AIR CANVAS AND EMOJINATOR

Submitted in partial fulfilment for the award of the degree of

B.Tech (Computer Science and Engineering)

By

Parth Sachan | 18BCE2373

Shashank Shukla | 18BCE2522

Mihir Agarwal | 18BCE2526

Prepared For

Human Computer Interaction (CSE4015)

Under the Guidance of

Prof. Krishnamoorthy A

Assistant Professor (Senior)
School of Computer Science and Engineering

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Table of Contents

1. Abstract

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 and methods that would reduce the processing time while providing higher recognition accuracy. Object tracking is considered as an important task within the field of Computer Vision. The invention of faster computers, availability of inexpensive and good quality video cameras and demands of automated video analysis has given popularity to object tracking techniques. The project takes advantage of this gap and focuses on developing an air canvas as well as an emojinator. This project is a reporter of occasional gestures and feelings (through emojis). It will use computer vision to trace the path of the finger or user's particular action. The air canvas can be used to draw something in the air and the emojinator can be used in speech-to-text systems in order for system to take emojis as input.

Keywords - Air Writing, Object Detection, Real-Time Gesture Control System, Emojinator, Computer Vision.

2. Chapter 1

2.1 Aim

To make an air canvas for user to draw in air and to make an emojinator which can detect any hand emoji as per user's action in front of the video feed.

2.2 Objective

The objective of this project is to make an air canvas that will enable the user to draw in air in different colours thereby eliminating the need to use a mouse or a drawing graphics tablet. Air canvas is implemented using open cv, it detects colour of the pen adds required colour on the canvas. Different image processing techniques of open cv are also taken in to account. The project also includes an emojinator (emoji detector) that can detect an emoji that a user is making and can be used in various speech-to-text systems where putting an emoji as an input may not be feasible. It is implemented using open cv, tensorflow object detection API and keras. A model is trained to detect different emojis made by hand gestures, then hand is recognized using object detection and detected emoji is overlayed on the frame using open cv.d

2.3 Introduction

In the era of 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 the digital art. From the recipient to analyse, it can simply be divided into visual art, audio art, audio-visual art and audio-visual imaginary art, which includes literature, painting, sculpture, architecture, music, dance, drama and other works of 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 same situation happens in art. In the present circumstances, digital art and traditional art are inclusive of the symbiotic state, so we need to systematically understand the basic knowledge of the form between digital art and traditional art. The traditional way includes pen and paper, chalk and board method of writing. The essential aim of digital art is of building 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, we are using hand gesture recognition with the use of deep learning by using python programming, which creates natural interaction between man and machine. With the advancement in technology, the need of development of natural ‘human – computer interaction (HCI)’ systems to replace traditional systems is increasing rapidly.

Users are used to draw on virtual canvases using mouse or the mouse pad in their laptops but we are proposing a canvas that’s pretty much invisible. This is a basic project based on computer vision made in OpenCV Python which enables the user to draw on their system screen by drawing in air with a target, preferably the tips of your finger, which is tracked by the computer webcam. The aim is to track the target first, and then its motion and be able to replicate its path on the screen. This is done using OpenCV filters like Gaussian blur. The location of the target is tracked, the image is masked and the centre of the target is calculated. Then the path of the centre of the target is drawn on the screen. This is the basic idea of what is planned to be done. Some other features to be added will include a colour palette, and other features of a classic Paint application. The scope for other improvements could include incorporating a model to interpret what is written, atleast numbers.
Emojis are ideograms and smileys used in electronic messages and web pages. Emoji exist in various genres, including facial expressions, common objects, places and types of weather, and animals. They are much like emoticons, but emoji are actual pictures instead of typographics. Emojinator is an emoji detector software which can recognize and classify different hand emojis.

3. Chapter 2

3.1 Related Work or Literature Survey

1] Egocentric-View Fingertip Detection for Air Writing Based on Convolutional Neural Networks

The authors in [1] investigated a real-time fingertip detection in frames captured from the increasingly popular wearable device, smart glasses. The egocentric-view fingertip detection and character recognition proposed in this paper can be used to create a novel way of inputting texts. Unity3D is used to build a synthetic dataset with pointing gestures from the first-person perspective. Following that, a modified Mask Regional Convolutional Neural Network (Mask R-CNN) is proposed, consisting of a region-based CNN for finger detection and a three-layer CNN for fingertip location. The speed concluded by the authors is high enough to enable real-time "air-writing", where users are able to write characters in the air to input texts or commands while wearing smart glasses. The characters can be recognized by a ResNet-based CNN from the fingertip trajectories.

2] Hand Gesture Recognition in Real Time for Automotive Interfaces

In [2], the authors have developed a vision-based system that employs a combined RGB and depth descriptor to classify hand gestures. The method is studied for a human–machine interface application in the car. Two interconnected modules are employed: one that detects a hand in the region of interaction and performs user classification, and another that performs gesture recognition. The authors have also demonstrated the feasibility of the system by using a challenging RGBD hand gesture data set collected under settings of common illumination variation and occlusion.

3] Text Writing in Air

The authors in [3] presented a real time video based pointing method which allows sketching and writing of English text over air in front of mobile camera. The proposed method has two main tasks: first it tracks the colored finger tip in the video frames and then apply English OCR over plotted images in order to recognize the written characters. Moreover, the proposed method provides a natural human-system interaction in such way that it does not require keypad, stylus, pen or glove etc for character input. The proposed system is a software-based approach and relevantly very simple, fast and easy. It does not require sensors or any hardware rather than camera and red tape. The only drawback is that it is color sensitive in such a way that existence of any red color in the background before starting the character writing can lead to false results.

