HAND GESTURE RECOGNITION

PROJECT REPORT

18CSE390T – COMPUTER VISION

(2018 Regulation)

III Year/ V Semester

Academic Year: 2022 -2023

By

Ajay Abhinav (RA2011026010280)

Varsha.S (RA2011026010286)

Tharun Reddy (RA2011026010299)

Mehvish Ahmed (RA2011026010302)

Under the guidance of

Dr. R. BEAULAH JEYAVATHANA

Associate Professor

Department of Computational Intelligence

FACULTY OF ENGINEERING AND TECHNOLOGY

SCHOOL OF COMPUTING

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

Kattankulathur, Kancheepuram

NOVEMBER 2022

BONAFIDE

This is to certify that **18CSE390T – COMPUTER VISION project report** title **"Hand Gesture Recognition"** is the bonafide work of **Ajay Abhinav(RA2011026010280) Varsha.S (RA2011026010286),Tharun Reddy (RA2011026010299), Mehvish Ahmed (RA2011026010302)** who undertook the task of completing the project within the allotted time.

Signature of the Guide

Dr. R. Beaulah Jeyavathana

Assistant Professor

Department of CINTEL

SRM Institute of Science and Technology

ABSTRACT :

Hand gestures are a form of nonverbal communication that can be used in several fields such as communication between deaf-mute people, robot control, human–computer interaction (HCI), home automation and medical applications. Research papers based on hand gestures have adopted many different techniques, including those based on instrumented sensor technology and computer vision. In other words, the hand sign can be classified under many headings, such as posture and gesture, as well as dynamic and static, or a hybrid of the two. In addition, it tabulates the performance of these methods, focusing on computer vision techniques that deal with the similarity and difference points, technique of hand segmentation used, classification algorithms and drawbacks, number and types of gestures, dataset used, detection range (distance) and type of camera used. The ability to perceive the shape and motion of hands can be a vital component in improving the user experience across a variety of technological domains and platforms. For example, it can form the basis for sign language understanding and hand gesture control, and can also enable the overlay of digital content and information on top of the physical world in augmented reality.

MODULE:

The basic module of Hand Gesture Recognition consists of four steps: **Image Acquisition, Hand Segmentation, Feature Extraction, and Hand Gesture Recognition**

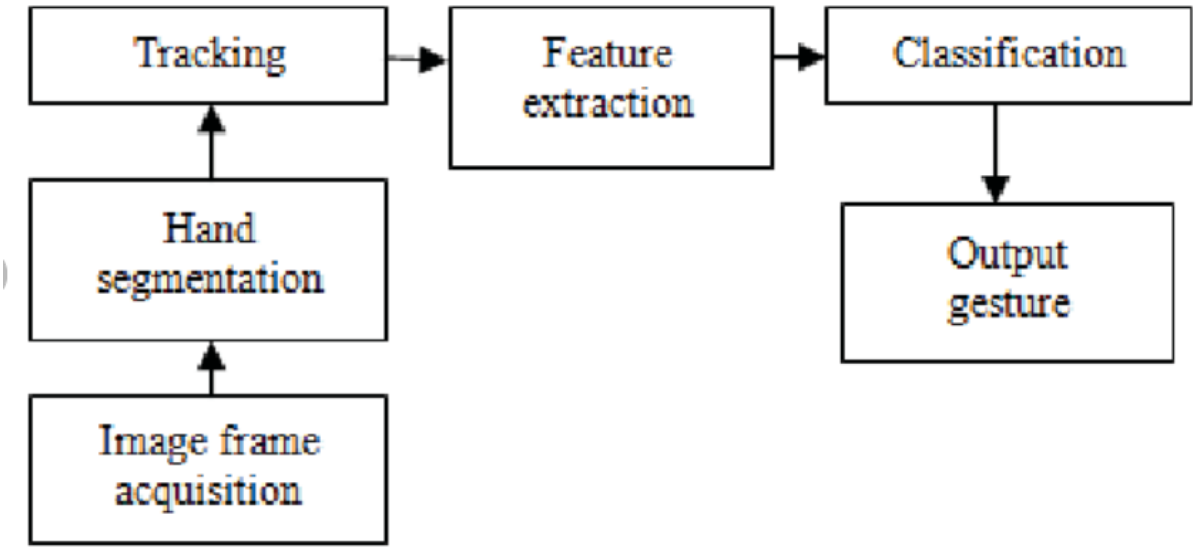
Image acquisition is the process of capturing images for vision-based approach.

