Project Report

SIGN LANGUAGE DETECTION

Enrollment No.(s) 17103166,17103179,17103191

Name of student(s) Aryan Aggarwal, Rohit Ramchandani,Anmol Sharma

Name of supervisor: Vikas Hassija

Submitted in partial fulfillment of the requirement for the degree of

Bachelor of Technology

In

Computer Science Engineering

Department of Computer Science & Engineering and Information Technology

JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA

DECLARATION

We hereby declare that this project is our own work and, to the best of our knowledge and belief, does not contain any material previously published or written by any other person or material accepted for the achievement of another degree or diploma from another university or university; except in cases where this has been recognized in the text.

Place: JIIT Noida
Date: 1st May 2021

Name: Aryan Aggarwal
Enrollment No.: 17103166

Name: Rohit Ramchandani
Enrollment No.: 17103179

Name: Anmol Sharma
Enrollment No.: 17103191

CERTIFICATE

This is to certify that the work titled "**Sign Language Detection**" submitted by " **Aryan Aggarwal, Rohit Ramchandani, Anmol Sharma**" in partial fulfillment for the award of the degree of B. Tech, of Jaypee Institute of Information Technology, Noida has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Name of Supervisor: Vikas Hassija
Designation: Assistant Professor
Date: 1st May 2021

ACKNOWLEDGEMENT

We would like to express our special thanks of gratitude to our mentor Dr. Vikas Hassija for teaching us, helping us with the project as well as guiding us with the relevant resources and ultimately providing us with an opportunity to make and present this report. We would like to express our gratitude towards our parents & members of the Jaypee Institute of Information Technology for their kind cooperation and encouragement which helped us in the completion of this project.

Name: Aryan Aggarwal
Enrollment No.: 17103166

Name: Rohit Ramchandani
Enrollment No.: 17103179

Name: Anmol Sharma
Enrollment No.: 17103191

Date:1st May 2021

SUMMARY

Here we have built a project on **Sign Language Detection,** as we can see many people with disabilities who can't speak on their own.We have used the concept of Machine Learning to address this issue. We worked on PYTHON using various libraries such as OpenCV, Matplotlib, Sklearn, H5py etc. We made this project which will help the disabled person to communicate on their own using this thus reducing the effort.We also checked which of the ML algorithms would work best for this issue by comparing them such optimising the result. Overall this is a great effort to help these people to find a better place in this world.

LIST OF FIGURES

LIST OF SYMBOLS AND ACRONYMS

1.ML- Machine Learning

2.AI- Artificial Intelligence

3.CNN- Convolutional Neural Network

4.GPU- Graphics Processing Units

5.SVM- Support Vector Machine

6.SSD- Single Shot Detector

7.LR-Logistic Regression

INDEX TABLE

Chapter 1 : INTRODUCTION

1.1 General Introduction

Sign languages are a series of languages that use predefined methods and changes to convey a message. These languages were developed primarily to help Deaf and others with verbal challenges. They use a simultaneous and precise composition of hand gestures, alignment of the hands, hands, etc. Different regions have different sign languages, such as American Sign Language, Indian Sign Language, etc. We will focus on Indian Sign Language in this area.

Indian Sign Language (ISL) is a signature used in certain countries. It is sometimes known as Indo-Pakistanit Sign Language (IPSL). There are many specific features that are present in ISL that distinguish it from other character specifications. Features like family relationships, use of space, etc. There are good features of ISL. Furthermore, ISL has no temporal influence.

In this piece of project, our goal is to analyze and recognize various alphabets from the representation of drawn images. The database is made up of different images and each image is displayed in different lighting conditions with different hand orientations. With such a different amount of data, we can test our system at certain levels and thus achieve good results.

We are investigating various machine learning techniques such as Support Vector Machines (SVM), Logistic Regression, K Nearest Neighbors (KNN), and a neural network technology called Convolution Neural Networks (CNN) to recognize sign language.

1.2 Problem Statement

There is an associate degree plain communication downside between the deaf community and therefore the hearing majority. Innovations in automatic signing recognition attempt to raze this communication barrier. signing is the most vital communication method between the hearing impaired community and traditional persons. In recent years, signing has been widely

studied and supported multiple input sensors, like knowledge gloves, web camera, stereo camera, and so on. Although knowledge glove primarily based signing recognition achieves sensible performance even for big vocabularies, the device is just too overpriced to popularize. In vision-based signing recognition, the key issue is the correct and quick hand following and segmentation. However, it's terribly troublesome for the advanced backgrounds and illuminations.

The project aims to create a machine learning model that can be used to categorize the different hand gestures that are used to write fingers in sign language. In this user-independent model, the machine learning classification algorithms are trained on a single image data set, and the test is performed on a completely different data set. For the image data set, depth images are used which, due to the reduced pre-processing time, provided better results than some of the previous publications. Various machine learning algorithms are applied to the data sets.

1.3 Significance/Novelty of the problem

This problem is very important in present time as we can see there are many people with disabilities and they can’t talk without someone conveying their message, so this project is focused to reduce man power as they can use this to convey their message on their own.
Also we compared various algorithms so check which one gives us the best result.

1.4 Empirical Study

We have worked on the followings tools:

1. PyCharm offers code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCS) and supports data science with Anaconda. PyCharm is cross-platform with versions of Windows, macOS, and Linux.

2. Pillow is a Python Image Library (PIL) that you can use to open, edit, and save images. The current version identifies and reads a large number of formats. Writing support is intentionally limited to the most widely used interchange and presentation formats.

3. Joblib is a set of tools for providing a lightweight pipeline in Python. Joblib offers in particular: transparent caching of output values on the hard disk and lazy reevaluation (storage pattern).

4. An HDF5 file is a container for two types of objects: records that are matrix-based databases, and groups that are folder-shaped containers that contain records and other groups. The most basic things to consider when using h5py are: Groups act as dictionaries and records as NumPy fields.

1.5 Preliminary Information

Machine Learning

Machine Learning (ML) is the look at visual algorithms that, via experience, enhance automatically. It is visible that it is a synthetic intelligence sub-set. In order to make predictions or choices without being in particular programmed to do so, device mastering algorithms assemble a mathematical version primarily based totally on pattern statistics, acknowledged as "schooling statistics". In a huge variety of applications, including e-mail filtering and laptop vision, device mastering algorithms are used wherein it's far tough or not possible to create conventional algorithms to carry out the obligations required. Machine mastering, which specializes in making predictions of the usage of computers, is carefully associated with computational statistics. The evaluation of mathematical optimization presents the sector of device mastering with techniques, concept and alertness domains. Data mining is an associated location of research, concentrating via unsupervised mastering on exploratory statistics processing. In its utility throughout enterprise problems, device mastering is likewise known as predictive analytics.

Computer Vision

Computer vision is an interdisciplinary area of technology that addresses how computer systems can advantage and know-how of virtual photos or motion pictures at an excessive stage. Computer imaginative and prescient duties consist of strategies for collecting, processing, analysing , and decoding virtual photos and extracting excessive-dimensional records from the actual international so one can generate numerical or symbolic understanding, e.g. withinside the shape of decisions, decoding on this sense, from the engineering viewpoint, seeking to recognize and automate duties that the human visible device can do. This interpretation of the photograph may be visible because the disengagement of symbolic understanding from photographs records the usage of fashions advanced with the help of geometry, physics, statistics, and the principle of learning.
The principle in the back of synthetic structures that extract understanding from photos worries the clinical subject of laptop imaginative and prescient. There are many sorts of photograph records, including video sequences, perspectives from a couple of cameras, multidimensional 3-d scanner records, or scientific scanning devices. To assemble laptop imaginative and prescient structures, the technological subject of laptop imaginative and prescient ambitions to use its theories and fashions. Computer imaginative and prescient is an interdisciplinary vicinity coping with how machines may be created from virtual photos or motion pictures to attain an excessive-stage know-how. It ambitions to automate duties that the human visible device can do from the angle of engineering.

Computer vision involves extracting, analyzing, and mechanically understanding helpful data from one image or series of pictures. This includes the event of a theoretical and recursive basis for achieving automatic visual comprehension. computer vision as a bailiwick deals with a theory supporting artificial systems that separate data from pictures. Image information will take several forms, e.g. Video footage, multi-camera views, or use of space. As a technology discipline, computer vision seeks to use its theories and models to the development of artificial visionory systems.