4] Visual Gesture Recognition for Text Writing in Air

The authors in [4] solved problems faced by elderly people to type text on mobile phones by detecting gestures in air. In this system, a combination of computer vision and convolution neural networks is used for detecting drawn gesture and recognizing it. Few of the methods employed, use coloured fingertip for tracking the motion of the finger. Their proposed application supports drawing of gestures and writing of English text over air in front of the mobile camera by using bare fingertip. Convolution neural networks has been used for character recognition. The method presented can be applicable to other fields requiring hand gesture recognition for distant interaction with mobiles and computers. Their proposed System does not require sensors or any hardware other than the camera.

5] Air-swipe gesture recognition using OpenCV in Android devices

In [2], authors introduce Air-Swipe Gesture Recognition System which can be useful to enable user to make In-Air gestures in front of the camera and to do different operations. This System can give a user-friendly and a live-experience of interaction and visualization, enhancing the usability and making the android device more interactive. It does not require any hardware changes instead only uses the native camera of the device and a machine learning software such as Open-Source Computer Vision (OpenCV) algorithms to detect the changes in environment and respond accordingly in varying conditions. They tested this classification and found out the result that considering the frames to be divided into quadrants along x-axis and y-axis and found that the value of the frame matrix changes. Their approach has the capability of recognizing gestures with precision of almost 96%.

6] Text Recognition by Air Drawing

In this paper [3], the text drawn by the user in the air is captured by the computer's camera, followed by the identification of that text. So, the video camera will be turned on at the time of capturing the written text. Now the object is defined based on its colour to detect a movement done by the user. The colour is captured by the lower and upper bound of HSV (Hue Saturation Value), which finally leads to object detection at every instant. Lastly, the text will be recognized by the trained model. The model is trained by CNN (Convolution Neural Network) with an accuracy of 98.64% (training) and 98.24% (testing). For completion of the project, OpenCV, python programming language, and its libraries are used. This project requires only a camera and a defined object.

7] Air Drums: Playing Drums Using Computer Vision

The cost of a drum set is an investment that most aspiring drummers would eventually need to shoulder in order to continue their craft. What this research aims to do is to hasten the introduction of drummers to the drumming experience without the costs, and also to allow for drummers to be able to practice, at least casually, without a full drum set. This thus allows the experience of drumming to a wider audience. The solution we explore is the development of a prototype virtual drum set that would only require users to have a laptop with a camera, along with easily accessible markers representing the tips of drum sticks and knee movement, such as colored papers. OpenCV

based on Python was used to implement this, and used the concept of color-based blob detection for detecting the markers.

8] Air-Writing for Smart Glasses by Effective Fingertip Detection

This research investigates real-time fingertip detection in RGB images/frames captured from such wearable devices as smart glasses. A modified Mask Regional Convolutional Neural Network (Mask R-CNN) is proposed with one region-based CNN for hand detection and another three-layer CNN for locating the fingertip. The processing speed is high enough to facilitate several interesting applications. One application is to trace the location of a user's fingertip from first-person perspective to form writing trajectories. A text input mechanism for smart glasses can thus be implemented to enable a user to write letters/characters in air as the input and even interact with the system using simple gestures. Experimental results demonstrate the feasibility of this new text input methodology.

9] A new fingertip detection and tracking algorithm and its application on writing-in-the-air system

Writing-in-the-air (WIA) system provides a novel input experience using the fingertip as a virtual pen based on the color and depth information from only one Kinect camera. We present a new fingertip detection and tracking framework for the robust and realtime fingertip position estimation and further improve the air-writing character recognition accuracy. Firstly, we propose a new physical constraint and an adaptive threshold with the mode temporal consistency in order to classify various hand poses into two modes, i.e., the side-mode and frontal-mode. In the side-mode, a new choose-to-trust algorithm (CTTA) is proposed for the hand segmentation. The final segmentation result is generated by selecting a more trustable color or depth model-based segmentation result according to the fingertip-palm relationship. In the frontal-mode, we propose to estimate the fingertip position by a joint detection-tracking algorithm that successfully incorporates the temporal and physical constraints.

3.2 Existing and Proposed Systems

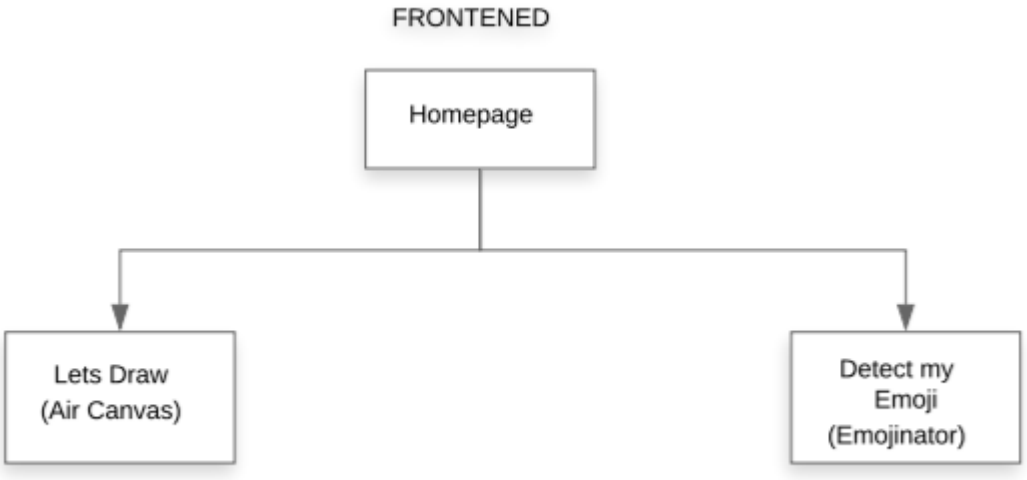
Traditionally, everyone uses old white boards or power points to give presentation. Specially in these days when everything is online, people use zoom and google meets while presentation, they need to use mouse to draw, but in our proposed system, people don’t need to use a mouse or a drawing tablet, they can directly use their fingers while demonstrating in a video call. While using the traditional white boards, people need to spend money for the white boards, but in our cost-effective method, they don’t need anything other than their camera.