Hand segmentation is a process to isolate hands and other features from the rest of the image in vision based systems. Canny Edge Detector is used to detect the edges of hands from an image. Canny Edge Detector is known for its optimal performance in detecting edges and low error rate.

Feature extraction is used to acquire features from the images captured. The features include background data, translation, scale, shape, rotation, angle, coordinates, and movements . Invariant Feature Transform (SIFT) algorithm had been used as feature extraction methods SIFT algorithm was robust against rotation, translation, or scaling variation, and produced a large collection of local feature vectors.

Classification is the final stage and an important stage in gesture recognition Word or sentence. in sign language is made up of continuous gesture, which changes over time. These applications implemented hand gesture recognition system methods to perform the basic function of the application. One example of a sign language recognition system utilized hand gesture recognition module.

SYSTEM ARCHITECTURE:



Firstly, the image frame is acquired from the input video. The next step is segmentation which partitions an image into its constituent parts or objects. Hand tracking is a high-resolution technique that is employed to know the consecutive position of the hands of the user. After the successful tracking there is a need to extract the important feature points from the available data points of the track path. In pattern recognition and in image processing, feature extraction is a special form of dimensionality reduction . After feature extraction, classifier plays a vital role in gesture recognition process. Classifier isa statistical method that takes feature set as input and gives a class labeled output, which are required output gestures.

DATASET DESCRIPTION:

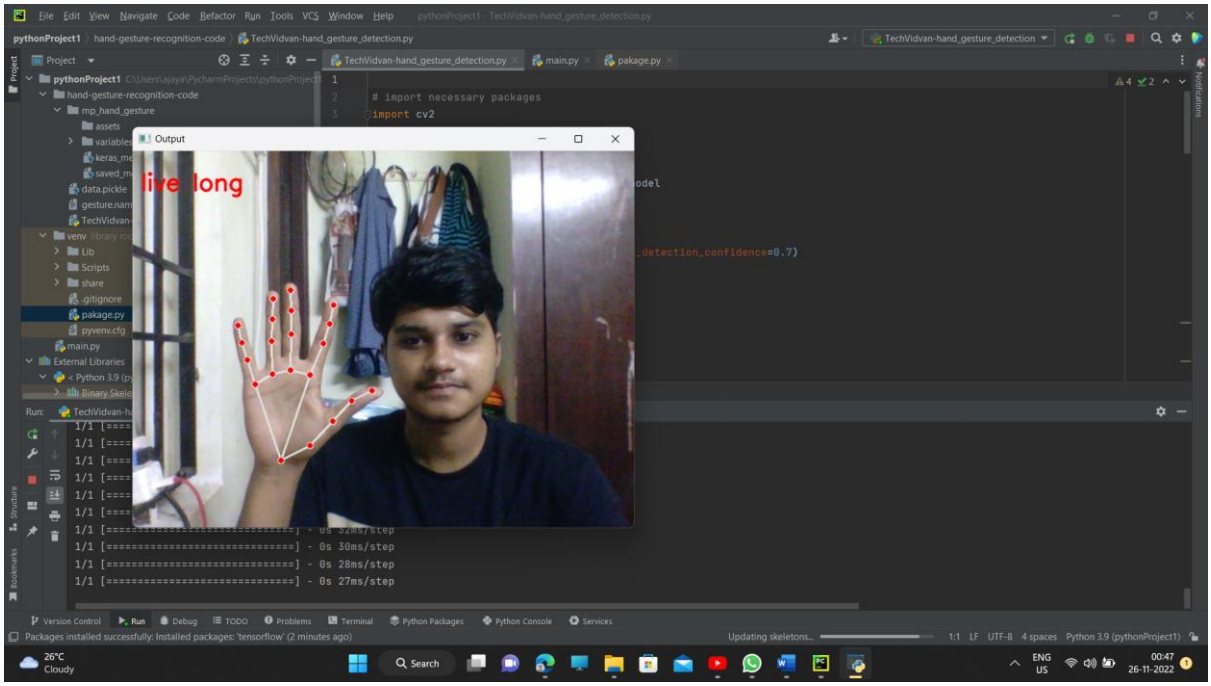
This dataset contains 509,153 training images from HaGRID (HAnd Gesture Recognition Image Dataset) modified for image classification instead of object detection. The original dataset is 716GB. I created this sample so readers can use the dataset in the free tiers of Google Colab and Kaggle Notebooks.