Deep Learning

Methods of deep learning are trying to study characteristic hierarchies with functions created through the composition of decreasing stage functions from better tiers of the hierarchy. At

more than one tiers of abstraction, computerized mastering functions permit a machine to study complicated features that map the enter to the output immediately from statistics, without depending absolutely on human-crafted functions. In order to find out top representations, frequently at more than one tiers, with better-stage found out traits laid out in phrases of decrease-stage traits, deep mastering algorithms try to take advantage of the unknown shape withinside the enter distribution. The hierarchy of ideas allows the gadget to study complicated ideas through building them out of less difficult ones. If we draw a graph displaying how those definitions are built on the pinnacle of every difference, with numerous layers, the graph is deep.We call this technique to AI deep mastering for this purpose. Deep mastering excels in hassle regions in which the inputs are analogous (and additionally output). In different words, they're now no longer only some quantities in a tabular format, however as a substitute pixel statistics images, textual content statistics files or audio statistics files. Deep mastering allows computational fashions that encompass more than one layers of processing to study statistics representations with more than one abstraction tiers.

OpenCV

OpenCV (Open Source Computer Vision Library) is a software program library for open supply laptop imaginative and prescient and device mastering. OpenCV has been created to offer a shared infrastructure for programs for laptop imaginative and prescient and to hurry up using device notation in patron products. OpenCV, as a BSD-certified software program, makes it smooth for businesses to apply the code and extrade it. There are more than 2500 optimised algorithms withinside the collection, which includes a complete variety of laptop imaginative and prescient and device mastering algorithms, each conventional and state-of-the-art. These algorithms may be used to perceive and realise faces, perceive objects, classify human behaviour in videos, reveal digital digicam movements, reveal transferring objects, extract 3-D item models, create 3-D factor clouds from stereo cameras, sew photos collectively to provide a whole scene with a excessive decision photo, locate comparable photos from an photo database, get rid of purple eyes from photos interested in flash OpenCV has a person base of greater than 47,000 humans and an expected 18 million downloads.

Chapter 2 : LITERATURE SURVEY

CNN's latest image ranking models are judged by their performance in the ImageNet Challenge. They have gradually evolved since AlexNet was launched in 2012 (Krizhevsky et al., 2012), with changes to the architecture that improved performance. In 2014, Szegedy et al., 2015) introduced GoogLeNet, an improvement over AlexNet, mainly by greatly reducing the number of parameters involved. In 2014 Simonyan & Zisserman (2014) introduced VGGNet, which performed well due to the depth of the network. More recently, in 2015 (He et al., 2015b), ResNet was introduced, which uses "skip connections" and batch normalization. The performance of these models in ImageNet is shown here. Error rates for ImageNet Challenge. Top 5 model AlexNet error rate 15.3% GoogLeNet 6.67% VGGNet 7.32% ResNet 5.71%.

Now We want to compare all possible classifiers to get the best accuracy for our basic sign language detection stuff. Last time for the major project ‘Face mask detection’ made by the same group of ours was only went through CNN and we got an accuracy of 91.31% which was quite good but now as we used SVM for real time analysis we are getting 94.2% accuracy , that’s why we are going to compare CNN, LR(Logistic Regression), KNN and SVM for this time.

After reading all the papers we came to a conclusion that there are still many areas that we need to improve in order to make Sign Language detection more robust. The knowledge we gathered while going through the material has helped us to reach the point of the project right now.

The current work is only the first step in defining an environment and a user-independent sign language. The removal of additional pictograms is an issue for the future. The texture and edges are interesting as the hands have no noticeable texture and are limited by borders. Further work also deals with the adaptation of the color model to the user during operation. Monitoring and prediction methods are also used to deal with occlusion and interference.

Then feature extraction is used based on appearance and appearance, resulting in user and user modeling.

Chapter 3 : REQUIREMENT ANALYSIS AND SOLUTION APPROACH

3.1 Project Description

In this project, we are building a Indian Sign language detector for the deaf as these people need someone to convey a message and with the help of this software we can reduce the effort. Also we are comparing various algorithms such as so that we can find out which of these algorithms is the best to be used considering the factors such as the cost of running, time and effort.

3.2 Requirement Analysis

Tools and Environment Used

1. Sublime Text Editors
2. Virtual Environment
3. Python IDE
4. PyCharm

Python Libraries used

1. Numpy
2. OpenCV
3. Matplotlib
4. H5py
5. Pillow
6. Scipy
7. tqdm
8. Sklearn
9. Joblib
10. Keras

3.3 Solution Approach

Image Preprocessing

We have used 6 layers of preprocessing which are combined by 3 layers of convolutional layers and 3 of maxpooling layers. We start to preprocess with convolutional 2D layer first then it is processed through maxpooling layer and this process is further repeated 2 more times.

We have taken a huge dataset to train our model because hands are 3D and to recognize the gestures is comparatively harder.

Segmentation

The main motive for this is to make the background that way to recognize the sin colour of our hand . For this we decided to convert the RGB image to HSV and then what if lighting changes, so we came up with an idea to be able to the HSV values for time being, and we applied a proper trackbar for HSV values in our testing model.

Classification

The details we got from each image differ in number with the same dimension (64). However, a multi-class SVM requires uniform dimensions of the feature vector as input. For a series of descriptors, the grouping of K means classifying the number of details giving things in K numbers from the center of the group. And thought K as of now =10.

The grouped features form the visual vocabulary, with each feature corresponding to an individual sign language gesture. With visual vocabulary, each image is represented by the frequency of occurrence of all of the grouped features,in this case the histogram of 26 classes of sign language gestures.

Classifiers

As we train the model first with a single algo using some layers in OpenCv. Following classifiers are used :

- CNN (Convolutional Neural Network)
- KNN (K - Nearest Neighbouring)

- SVM (Support Vector Machines)
- LR (Logistic Regression)