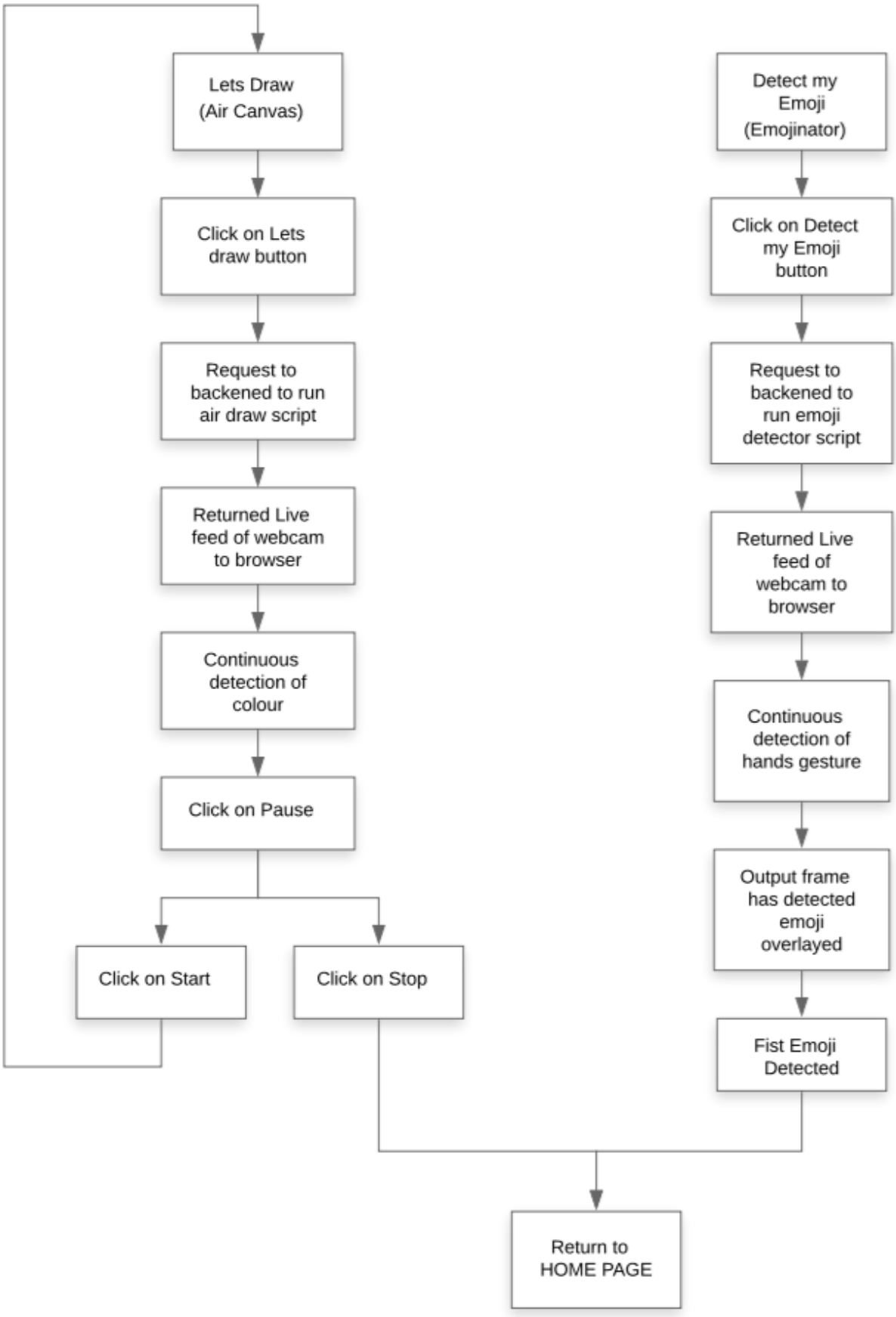
In our proposed emojinator, people can use emojis just by their hands, they don’t need to use the keyboard to express their emotions. While they are presenting, our emojinator will display the appropriate emoji on the screen. This can be used during an online show or meeting where reactions can be expressed without typing and just within a few seconds.

