

MICA

A Gesture Controlled Desktop Assistant

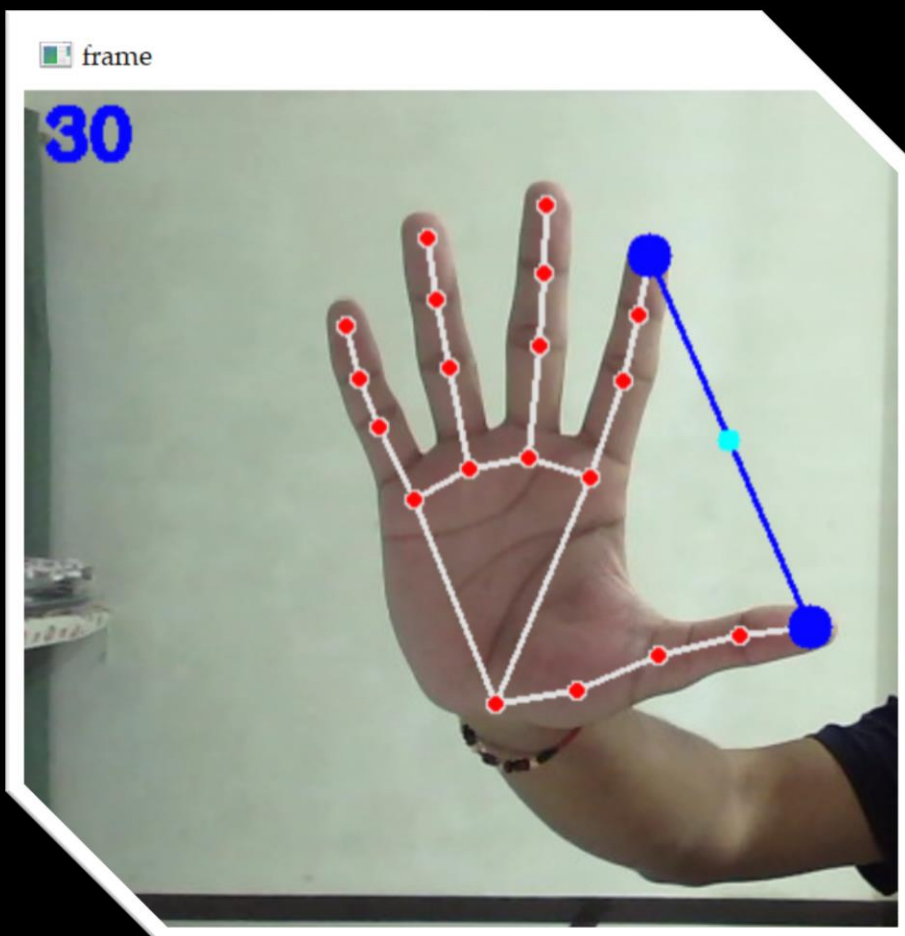
The program is created for the ease of users to access their favorite applications at the movement of a gesture, more appropriately, alphabets. Equipped with Convolutional Neural network (VGG model), the program is able to identify alphabetical gestures with an accuracy of 99.45 percent.

The model requires fast hardware to execute special commands with ease. Also, the model has the feature to execute multiple commands at the same time. Perfect for saving the time and for repeating processes, the model can execute all commands one needs by the control of a single gesture.

