

FINAL REPORT

HAND GESTURE RECOGNITION

Problem Statement

We know conducting examination in a smoothly manner is a crucial task for an invigilator. In examination students do require some sort of help from the invigilator like asking for a supplementary, requesting for going to toilet, help on some doubt related to question paper/or anything or might have completed his/her paper and want to inform the invigilator. While students might inform by speaking to the invigilator from his/her place, it might happen to disturb the other classmates who are writing the paper. Also, in a large classroom it might be tedious task for a student to interact with the invigilator instantly, and we know time is a crucial factor in examinations.

But to solve this problem, instead of students interacting with invigilators via speech they can use hand signals to inform the invigilator to get the help they need. But here also the invigilator has to be alerted all the time to look for any signs from the students and also, he/she might have to look carefully for the signs which the student is projecting towards the invigilator. For example, raising hand with fingers fully opened might say that the student wants a supplementary, or pointing out a victory sign might say that the student has completed his/her exams etc. Below mentioned some scenarios of hand signals projecting by a student to the invigilator.

For helping out invigilators in such cases where the classroom is big or to run the examination by minimal disturbance to the students, we can use the hand gesture recognition task by a surveillance camera. The gesture made by students will be picked up by the camera and thus the invigilator will be informed by the same into his/her phone via an app.



Fig. Need Help



Fig. Need Supplementary paper



Fig. Completed the exams



Fig. Requesting for going to toilet

Project Overview

In this hand gesture recognition project, we created a hand gesture detector which detects various signs like help, supplementary, toilet and completed and it can be extended to cover a vast multitude of other signs and hand gestures.

This is an application of Convolution Neural Networks combined with background elimination to detect different hand gestures. A background elimination algorithm extracts the hand image from webcam and uses it to train as well predict the type of gesture that is. We have used OpenCV and tensorflow modules of python to develop this project.

Steps to develop hand gesture recognition project

It is divided into the following three parts:

- 1) Creating a dataset
- 2) Training a CNN on the captured dataset
- 3) Predicting the data

Creating the dataset for hand gesture recognition

It is fairly possible that we can get the dataset from the internet but in this project, we will be using the dataset from our won.

We will be doing a live feed from the video camera and every frame which is detected from the region of interest (ROI) that will be saved in some directory, which will have 4 different folders for storing 4 different types of hand gestures.

For creating the dataset, we will do the live camera feed using OpenCV and create an ROI to detect the hand in for the gestures.

For differentiating between the background, we will calculate the accumulated weighted average for the background and then subtract this from the frames that contain some object in front of the background that can be distinguished as foreground.

This is done by calculating the accumulated weight for some frames (here for 30 frames) we calculate the accumulated average for the background.

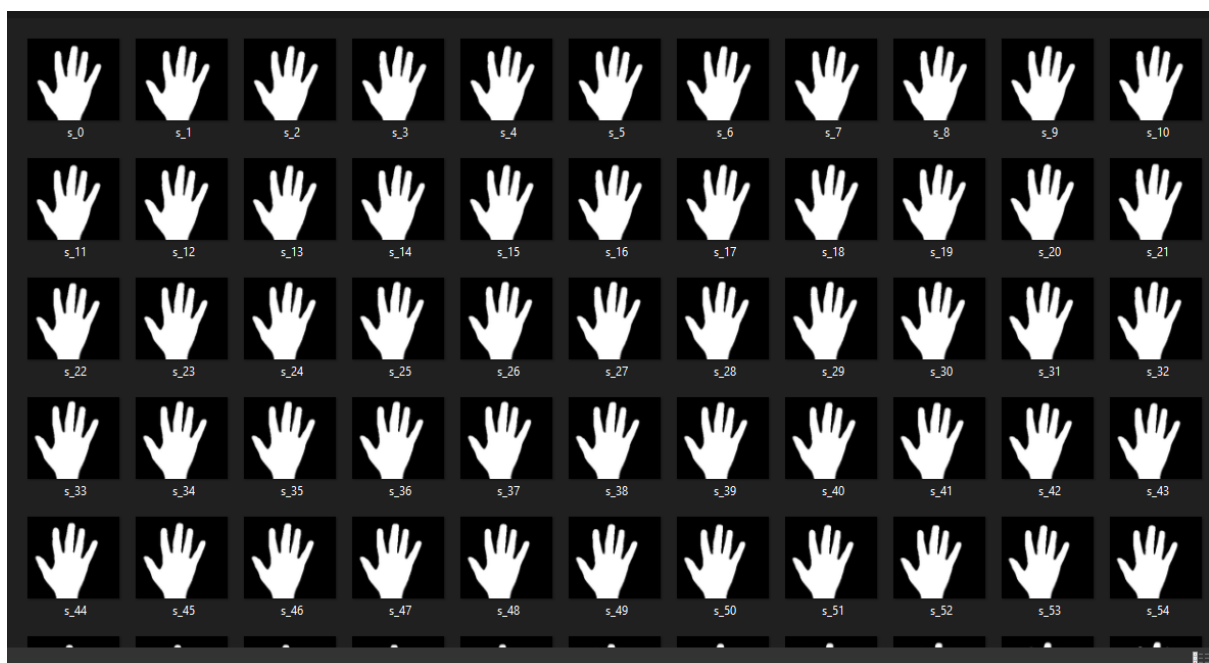
After we have the accumulated average for the background, we subtract it from every frame that we read after 30 frames to find any object that covers the background.