4. Chapter 3

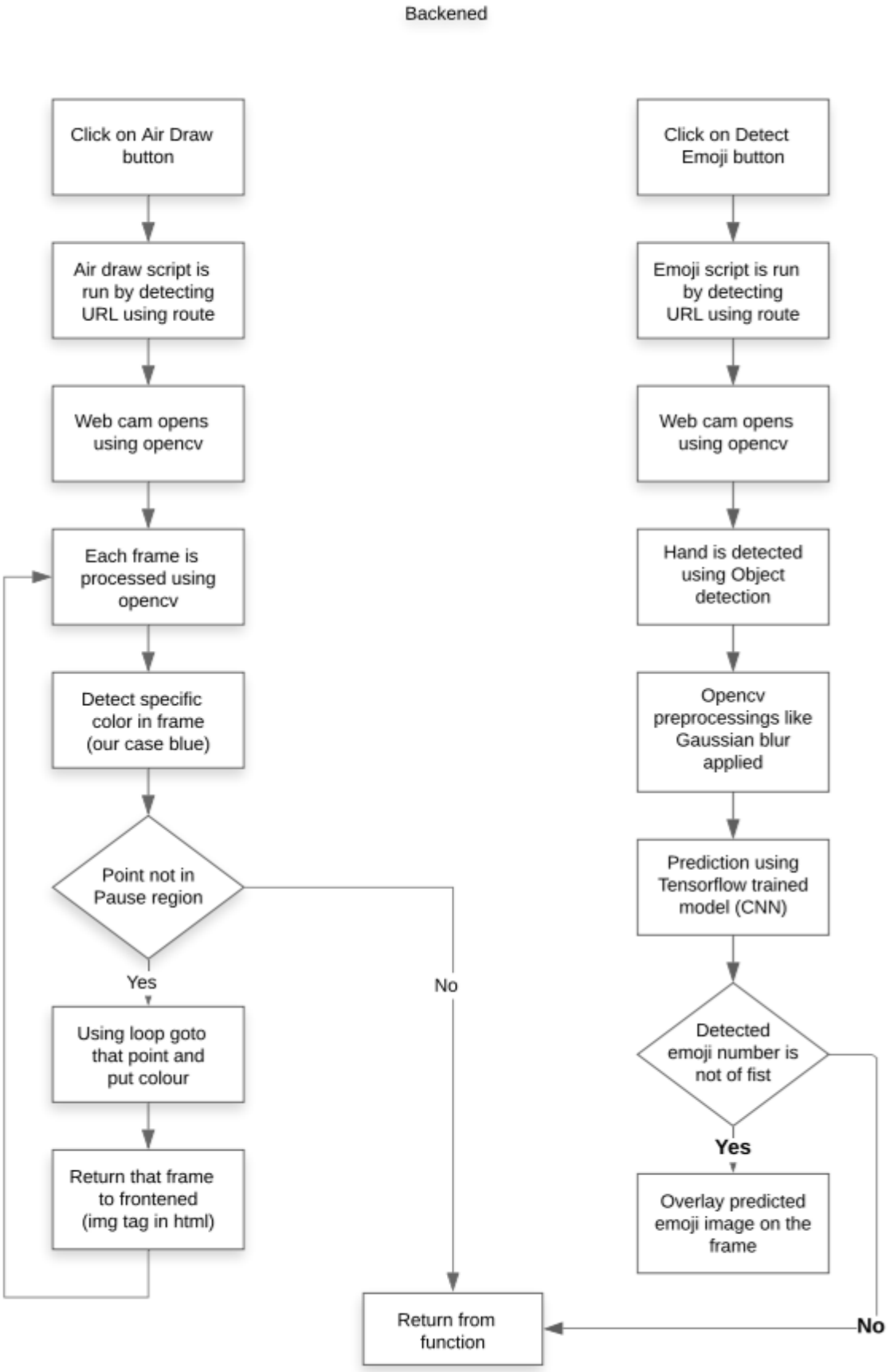
4.1 Proposed System Design Architecture

4.1.1 Frontend




Lets Draw
(Air Canvas)
Click on Lets
draw button
Request to
backened to run
air draw script
Returned Live
feed of webcam
to browser
Continuous
detection of
colour
Click on Pause
Click on Start
Click on Stop
Detect my
Emoji
(Emojinator)
Click on Detect
my Emoji
button
Request to
backened to
run emoji
detector script
Returned Live
feed of
webcam to
browser
Continuous
detection of
hands gesture
Output frame
has detected
emoji
overlayed
Fist Emoji
Detected
Return to
HOME PAGE

4.1.2 Backend


Backened
Click on Air Draw button
Air draw script is run by detecting URL using route
Web cam opens using opencv
Each frame is processed using opencv
Detect specific color in frame (our case blue)
Point not in Pause region
Yes
No
Using loop goto that point and put colour
Return that frame to frontened (img tag in html)
Click on Detect Emoji button
Emoji script is run by detecting URL using route
Web cam opens using opencv
Hand is detected using Object detection
Opencv preprocessings like Gaussian blur applied
Prediction using Tensorflow trained model (CNN)
Detected emoji number is not of fist
Yes
No
Overlay predicted emoji image on the frame
Return from function

4.1.2 Tools and Technologies used / Hardware – Software Requirements

- Tools and Technologies used:
  - Python v3.9
  - Flask
  - openCV
  - Tensorflow
  - Keras
  - HTML
  - CSS
  - Bootstrap
  - Git
- Hardware Requirements:
  - NVIDIA graphics card and respective drivers
  - CUDA driver v11.1 (for tensorflow GPU)
  - cuDNN v8.1
  - Inbuild webcam
- Software Requirements:
  - Web browser (Chrome version 68 and above)
  - Anaconda
  - VSCode

4.2 Architecture Explanation

Frontend:

Homepage- In homepage, two buttons are provided which can be selected by mouse click. “Let’s draw” button runs the air canvas script which further opens webcam feed into the browser enabling the user to draw in the air. Second button is “Detect my emoji” which enables users to run emoji detector script. After clicking on this button, webcam feed is shown in browser detecting real time emoji gestures made by hand.

Air Draw page- When Let’s draw button is clicked, a request to blackened is made to display live feed of webcam. After clicking, on next page, live feed is shown with different options to select different colours and start or stop the feed. When detecting colour is shown (in our case as blue), it is detected and successfully draws on the screen shown. On clicking stop, feed is stopped the given point and clicking on start, resumes it again. Back button takes back to homepage.

Emoji Detector page- When Detect my emoji button is clicked, a request to backend is made to run the model and show the detected emoji as well the live webcam feed on to the browser. On clicking, it successfully shows the feed on the page with the detected emoji over laying on the same feed.