Working :-

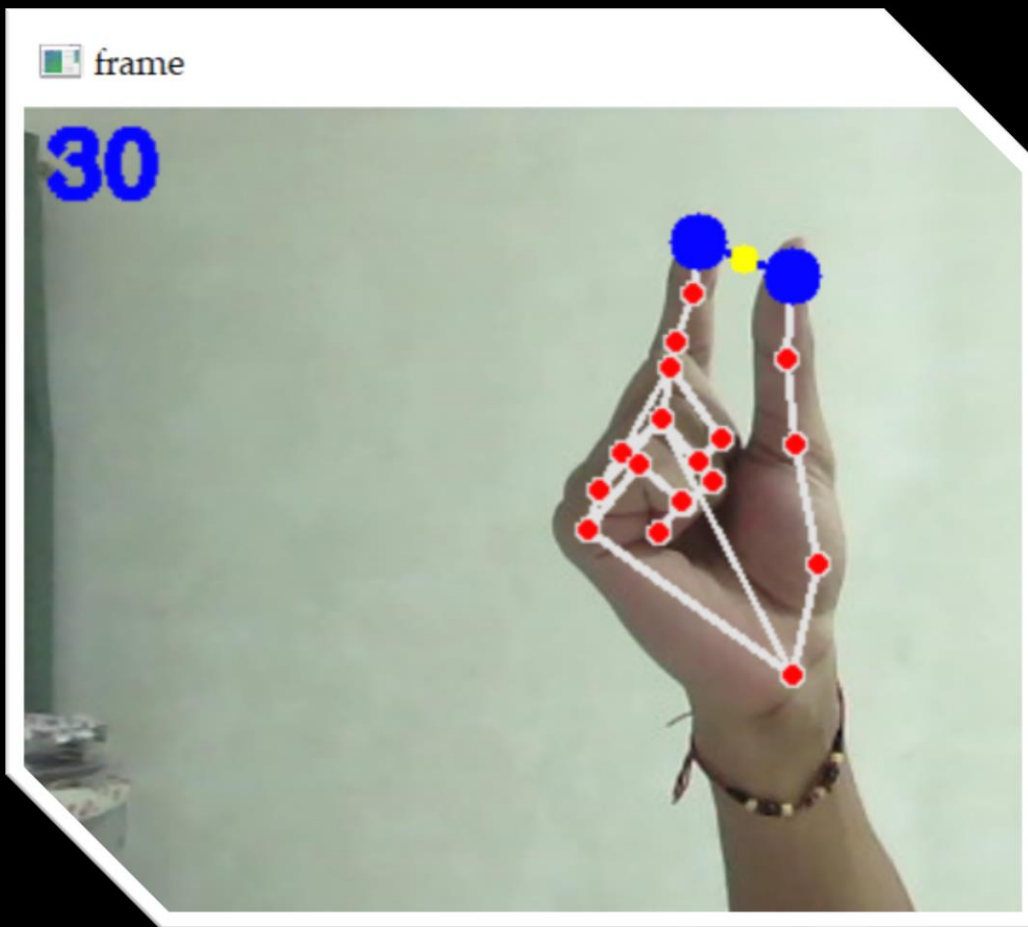
Step 1 :-

The program initializes with a video screen and recognizes various hand landmarks with the help of mediapipe library. This is what the **M stands for In MICA**. The MediaPipe library provides the position of all the hand landmarks.



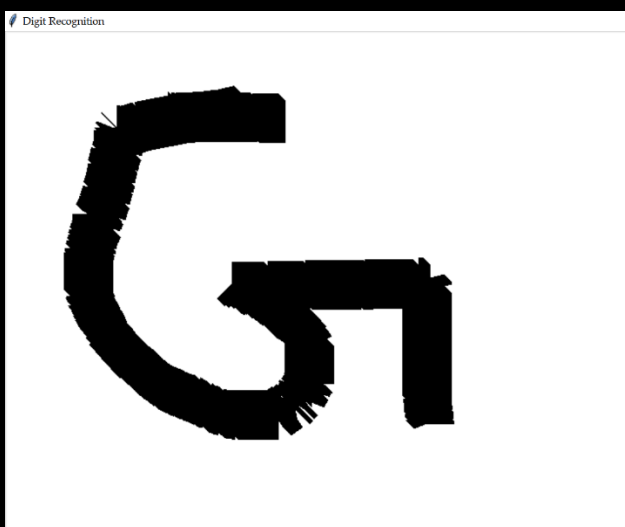
Step 2 :-

The next step is to put the index as thumb finger closer like holding a pencil. This allows the user to draw the gestures. The drawing phase is denoted by a yellow circle.



Step 3:- Once the circle turns yellow, trace the outline of any alphabet you want to draw.