Calculate Threshold Value

Now we calculate the threshold value for every frame and determine the contours using `cv2.findContours` and return the max contours (the most outermost contours for the object) using the function `segment`. Using the contours, we are able to determine if there is any foreground object being detected in the ROI, in other words, if there is a hand in the ROI.

When contours are detected (or hand is present in the ROI), We start to save the image of the ROI in the different folders for the signs we are detecting it for.

We will be creating a dataset of 1000 images for each set of signs/ categories as shown below.



Resizing Images

Now we will resize the images in a particular ratio such that the base width of 100 will be fixed and the height will be adjusted according to the ratio of base width adjusted. First, we will load the images from the folders, and then apply the `resize` function to the datasets.

Training the CNN

Now on the created dataset we train a CNN.

We design the CNN depending on some trial-and-error method with hyper tuning the model.

We built network which contains 7 hidden convolution layers with Relu as the activation function and 1 Fully connected layer.

The network is trained across 50 iterations with a batch size of 64.

The ratio of training set to validation set is 1000: 100.

After compiling the model, we fit the model using 8 epochs and save the model for it to be used in the last module (GestureRecogModel.tfl)

Predict the Gesture

In this, we create a bounding box for detecting the ROI and calculate the accumulated average as we did in creating the dataset. This is done for identifying any foreground object.

Now we find the max contour and if contour is detected that means a hand is detected so the threshold of the ROI is treated as a test image.

We load the previously saved model and feed the threshold image of the ROI consisting of the hand as an input to the model for prediction.

Sign Language recognition Outputs

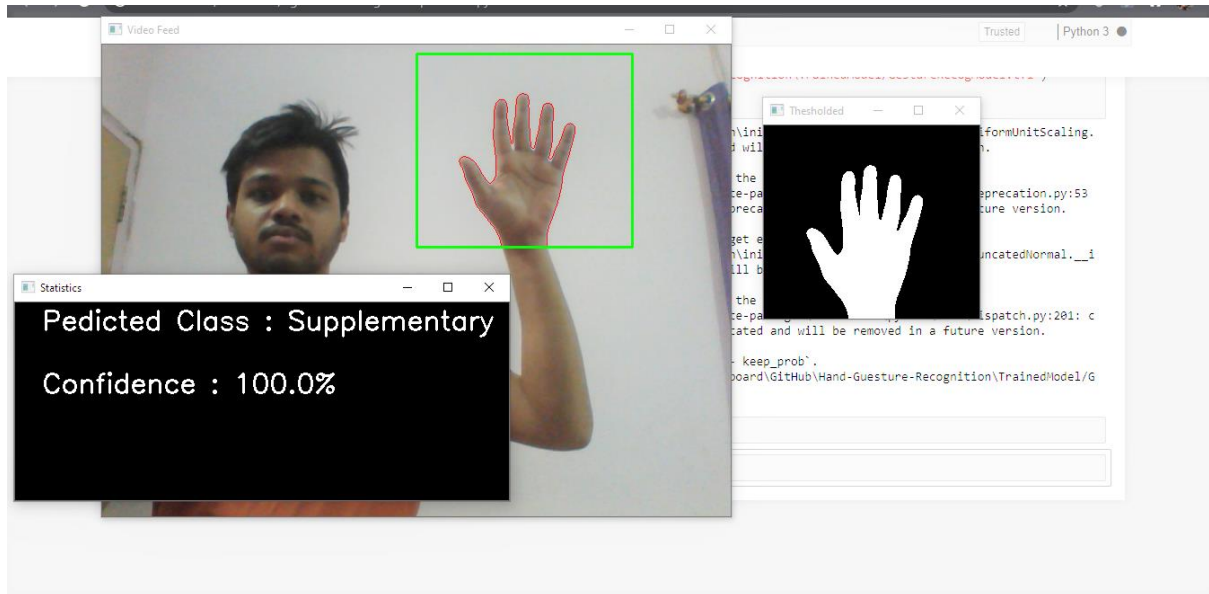


Fig. Requesting for a supplementary

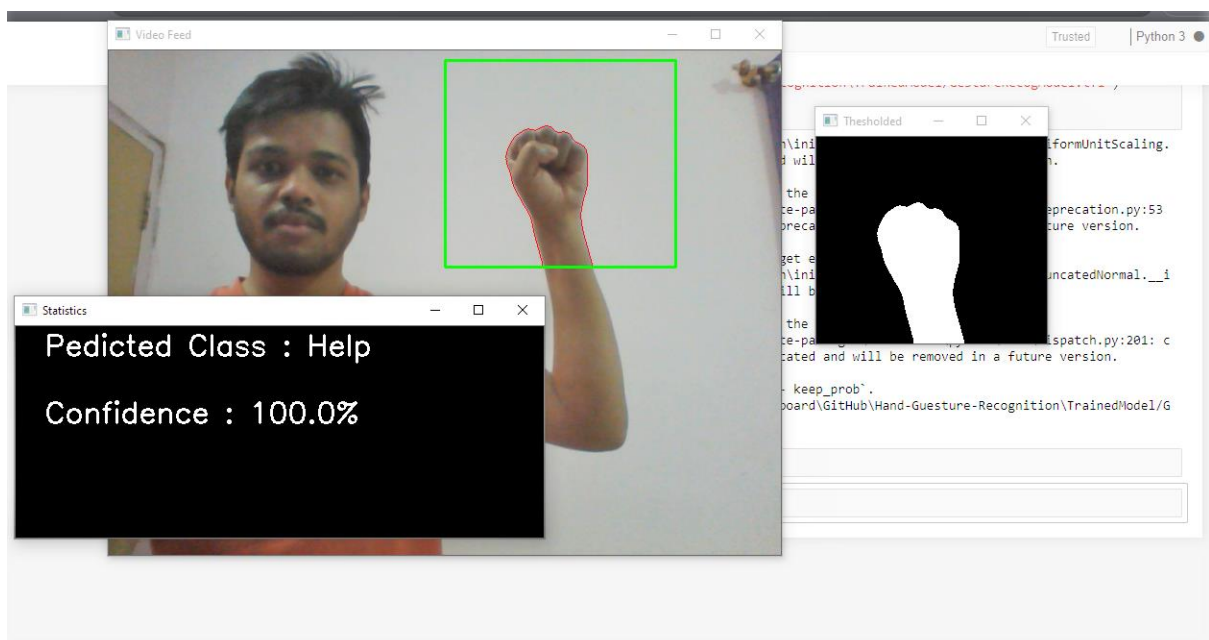


Fig. Asking for help

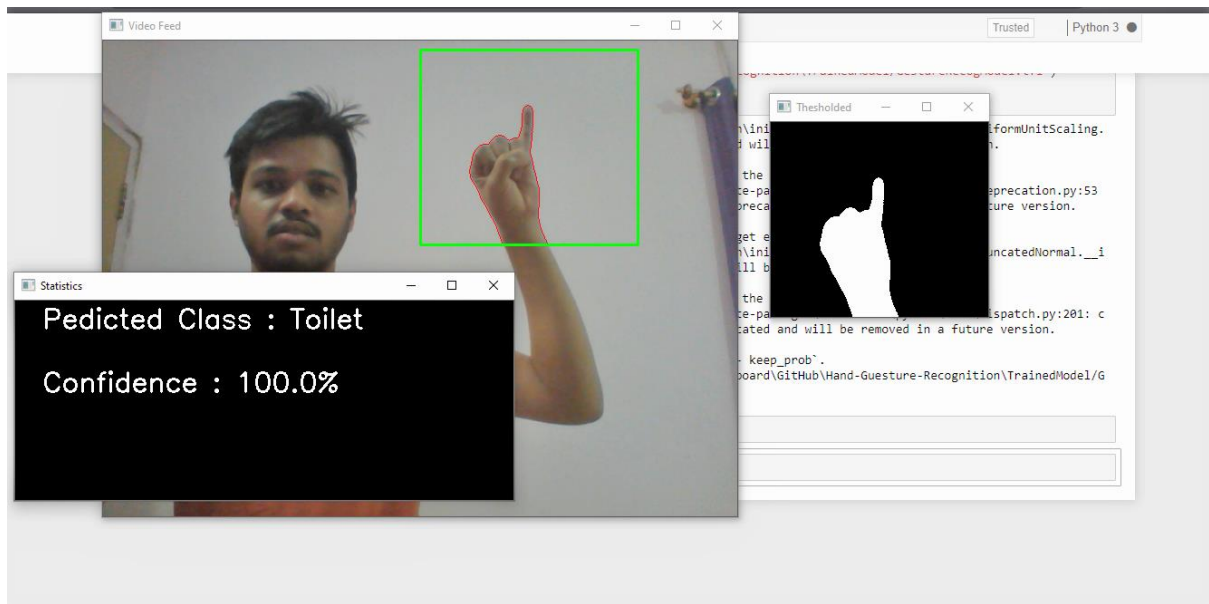


Fig. Requesting for going to washroom

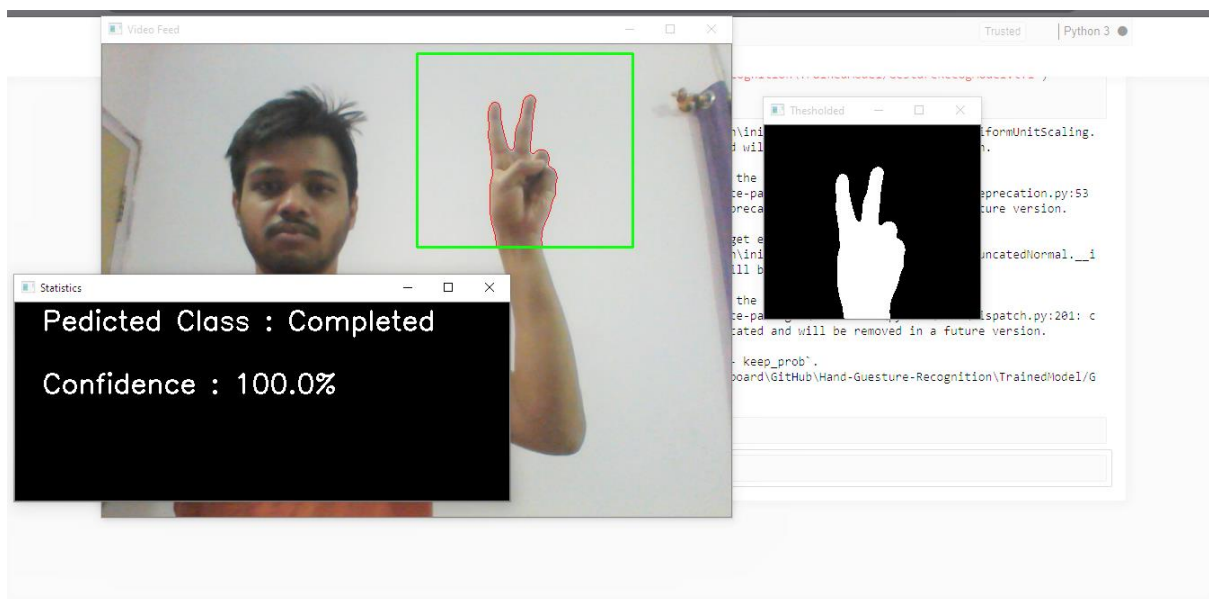


Fig. Informing that the paper is completed