Backend:

Main page script-

Air Draw script- When URL route matches with video URL, video.html file gets rendered and webpage having live webcam feed is sent to frontend using flask. The code function which produces webcam feed uses open cv and performs many other pre-processing to detect the colour of the pen.

In that function, first a trackbar is created to set hue and saturation to detect blue colour (or any colour of our choice using which we will write in air canvas). Frames are generated using open cv and each frame is pre-processed using different open cv techniques. Different rectangles of different colours are drawn on each of the captured frames using which different colours can be selected. Different open cv pre-processing like Eroding, Morphology, Dilation are used to improve detection of the colour. Then contour detection is done to detect specific point of the colour in the whole pixel range. Detected point is checked using if else that it lies in what region. According to region different colours are selected, similarly every frame of the video feed is processed and passed to frontend that combine and forms complete video.

Emoji Detector page- When URL route matches with emoji URL, emoji.html file gets rendered and webpage having live webcam feed is sent to frontend using flask. The code function which produces webcam feed uses open cv and performs many other pre-processing to detect the emoji made by hand gesture.

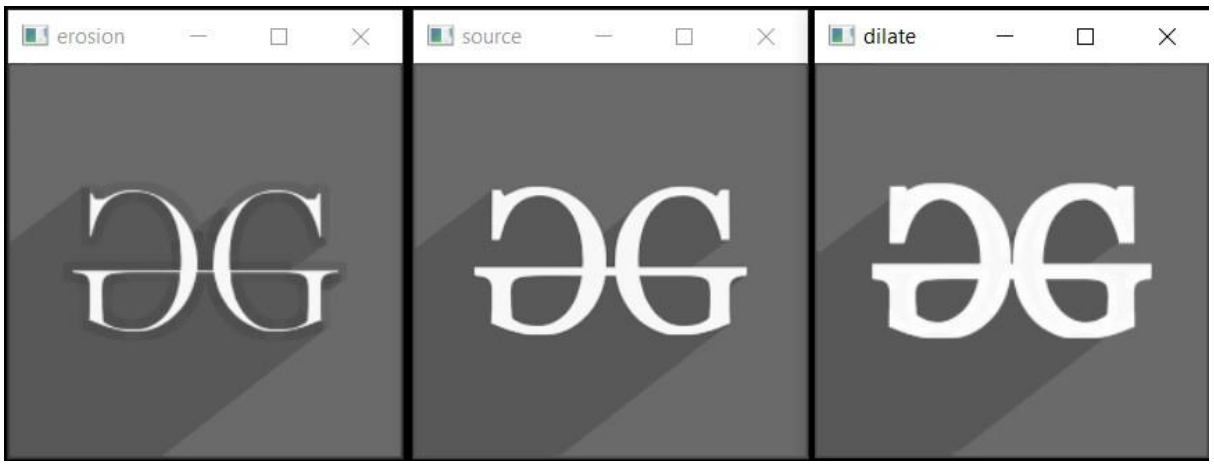
In that function, first a frame is extracted from the video using open cv, it is converted from BGR to RGB. Object detection algorithm is run on it to detect the required bounding box around the hand. In our case, TensorFlow Object Detection API is used to detect the object. Other open cv pre-processing techniques like Bitwise, Gaussian Blur, Morphology, and Dilation are used on the respective frame. After all the pre-processing, prediction using already trained Keras model is done using a predict function. The predicted emoji class number and the prediction probability are stored in the variables. Image of predicted emoji is extracted from one of the folders having emojis pictures and their class numbers. Emoji is then overlayed on the frame using overlay function. After all the steps, frame is sent to frontend to show it as video.

Model is trained using CNN (Convolutional Neural Network). CNN's are the state-of-the-art algorithms for training on image data, their architecture learns faster and more accurately when the training data are images. Standard layers like convolution, max pool, dense are used to make the model of CNN for our project. The model is trained for 10 epochs, and it gets converged in such fewer number of epochs.