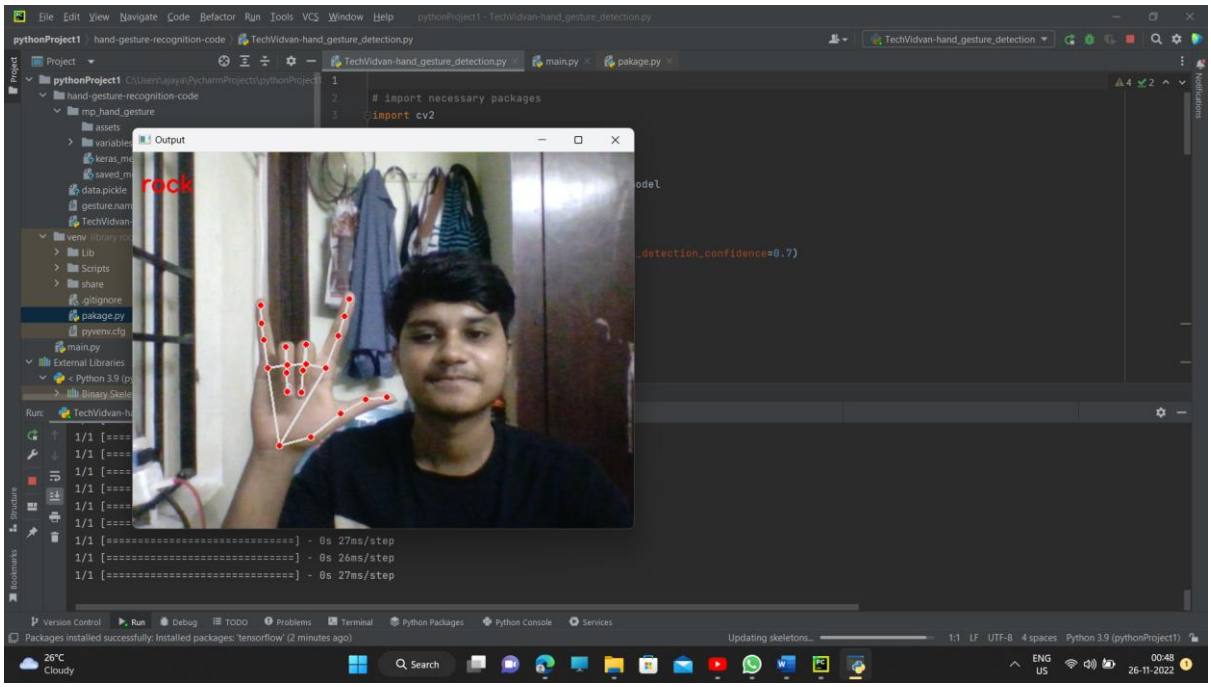
https://www.kaggle.com/datasets/innominate817/hagrid-classification-512p