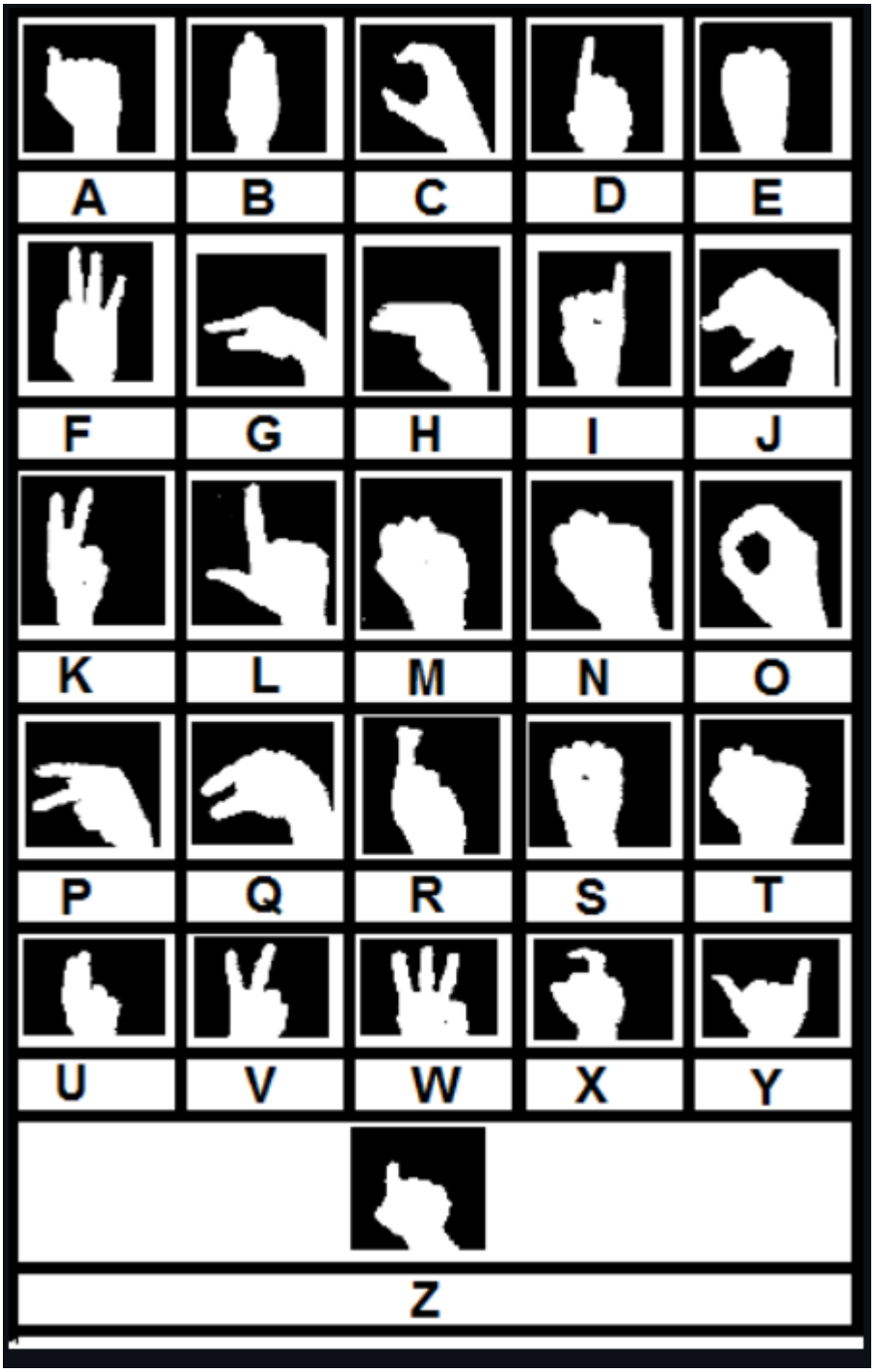

Figure 1 : Hand gesture

Chapter 4 : MODELLING AND IMPLEMENTATION DETAILS

4.1 Design Diagram

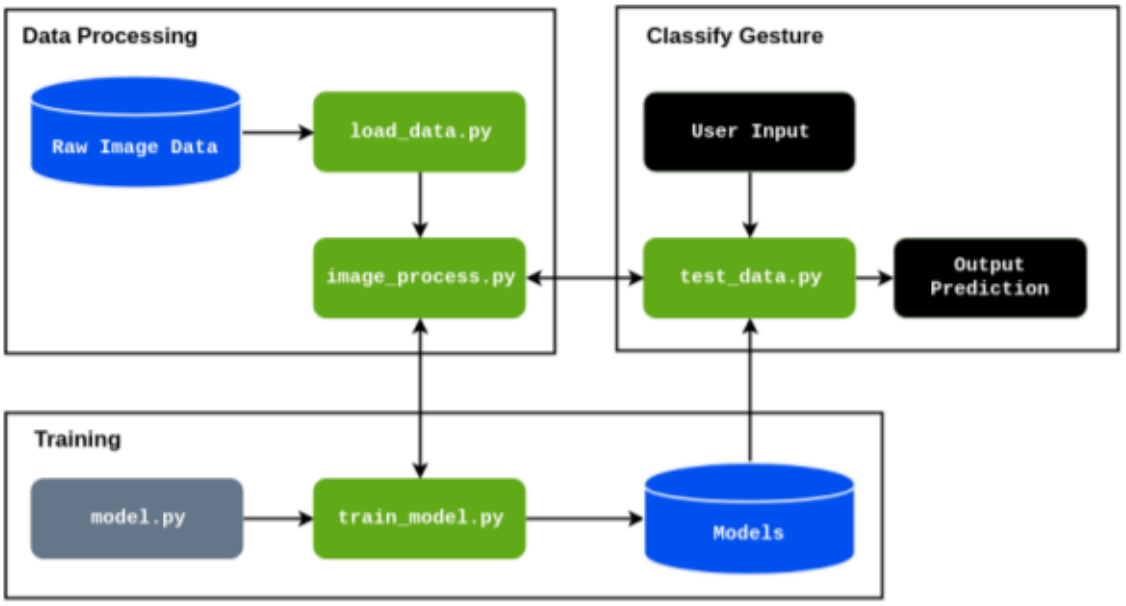

Figure 2 : Process Diagram

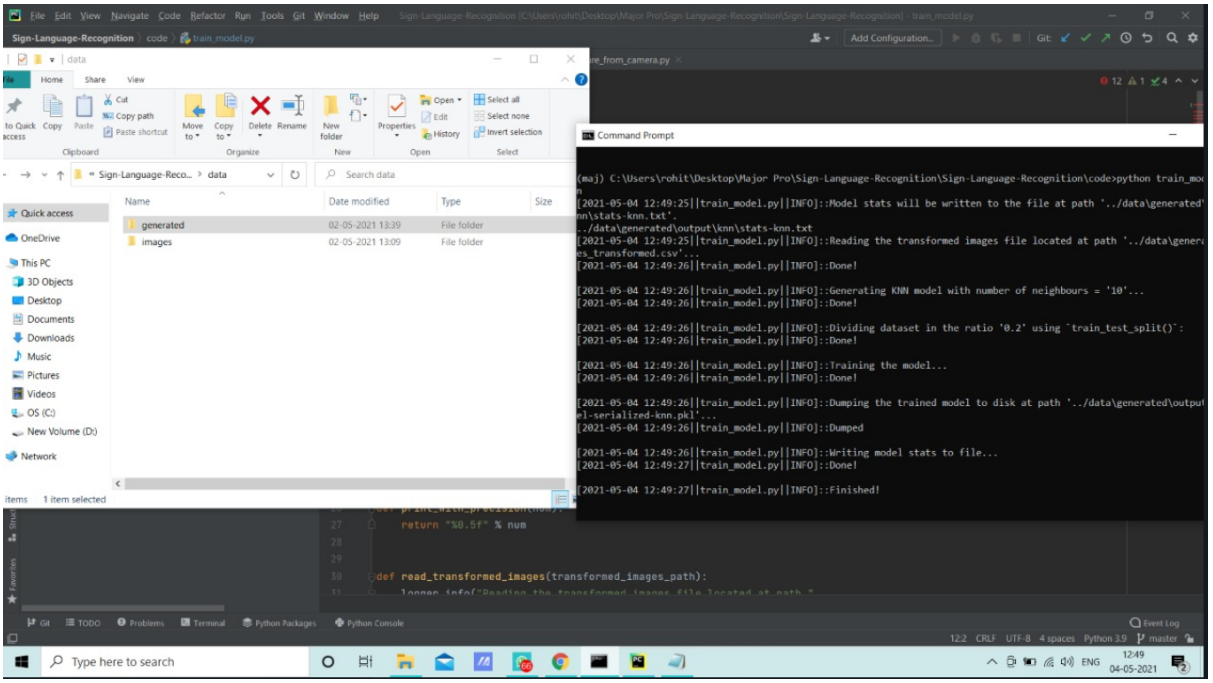

Figure 3 : Training Data (KNN)

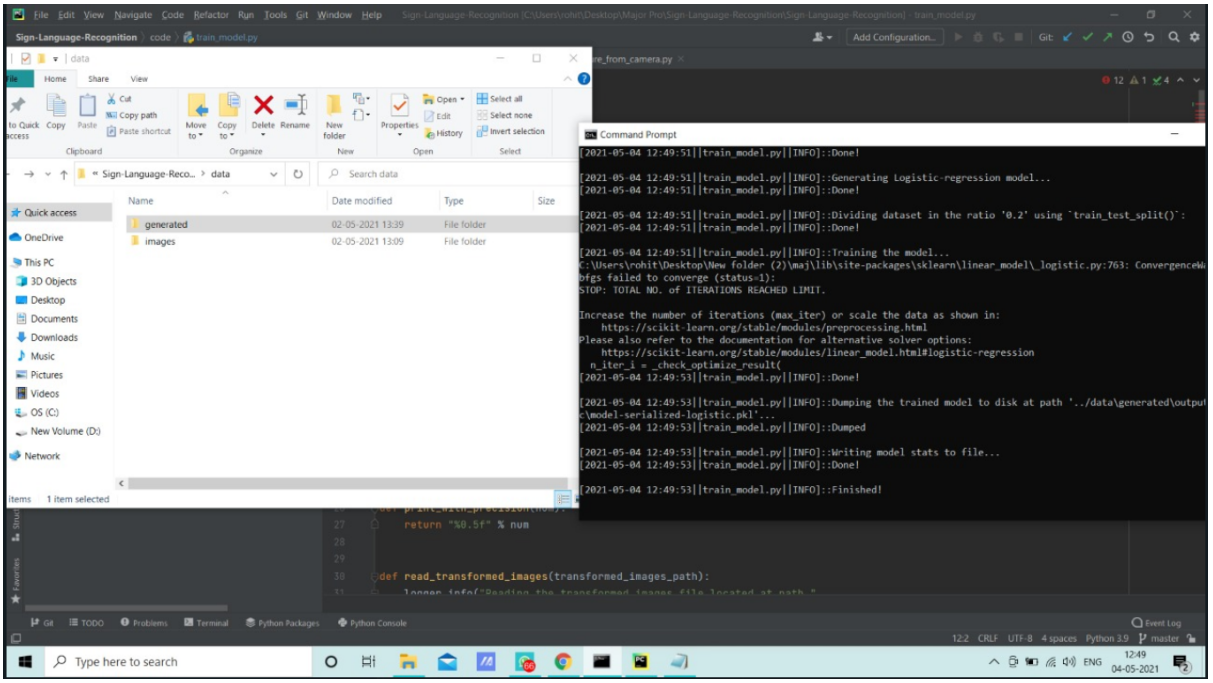

Figure 4 : Training Data (LR)