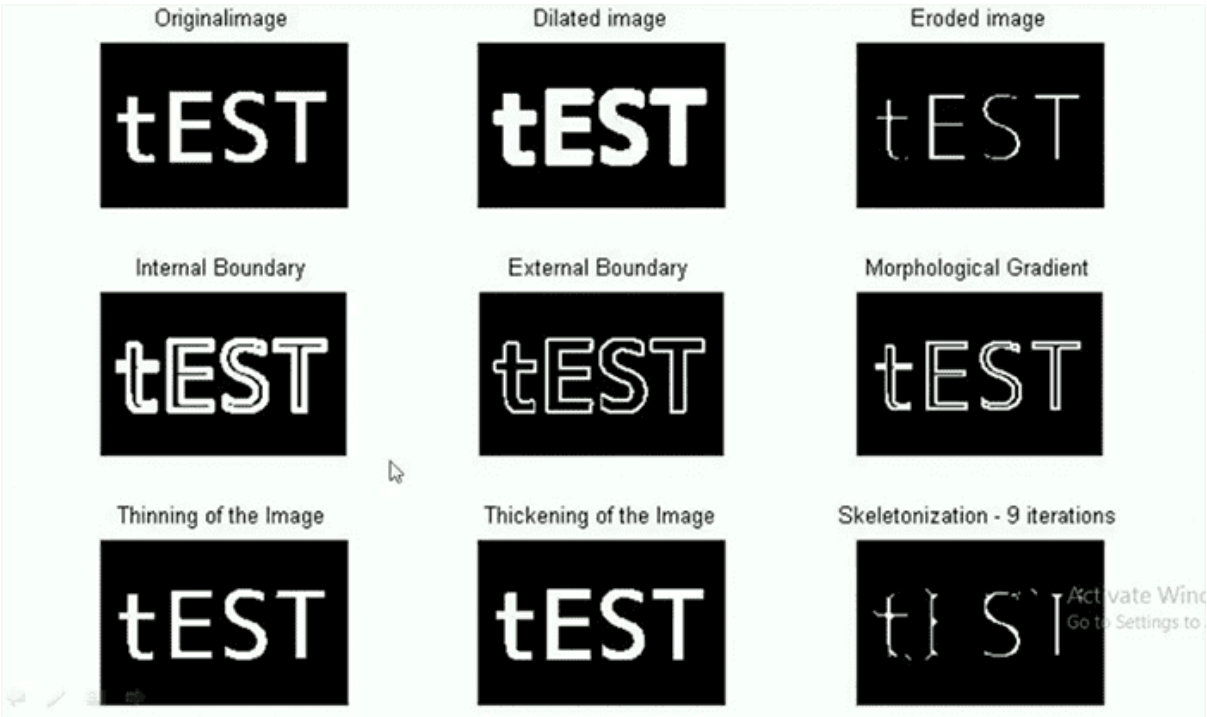
4.3 Algorithms & Pseudocode

Algorithms:

Erosion and Dilation- The most basic morphological operations are dilation and erosion. Dilation **adds pixels to the boundaries of objects in an image**, while erosion removes pixels on object boundaries. The rule used to process the pixels defines the operation as a dilation or an erosion.



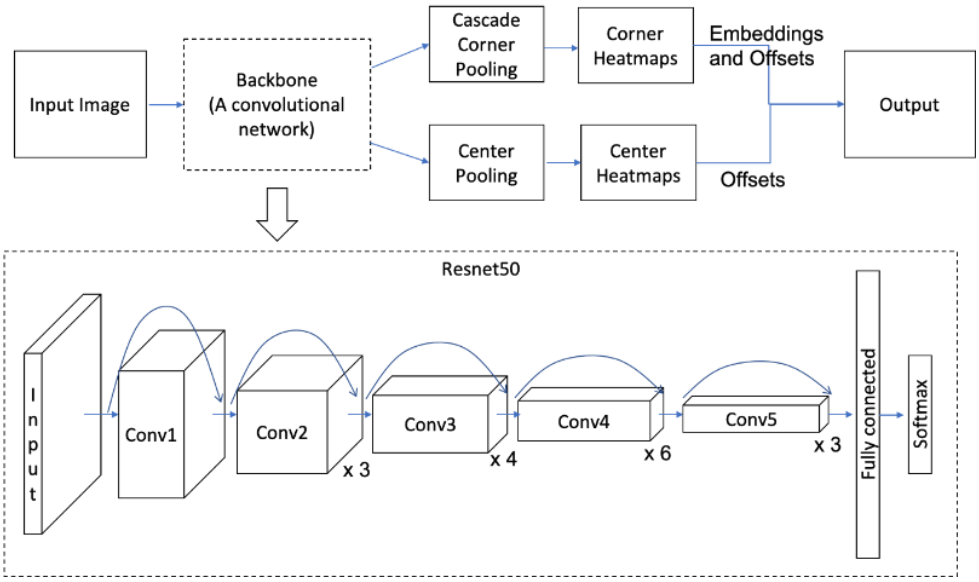
Morphology- Morphological operations are simple transformations applied to binary or grayscale images. We can use morphological operations to **increase the size of objects in images as well as decrease them.**



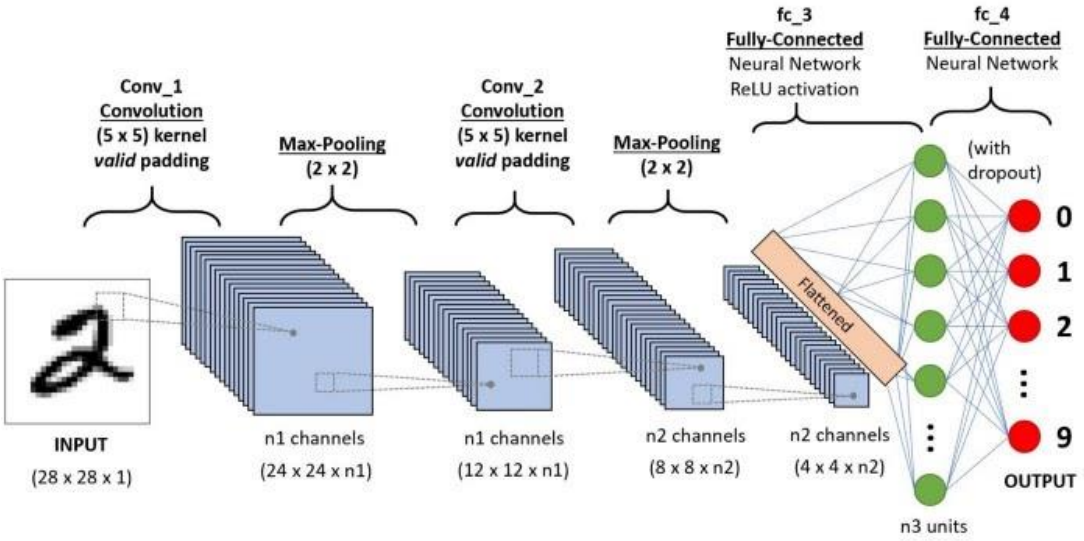
Gaussian Blur- In image processing, a Gaussian blur (also known as Gaussian smoothing) is the result of blurring an image by a Gaussian function (named after mathematician and scientist

Carl Friedrich Gauss). It is a widely used effect in graphics software**, typically to reduce image noise and reduce detail.**

Tensorflow Object Detection- Object Detection using Tensorflow is a **computer vision technique.** As the name suggests, it helps us in detecting, locating, and tracing an object from an image or a video.