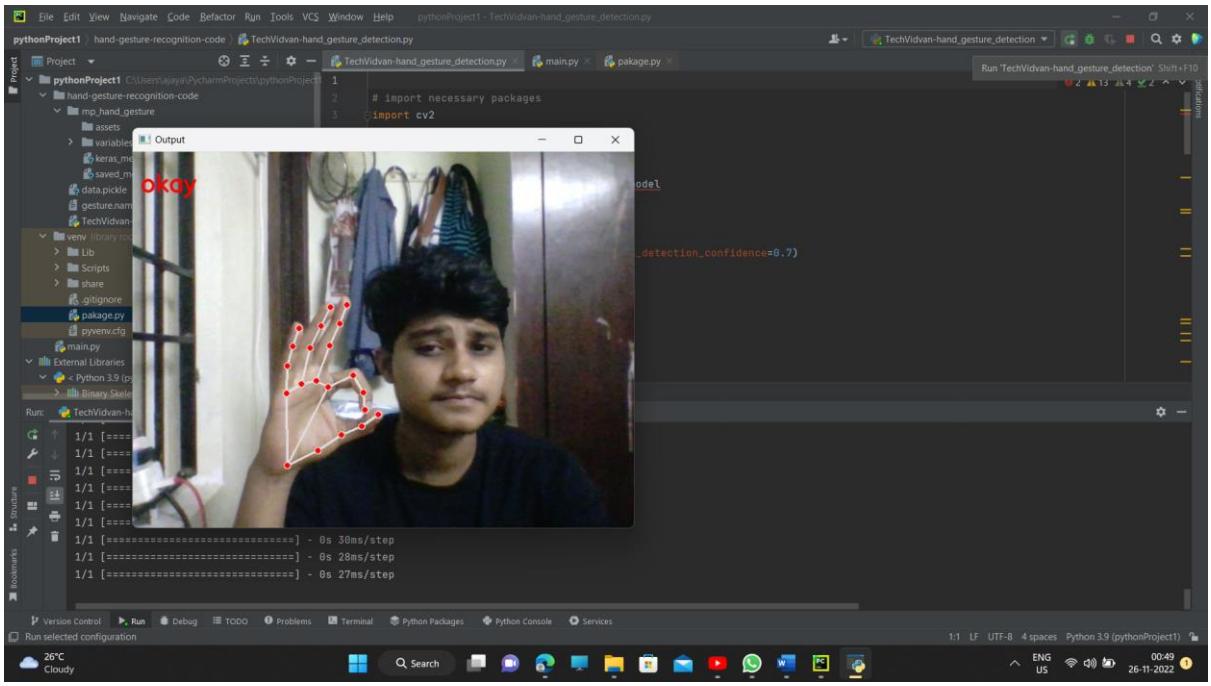
CODE:

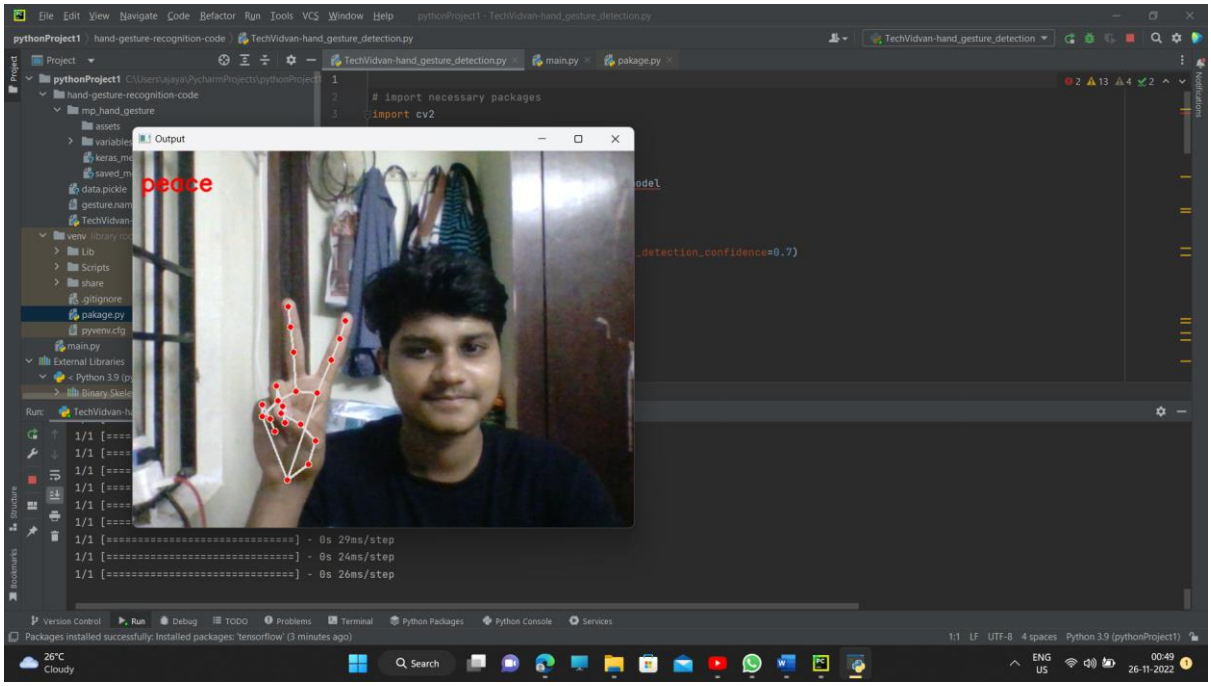
```
# import necessary packages
import cv2
import numpy as np
import mediapipe as mp
import tensorflow as tf
from tensorflow.keras.models import load_model

# initialize mediapipe
mpHands = mp.solutions.hands
hands = mpHands.Hands(max_num_hands=1, min_detection_confidence=0.7)
mpDraw = mp.solutions.drawing_utils

# Load the gesture recognizer model
model = load_model('mp_hand_gesture')

# Load class names
f = open('gesture.names', 'r')
classNames = f.read().split('\n')
f.close()
print(classNames)


# Initialize the webcam
cap = cv2.VideoCapture(0)

while True:
    # Read each frame from the webcam
    _, frame = cap.read()

    x, y, c = frame.shape

    # Flip the frame vertically
    frame = cv2.flip(frame, 1)
```

```
    framergb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

    # Get hand landmark prediction
    result = hands.process(framergb)

    # print(result)

    className = ''

    # post process the result
    if result.multi_hand_landmarks:
        landmarks = []
        for handslms in result.multi_hand_landmarks:
            for lm in handslms.landmark:
                # print(id, lm)
                lmx = int(lm.x * x)
                lmy = int(lm.y * y)

                landmarks.append([lmx, lmy])

            # Drawing landmarks on frames
            mpDraw.draw_landmarks(frame, handslms,
mpHands.HAND_CONNECTIONS)

            # Predict gesture
            prediction = model.predict([landmarks])
            # print(prediction)
            classID = np.argmax(prediction)
            className = classNames[classID]

    # show the prediction on the frame
    cv2.putText(frame, className, (10, 50), cv2.FONT_HERSHEY_SIMPLEX,
                   1, (0,0,255), 2, cv2.LINE_AA)
cv2.destroyAllWindows()
```




Output
rock
# import necessary packages
import cv2
1/1 [==============================] - 0s 27ms/step
1/1 [==============================] - 0s 26ms/step
1/1 [==============================] - 0s 27ms/step


Output
okay
# import necessary packages
import cv2
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 28ms/step
1/1 [==============================] - 0s 27ms/step



CONCLUSION :

Hand gesture recognition is of great importance for human computer interaction (HCI) because of its extensive applications in virtual reality and sign language recognition etc.so Hang Gesture recognition implemented successfully.