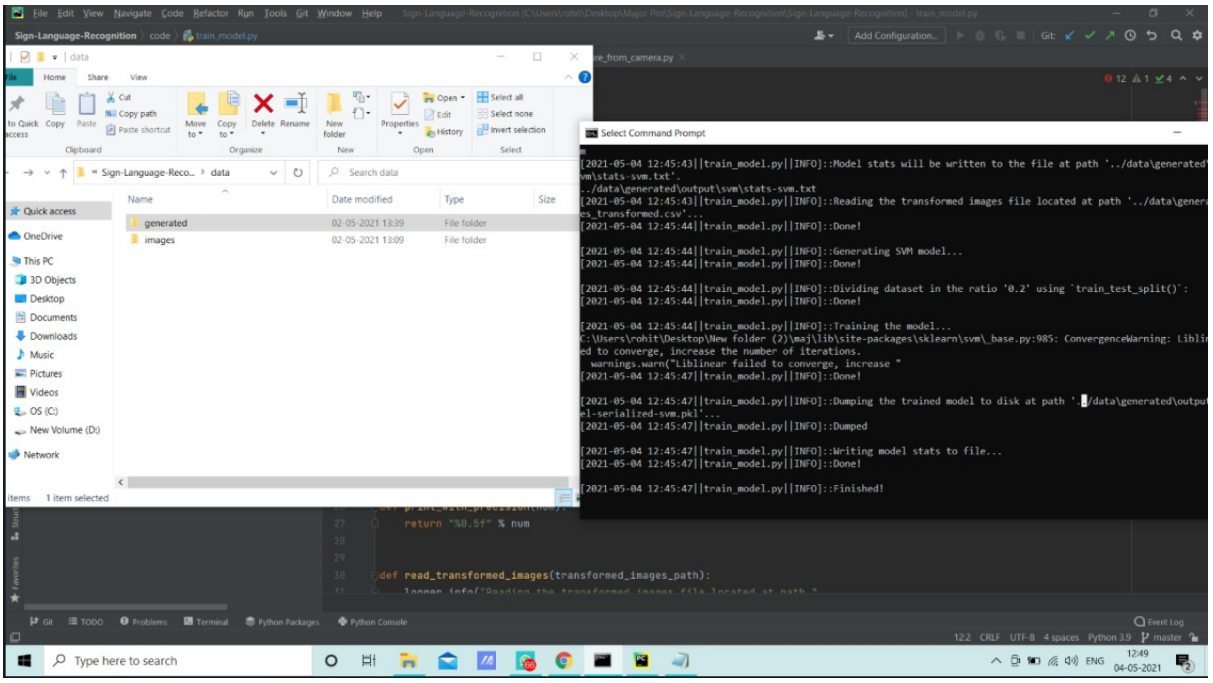

Figure 5 : Training Data (SVM)

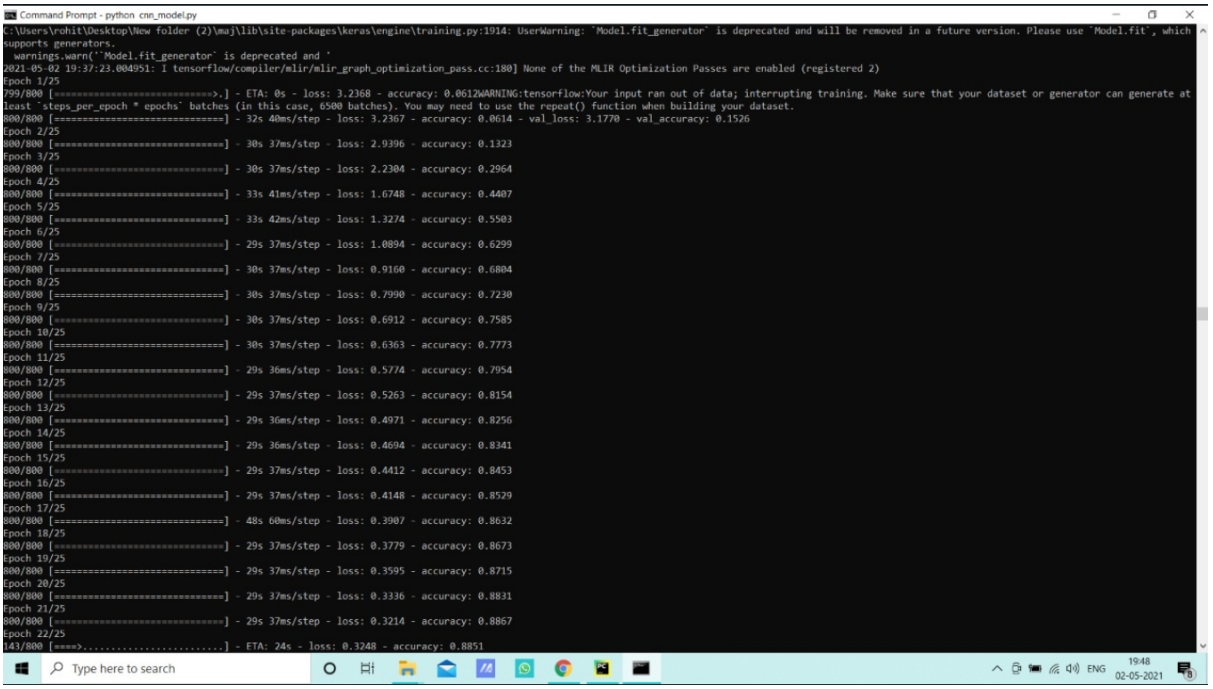

Figure 6 : Training Data(CNN)

