 370-377, 2016.
4. Comparison of OpenCV's Feature Detectors and Feature Matchers Frazer K. Noble Centre for Additive Manufacturing School of Engineering and Advanced Technology Massey University New Zealand.
5. Free Air Finger Painting on a Projected Canvas Jon Moeller, Nic Lupfer, Bill Hamilton, Haiqiao Lin.Interface Ecology Lab Texas A&M University Department of Computer Science and Engineering 3112 TAMU College Station, TX USA.