For example, G is drawn as :-

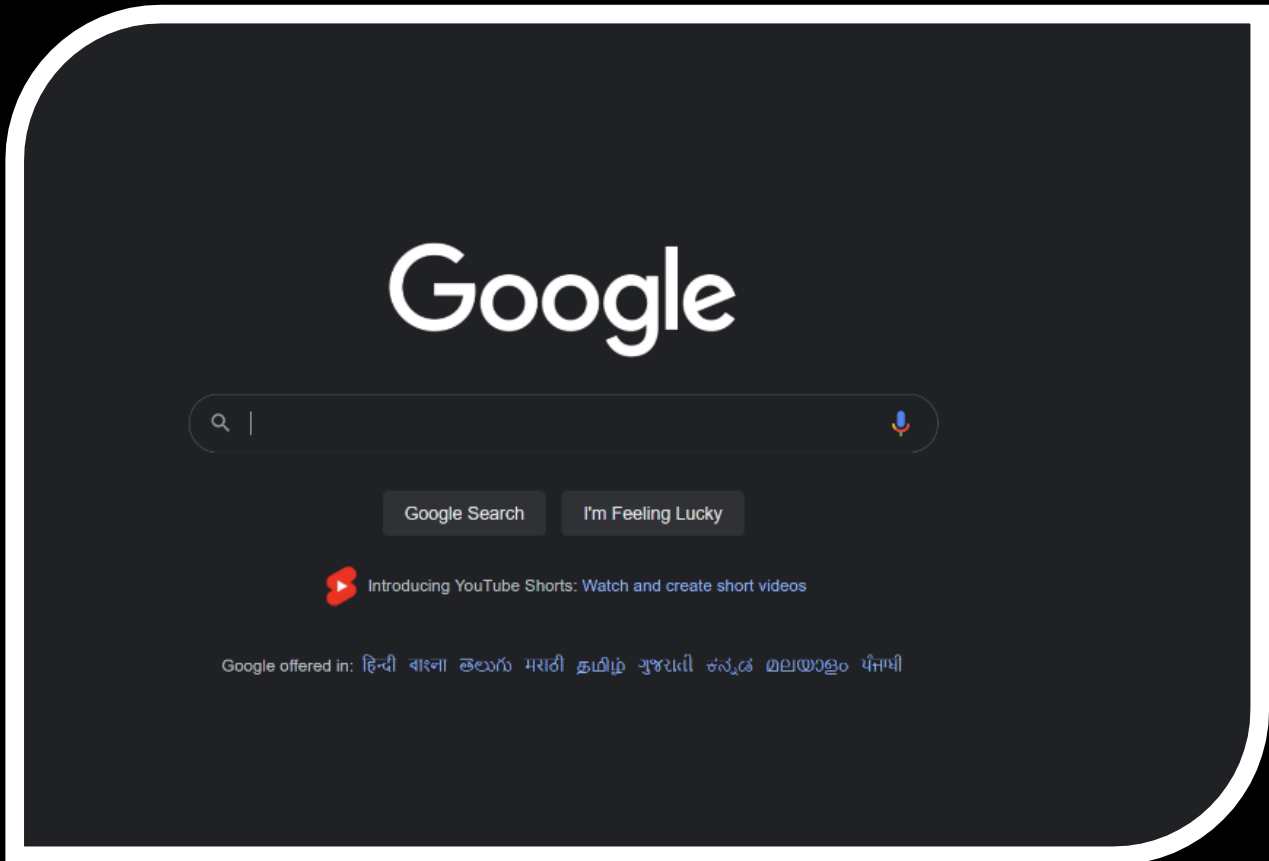


This part is interaction, **The I in MICA**

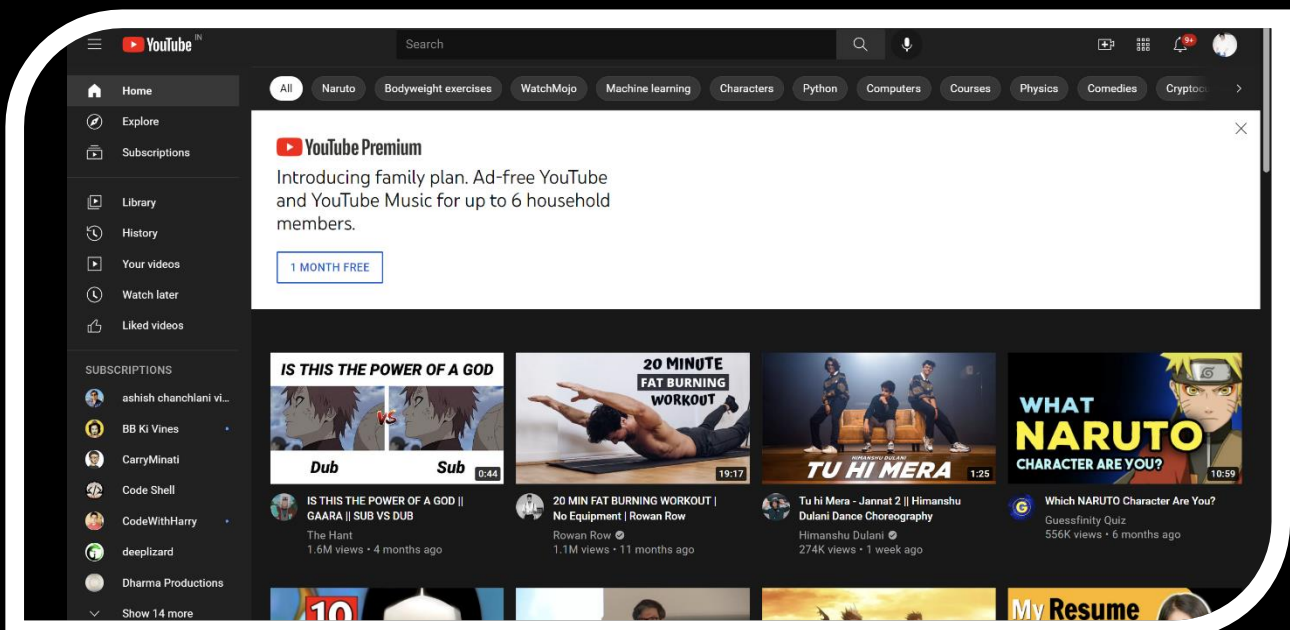
Step 4:-

The user can discard the gesture to draw a new one, open the camera or draw another gesture with the mouse, as per convenience. If the alphabet drawn is readable, the deep learning model recognizes the alphabet for you and executes the command it is programmed to.

In this case, drawing **G**, will open **Google** :-



Drawing **Y**, will open **YouTube** :-



Similarly, Till now, the following alphabets are assigned to following :-

G – Google

Y – Youtube

P – Paint

C – Cmd

S – Studies Folder

W – Wikipedia

E – Email

The actions can be updated as per the user's preferences.

Implementation :-

1. The OpenCV library starts the Video Capture.
2. The MediaPipe library detects the HandLandMarks on every frame and renders it.
3. The location of each landmark is stored which is used to access the pencil gesture.
4. Once drawn, the trace of each pixel point is stored and rendered by the algorithms on a white screen.
5. The Image generated is stored.
6. The user has the option to discard the image by <BackSpace>, Predict the image by <Enter>, or start the camera again with <Space>. Also the user can draw its own gestures using mouse control.
7. Once <Enter> is pressed, the model is sent to the brain(CNN) to predict the letter.
8. The model executes actions according to prediction.

Uses, Features and Future aspects :-

- The model is able to execute multiple operations as per user needs. The user just needs to code in as per requirement. This improves speed and reduces time taken for tasks.
- The model is capable of predicting upto more than 99% accuracy. Accuracy combined with speed makes the model capable.
- For future aspects, the model will be converted into a start-up app to execute as soon as the computer is switched on, so as to provide better assistance.
- Moreover, the option of Voice input and control will also be implemented.

Other Notes :-

- Dataset link :- <https://www.kaggle.com/sachinpatel21/az-handwritten-alphabets-in-csv-format>
- How to run:- Run GUI.py File
- If you want to train your own model, propose changes to try1.ipynb