Convolutional Neural Network (CNN)- A convolutional neural network (CNN) is a type of artificial neural network used in image recognition and processing that is specifically designed to process pixel data. CNNs are powerful image processing, artificial intelligence (AI) that use deep learning to perform both generative and descriptive tasks, often using machine vison that includes image and video recognition, along with recommender systems and natural language processing (NLP).



Pseudocode:

.

Frontened-

1- Homepage load
   a. If (Click on Let's Draw button)
      i. Choose colour from shown
      ii. Draw on screen
      iii. If (click on pause)
         1. Click on Start
            a. Repeat same process again
         2. Click on Stop
            a. Return to Home Page
   b. Else (Click on Detect my emoji button)
      i. Make hand gesture
      ii. Predicted emoji shown on webcam
      iii. If (Show fist emoji)
         1. Feed stops
      iv. If (Click on Back)
         1. Return to Home Page

Backened-

If (URL route == "video")

1- Runs Air Draw script
2- Processing each frame
3- Applying all algorithms
4- Detect specific colour
   If (point region == blue)
      Colour=blue
   Else if (point region == yellow)

Colour=yellow
Else if (point region == red)
Colour=red
Else if (point region == green)
Colour=green
Else if (point region == clear all)
Colour=no colour
Else if (point region == stop)
Break

5- If (button == back)
Return to main page
Else if (button == start)
Refresh the same page
6- Use loop to put colour up to that point region
7- Return the modified frame
8- All frame combines to make the video feed

Else if(URL route == “emoji”)

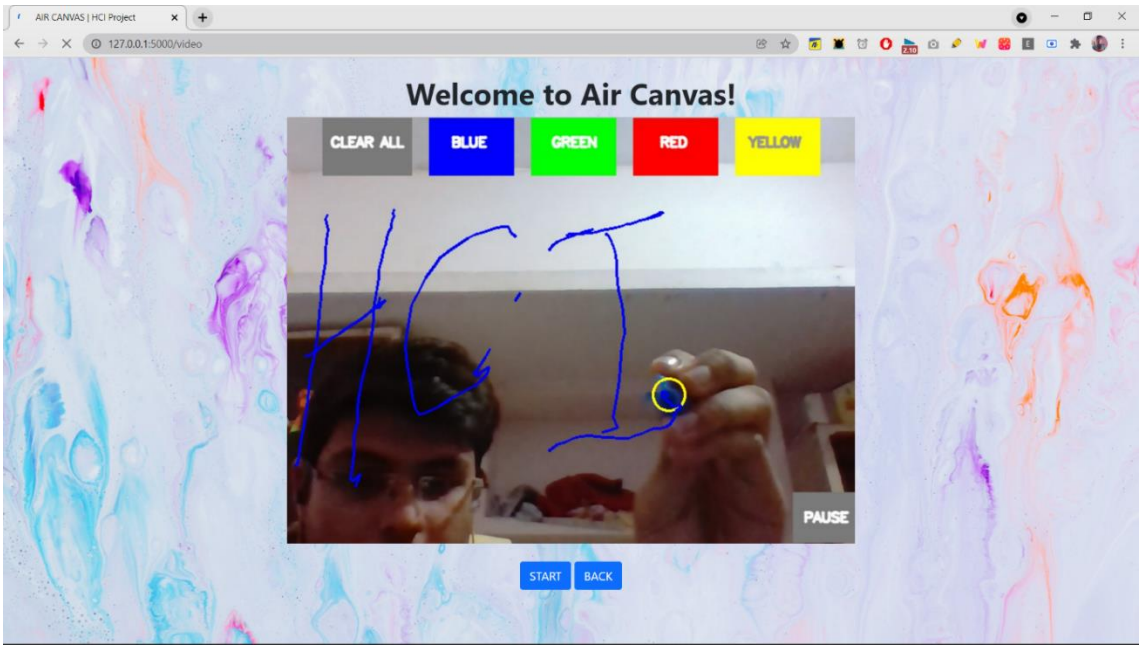
1- Runs Emoji detect script
2- Colour of detected frame is changed to RGB using open cv
3- Object detection algorithm is run (detect object function)
4- Draw box on image function is run to draw box
5- Gaussian Blur is applied using open cv
6- Similarly, morphology and dilation are applied
7- Thresholding is done using open cv
8- Image resized to 50 by 50
9- Prediction using keras.model.predict
10- If (pred class! = 10)
    a. Overlay emoji on the frame
    b. Convert it to jpg
    c. Yield it and send to frontend in img tag

Else
break

5. Chapter 4

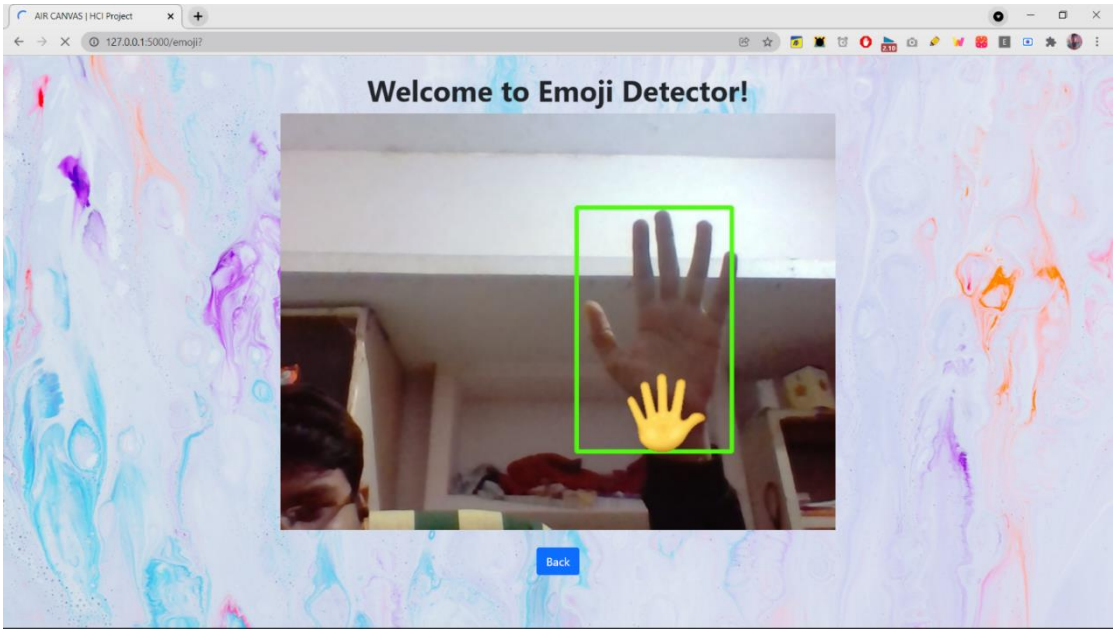
5.1 Results & Discussions

5.1.1 Air Canvas



In the above fig. we see the implementation of the project where a finger is used to draw in different colours, which can be used by professors to teach in online classes. The erase button can also be used to to erase the drawing once done. Today people use traditional whiteboard, and draw with mouse, but using our project, people can draw as if they are teaching in offline classes.

5.1.2 Emojinator



In the 2nd image it can be seen that emojis are generated according to the person's actions in camera. Which can be used by people while video calls. Today people use the reactions feature in zoom to show emojis, which will not be required after our project, people can show a thumbs up really instead of clicking on its icon, giving him the physical feel, which lacks in online meetings.

5.2 Conclusion & Future Work

The system has the potential to challenge traditional presentation and teaching methods. It will also serve a great purpose in helping especially abled people communicate easily. Even senior citizens or people who find it difficult to use keyboards will able to use system effortlessly. Extending the functionality, system can also be used while video calls shortly. The system will be an excellent software for smart wearables using which people could better interact with the digital world. Augmented Reality can make text come alive. There are some limitations of the system which can be improved in the future. Firstly, using a handwriting recognizer in place of a character recognizer will allow the user to write word by word, making writing faster. Secondly, hand-gestures with a pause can be used to control the real-time system as done by [1] instead of using the number of fingertips. Thirdly, our system sometimes recognizes fingertips in the background and changes their state. Air-writing systems should only obey their master's control gestures and should not be misled by people around. Also, we used the EMNIST dataset, which is not a proper air-character dataset. Upcoming object detection algorithms such as YOLO v3 can improve fingertip recognition accuracy and speed. In the future, advances in Artificial Intelligence will enhance the efficiency of air-writing.

6. References

1] V. Joseph, A. Talpade, N. Suvarna and Z. Mendonca, "Visual Gesture Recognition for Text Writing in Air," 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), 2018, pp. 23-26, doi: 10.1109/ICCONS.2018.8663176

2] T. Sharma, S. Kumar, N. Yadav, K. Sharma and P. Bhardwaj, "Air-swipe gesture recognition using OpenCV in Android devices," 2017 International Conference on Algorithms, Methodology, Models and Applications in Emerging Technologies (ICAMMAET), 2017, pp. 1-6, doi: 10.1109/ICAMMAET.2017.8186632.

3] J. Patel, U. Mehta, K. Panchal, D. Tailor and D. Zanzmera, "Text Recognition by Air Drawing," 2021 Fourth International Conference on Computational Intelligence and Communication Technologies (CCICT), 2021, pp. 292-295, doi: 10.1109/CCICT53244.2021.00061.

4] Chen Y-H, Huang C-H, Syu S-W, Kuo T-Y, Su P-C. Egocentric-View Fingertip Detection for Air Writing Based on Convolutional Neural Networks. Sensors. 2021; 21(13):4382. https://doi.org/10.3390/s21134382

5] E. Ohn-Bar and M. M. Trivedi, "Hand Gesture Recognition in Real Time for Automotive Interfaces: A Multimodal Vision-Based Approach and Evaluations," in IEEE Transactions on Intelligent Transportation Systems, vol. 15, no. 6, pp. 2368-2377, Dec. 2014, doi: 10.1109/TITS.2014.2337331.

6] Baig, Fasial & Khan, Muhammad & Beg, Saira. (2013). Text writing in the air. Journal of Information Display. 14. 10.1080/15980316.2013.860928.

7] K. Li and X. Zhang, "A new fingertip detection and tracking algorithm and its application on writing-in-the-air system," 2014 7th International Congress on Image and Signal Processing, 2014, pp. 457-462, doi: 10.1109/CISP.2014.7003824.

8] Y. Chen, P. Su and F. Chien, "Air-Writing for Smart Glasses by Effective Fingertip Detection," 2019 IEEE 8th Global Conference on Consumer Electronics (GCCE), 2019, pp. 381-382, doi: 10.1109/GCCE46687.2019.9015389.

9] C. T. Tolentino, A. Uy and P. Naval, "Air Drums: Playing Drums Using Computer Vision," 2019 International Symposium on Multimedia and Communication Technology (ISMAC), 2019, pp. 1-6, doi: 10.1109/ISMAC.2019.8836175.