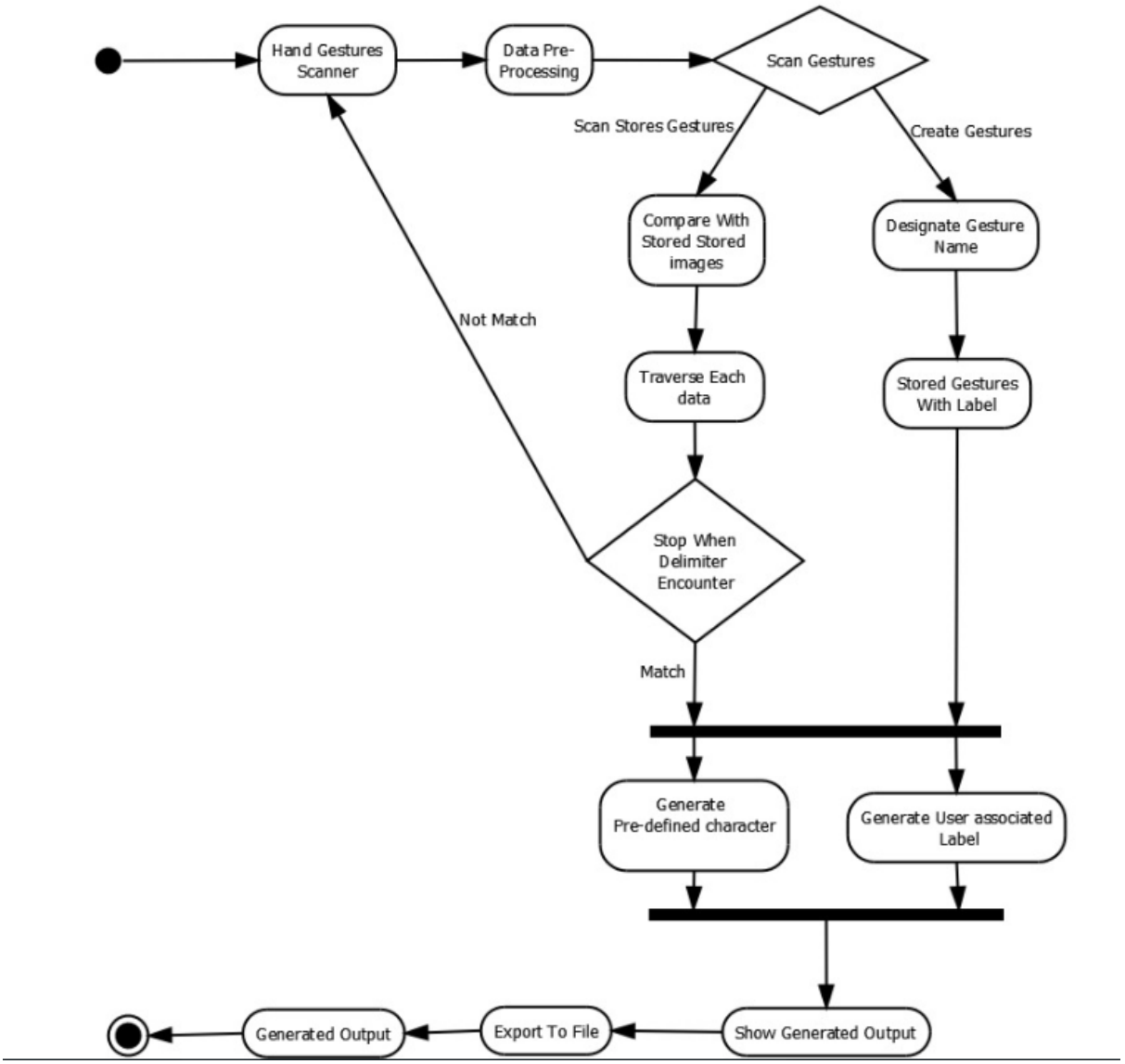

Figure 7 : Control flow

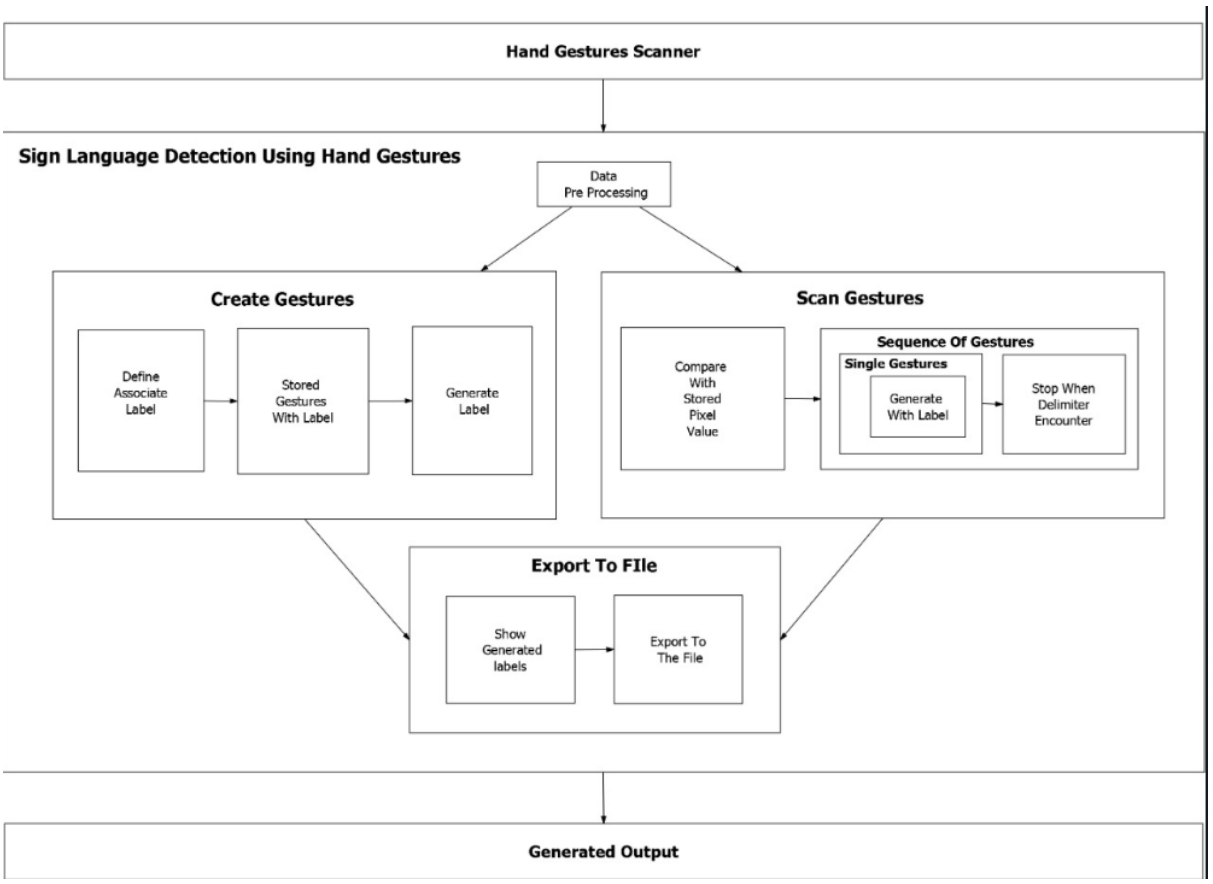

Figure 8 : Sequence Diagram

4.2 Implementation Details

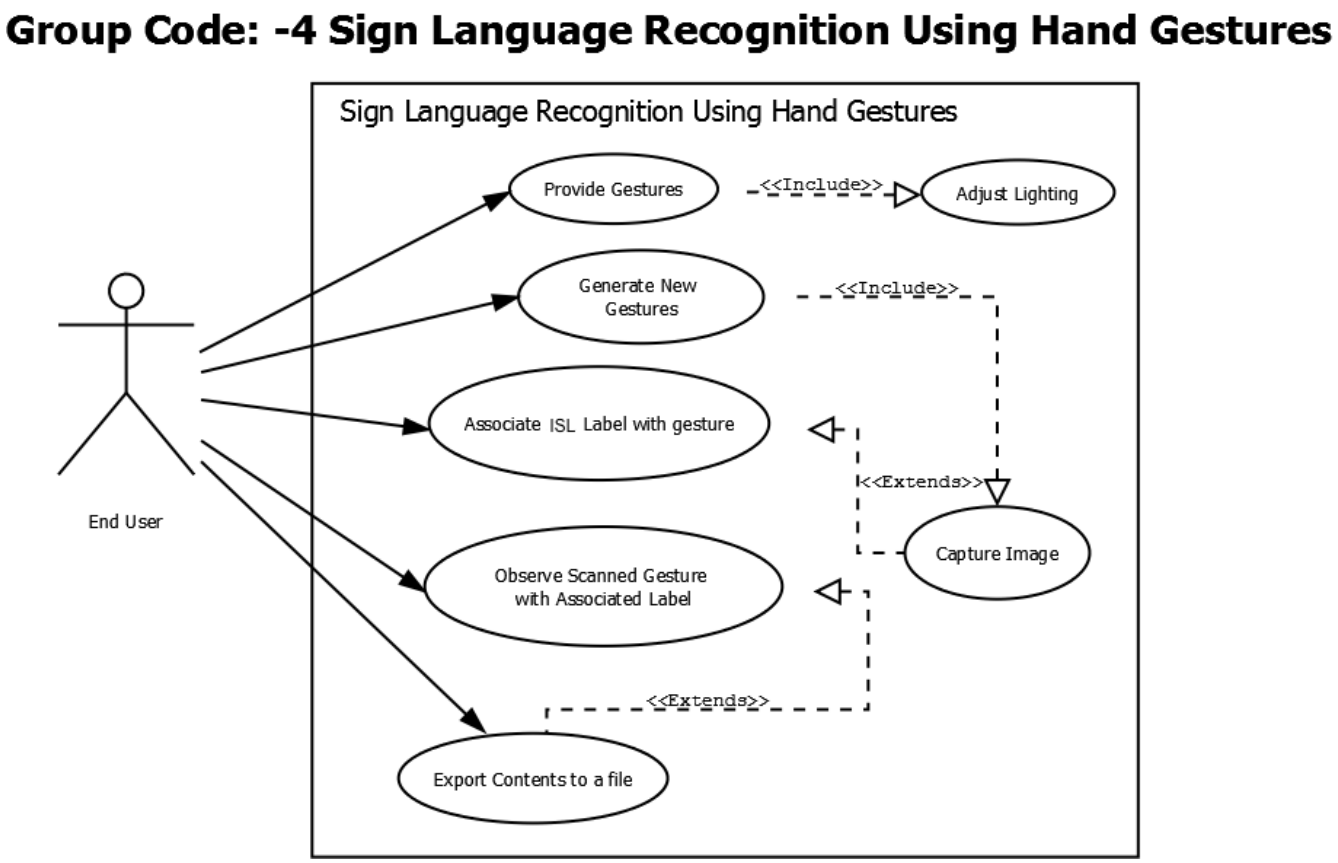

Figure 9 : Implementation Diagram

Chapter 5 : TESTING

5.1 Testing Plan

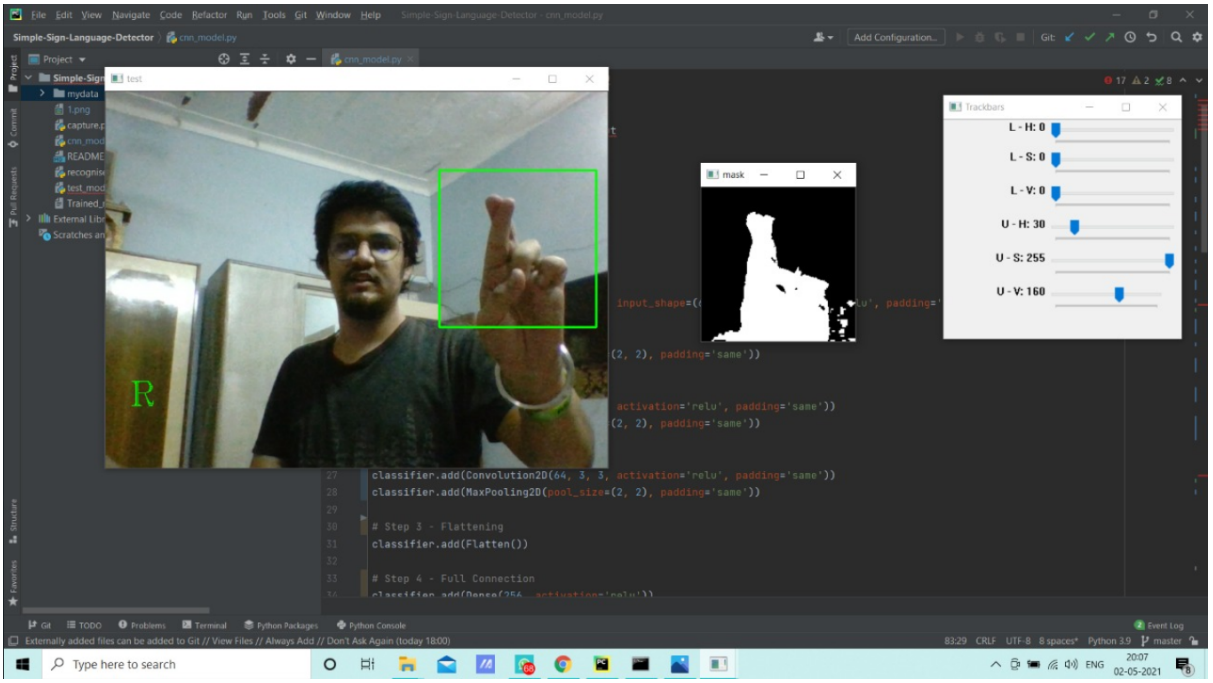

Figure 10 : Testing-SYMBOL R

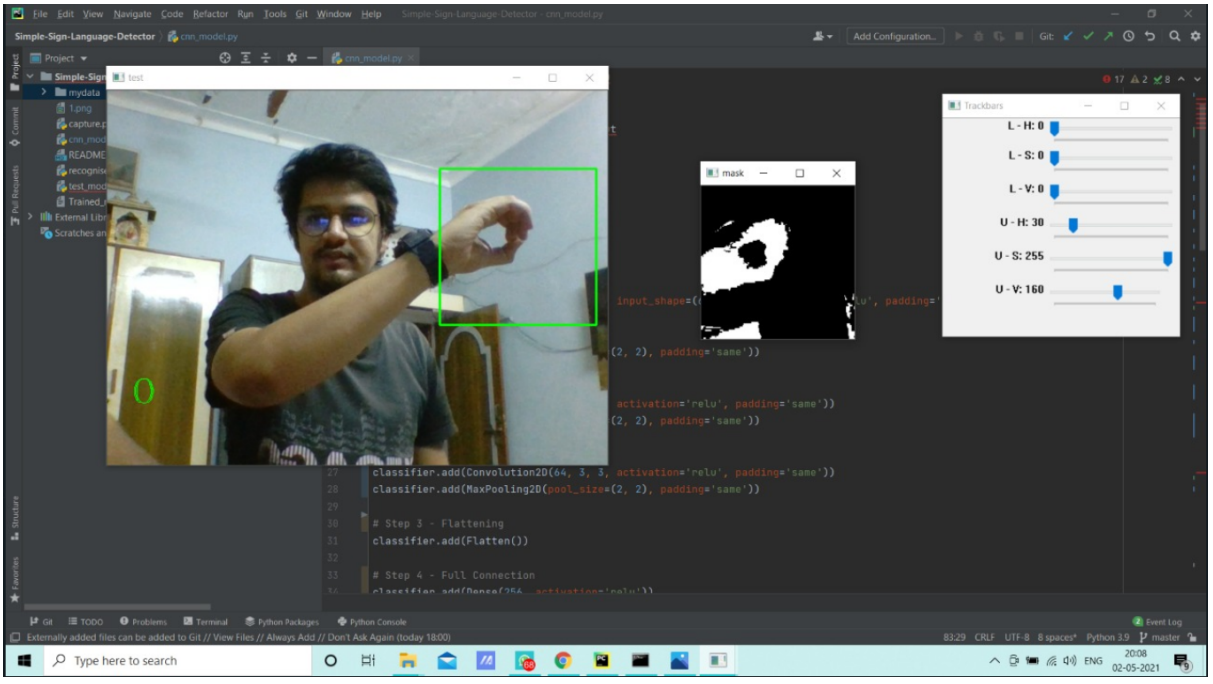

Figure 11 : Testing-SYMBOL O

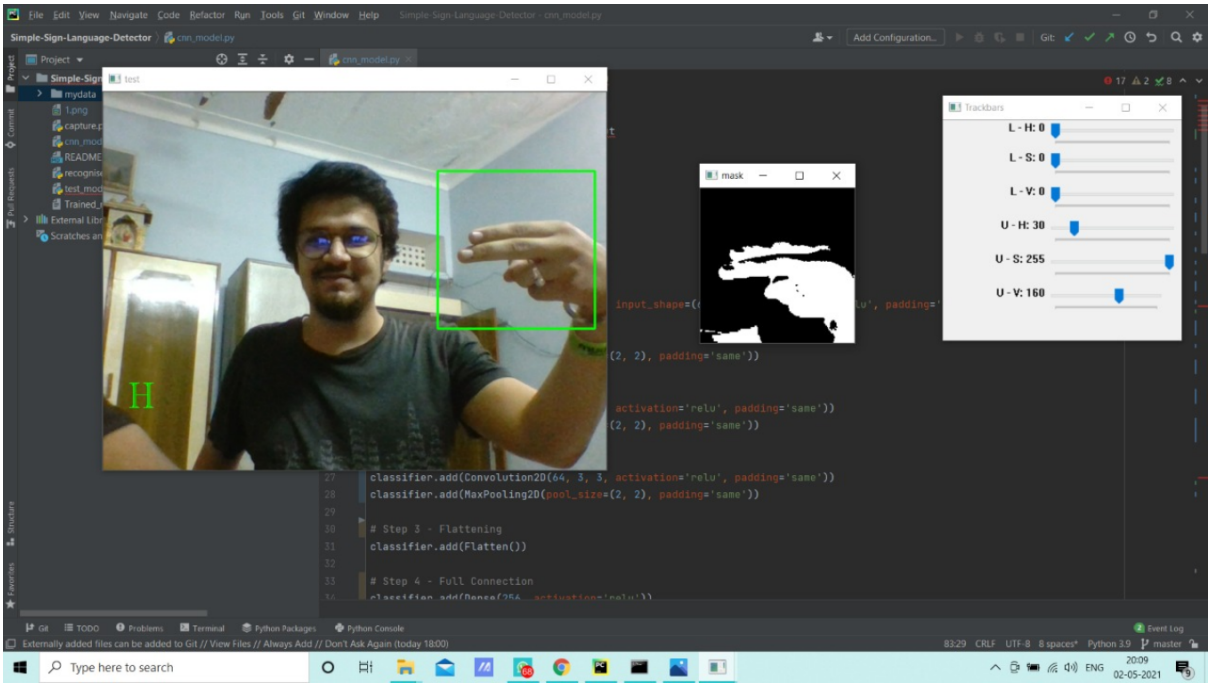

Figure 12 : Testing-SYMBOL H

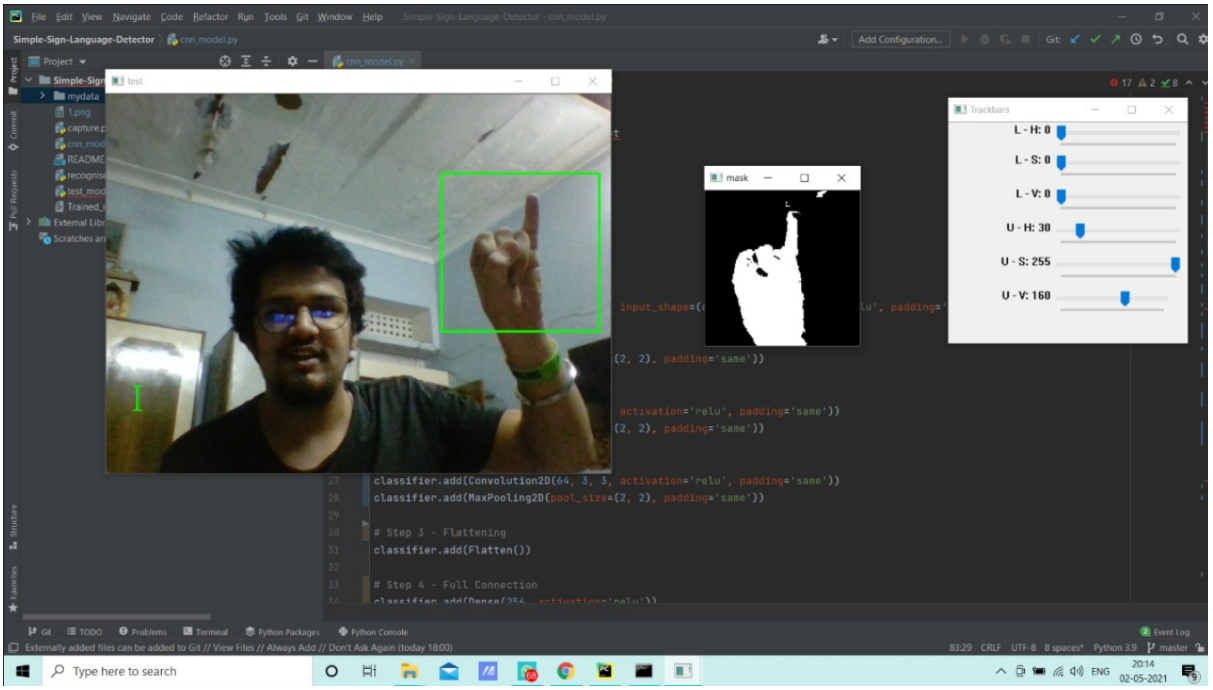

Figure-13 : Testing- SYMBOL I

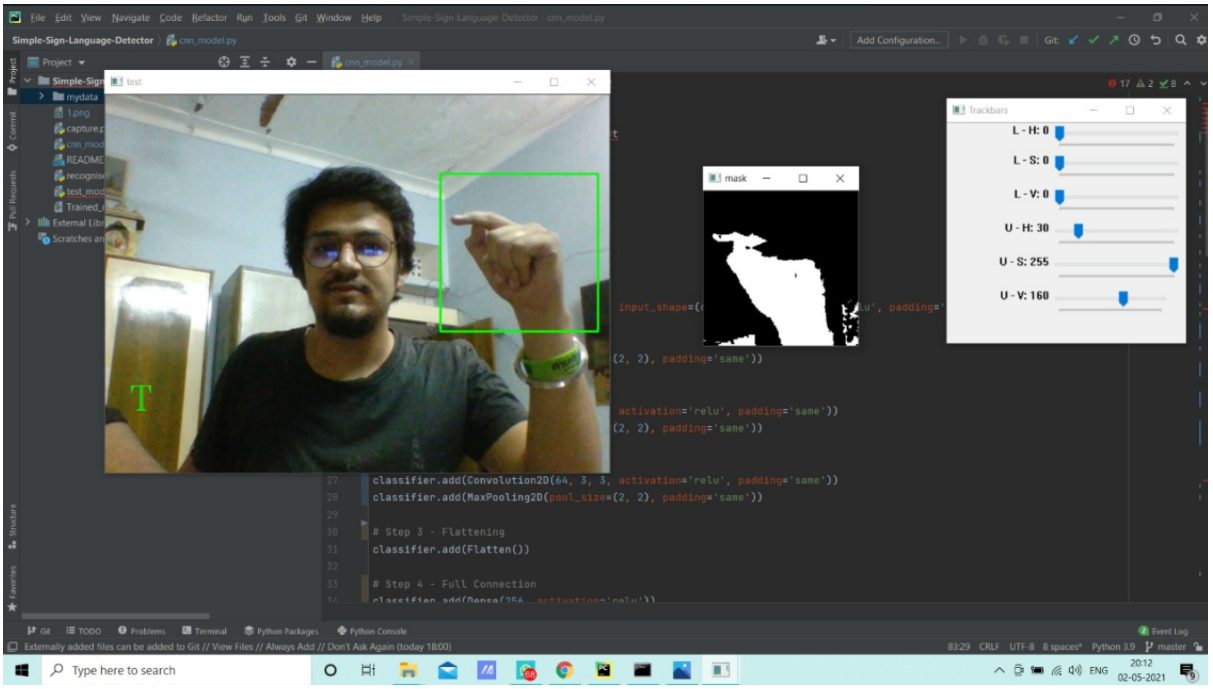

Figure-14 : Testing- SYMBOL T

5.2 Component decomposition and type of testing required

The objectives behind the testing of our developed models are:

- Evaluation of Parameters of the developed system
- Calculating accuracy
- Comparing models with each other
- User Level Testing

5.3 Limitations

A better dataset can be created so that the training of the algorithm can be done on a more accurate dataset rather than pretrained weights. This will increase the accuracy of results obtained.

And Manual debugging is needed for a few algorithms.

Chapter 6 : FINDINGS, CONCLUSION AND FUTURE WORK

6.1 Findings

We trained a model majorly in CNN which we will be shown in our project.
We also used KNN which gave us an accuracy of 89.2%. After that we used CNN and it gave us an accuracy of 90.82%. SVM gave us 94% accuracy and LR gave us an accuracy of 92.1%.
SVM is the best out of all because it takes less time to train and more accuracy.

6.2 Conclusion

This work is totally a good approach for getting the hand gestures to your screen very quick. This works on the bunch of low level images of motion and colour too. We already transformed our photos to change the colour map from skin colour and background colour to Black and White only through HSV transformation.So basically manpower will be easily replaced by this project which gives a better accuracy in SVM model and takes less time too, although it's just a Computer vision based detecting sign language.

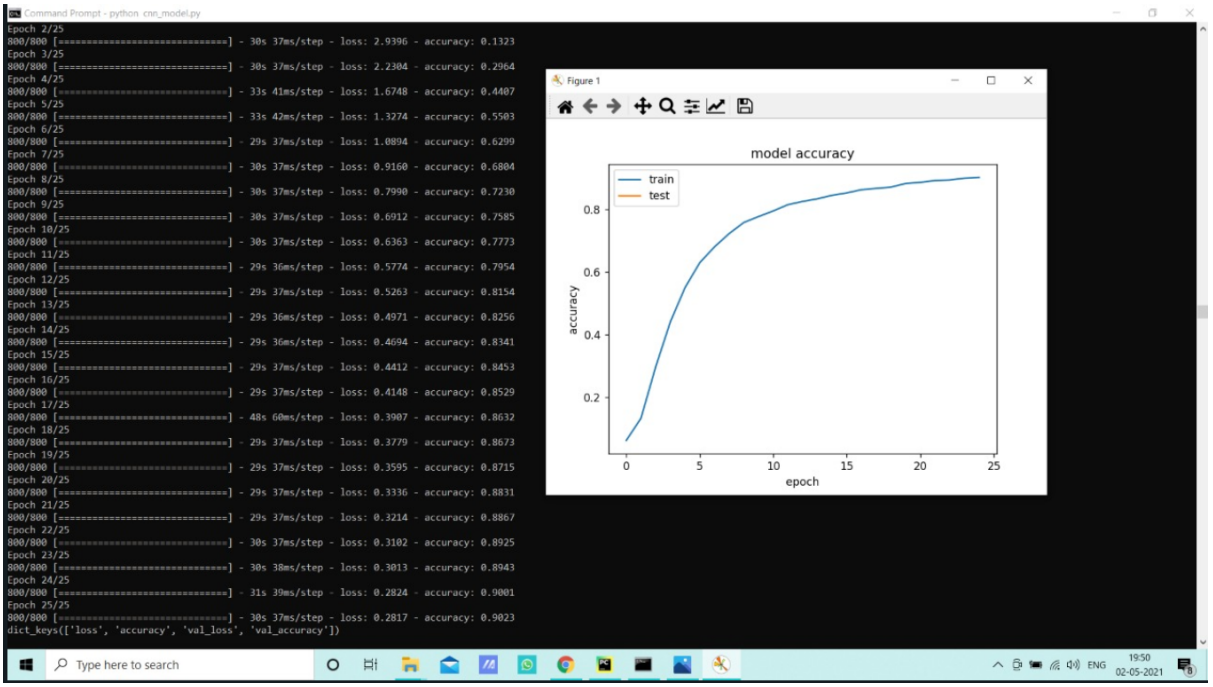

Figure 15 : Result 1(Accuracy)

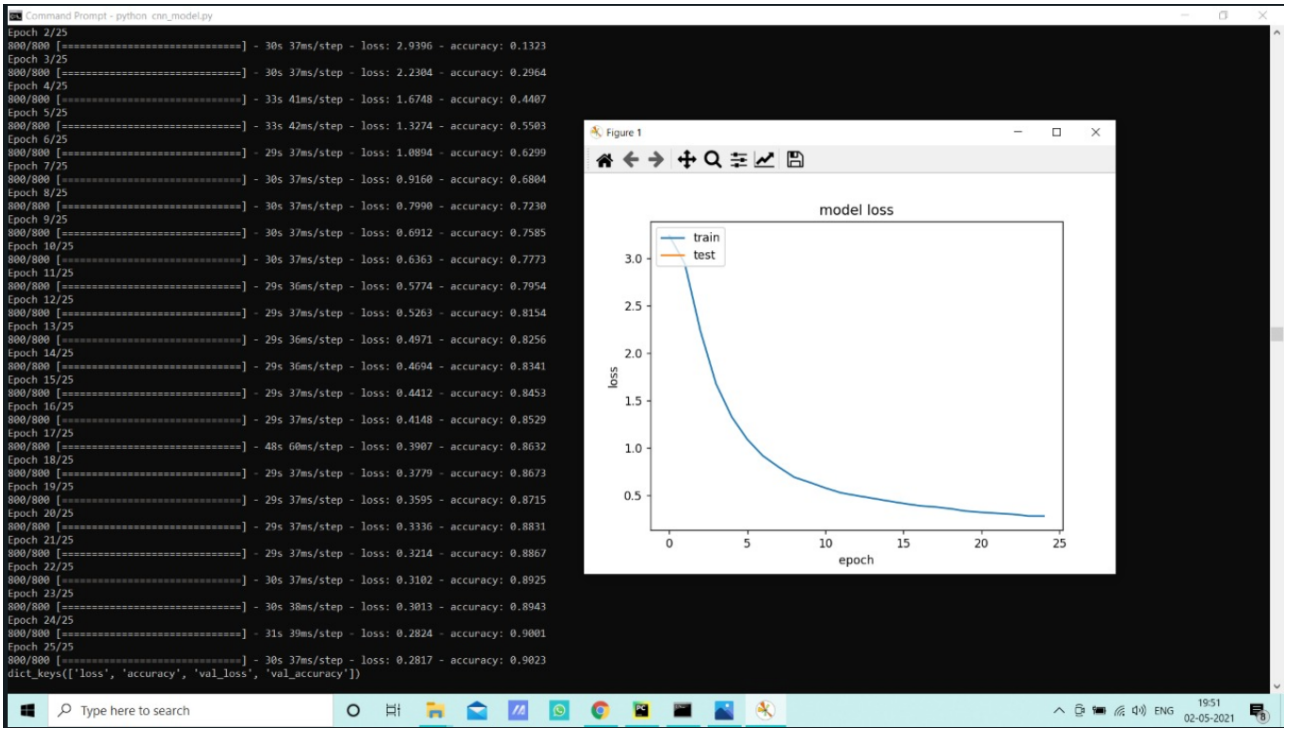

Figure 16 : Result 2(Loss)

```
Model used = 'knn'Classifier model details:
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=None, n_neighbors=10, p=2,
           weights='uniform')

Model score:
0.892

Classification report:
              precision    recall  f1-score   support

           A       0.91      0.86      0.89        43
           B       0.86      0.91      0.89        41
           C       0.91      0.91      0.91        55
           D       0.87      0.91      0.89        24
           E       0.87      0.89      0.88        55
           F       0.88      0.89      0.88        33
           G       0.91      0.86      0.88        59
           H       0.77      0.81      0.89        20
           I       0.91      0.91      0.91        28
           K       0.91      0.91      0.91        63
           L       0.91      0.87      0.89        23
           M       0.89      0.91      0.90        48
           N       0.87      0.87      0.85        46
           O       0.86      0.89      0.87        41
           P       0.88      0.91      0.89        55
           Q       0.91      0.91      0.91        18
           R       0.91      0.91      0.91        27
           S       0.91      0.86      0.88        38
           T       0.91      0.91      0.91        35
           U       0.91      0.91      0.91        21
           V       0.91      0.91      0.91        18
           W       0.91      0.80      0.85        19
           X       0.91      0.91      0.91        42
           Y       0.91      0.89      0.90        45

   micro avg       0.89      0.89      0.90       897
   macro avg       0.89      0.90      0.89       897
weighted avg       0.89      0.89      0.89       897
```

Figure 17 : KNN accuracy

```
Model used = 'logistic'Classifier model details:
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='warn',
          n_jobs=None, penalty='l2', random_state=None, solver='warn',
          tol=0.0001, verbose=0, warm_start=False)

Model score:
0.92142

Classification report:
              precision    recall  f1-score   support

           A       0.93      0.91      0.93        43
           B       0.91      0.91      0.92        41
           C       0.93      0.93      0.93        55
           D       0.93      0.93      0.93        24
           E       0.89      0.93      0.91        55
           F       0.84      0.90      0.87        33
           G       0.93      0.90      0.91        59
           H       0.80      0.93      0.86        20
           I       0.93      0.93      0.93        28
           K       0.93      0.93      0.93        63
           L       0.93      0.84      0.88        23
           M       0.93      0.91      0.92        48
           N       0.91      0.89      0.90        46
           O       0.93      0.93      0.93        41
           P       0.93      0.91      0.92        55
           Q       0.93      0.93      0.93        18
           R       0.93      0.93      0.93        27
           S       0.93      0.93      0.93        38
           T       0.90      0.90      0.90        35
           U       0.93      0.93      0.93        21
           V       0.93      0.93      0.93        18
           W       0.88      0.88      0.88        19
           X       0.91      0.93      0.93        42
           Y       0.93      0.93      0.93        45

   micro avg       0.92      0.92      0.93       897
   macro avg       0.92      0.92      0.92       897
weighted avg       0.92      0.92      0.92       897
```

Figure 18 : LR accuracy

```
Model used = 'svm'Classifier model details:
LinearSVC(C=1.0, class_weight=None, dual=True, fit_intercept=True,
     intercept_scaling=1, loss='squared_hinge', max_iter=1000,
     multi_class='ovr', penalty='l2', random_state=None, tol=0.0001,
     verbose=0)

Model score:
0.94125

Classification report:
              precision    recall  f1-score   support

           A       0.96      0.91      0.94        43
           B       0.91      0.94      0.92        41
           C       0.96      0.96      0.96        55
           D       0.96      0.96      0.96        24
           E       0.92      0.96      0.94        55
           F       0.87      0.93      0.90        33
           G       0.96      0.93      0.94        59
           H       0.83      0.96      0.89        20
           I       0.96      0.96      0.96        28
           K       0.96      0.96      0.96        63
           L       0.96      0.87      0.91        23
           M       0.96      0.94      0.95        48
           N       0.92      0.92      0.92        46
           O       0.96      0.96      0.96        41
           P       0.96      0.94      0.95        55
           Q       0.96      0.96      0.96        18
           R       0.96      0.96      0.96        27
           S       0.91      0.96      0.93        38
           T       0.96      0.93      0.95        35
           U       0.96      0.96      0.96        21
           V       0.96      0.90      0.93        18
           W       0.91      0.91      0.91        19
           X       0.94      0.96      0.95        42
           Y       0.96      0.94      0.95        45

   micro avg       0.94      0.94      0.94       897
   macro avg       0.94      0.94      0.94       897
weighted avg       0.94      0.94      0.94       897
```

Figure 19 : SVM accuracy

6.3 Future Work

As we have now 4 models trained based on these classifiers:

- KNN
- CNN
- SVM
- LR

We have one testing model which is the CNN model. But as per the final accuracy values , the best model to implement in this world to help deaf people is SVM. The major future work is to use this trained model(SVM) for testing and we know that we only have alphabets now to recognize using hand signs. There are gestures or moving gestures for a whole word or for a few sentences also, like Thank you, Come here, etc. So we will implement these moving gestures models to be trained on the best of these classifiers because OpenCv is a very widely used library and can be used for a wide range of gesture detection.

Background subtraction is vital in several computer vision applications. we have a tendency to use it to count the quantity of cars passing through a toll booth. we have a tendency to use it to count the quantity of individuals walking in and out of a store.

And we can use it for motion detection.

REFERENCES

1. Vivek Bheda and Dianna Radpour. “Using Deep Convolutional Networks for Gesture Recognition in American Sign Language”. In: CoRR abs/1710.06836 (2019). arXiv: 1710.06836.

2. S. Sarkar. Segmentation-robust representations, matching, and modeling for sign language recognition, CVPR Workshop on Gesture Recognition.

3. Mukesh Kumar Makwana, " Sign Language Recognition", M.Tech thesis, Indian Institute of Science, Bangalore.

4. Kang, Byeongkeun , Subarna Tripathi, and Truong Q. Nguyen. "Real-time sign language fingerspelling recognition using convolutional neural networks from depth map." Pattern Recognition (ACPR)

5. Pigou, Lionel, et al. ”Sign language recognition using convolutional neural networks.” Workshop at the European Conference on Computer Vision. Springer International Publishing.

6. Object Detection with Deep Learning Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu