Dr. D. Y. Patil Unitech Society's
DR. D. Y. Patil Institute of Technology, Pimpri, Pune.

Department of Electronics and Telecommunication Engineering

Sign Language Recognition & Multilingual voice conversion with Teachable Machine for Differently Abled Person

Group Members:

Vaishnavi Kolhe BETA57

Pranay Patil BET54

Kunal Giradkar BETA52

Under Guidance of : Mrs. Vandana Katarwar

Problem statement:

Deaf is the inability to hear and dumb is the inability to speak. They communicate using sign language among themselves and with normal people but normal people do not take seriously the importance of sign language. Dumb people use hand signs to communicate, hence normal people face problems in recognizing their language by signs made. Hence there is a need for systems which recognize the different signs and convey the information to the normal people.

Introduction

- Sign languages are vivid on a wide and world level. There are multiple sign languages in the world which are regularly in use that are ASL (American Sign Language) ISL (Indian Sign Language), BSL (Bangladesh Sign Language), MSL (Malaysian Sign Language). etc. These languages are Built and Developed with lots of work and practical testing with intention of feasibility to the deaf and dumb persons. Any language is created with its word and its meaning. Sign Language is created as “Sign” and “Action of That Sign”. Because here we are not able to make them understand the meaning of the sign by writing words. As they are deaf and cannot listen from birth so we cannot teach them words.

Aim and Objectives

Aim of the Project :

We are motivated with the aim to use new technologies for better humanity. We found Machine learning like technologies can be used for conquering the backwardness that occurred because of this physical disability.

Objectives :

1. To contribute to the social welfare domain with the existing technology to make it more reliable and efficient.
2. To study the American & Indian Sign Language in detail and provide image and pattern recognition solutions for identification and grading.

Block Diagram/Methodology

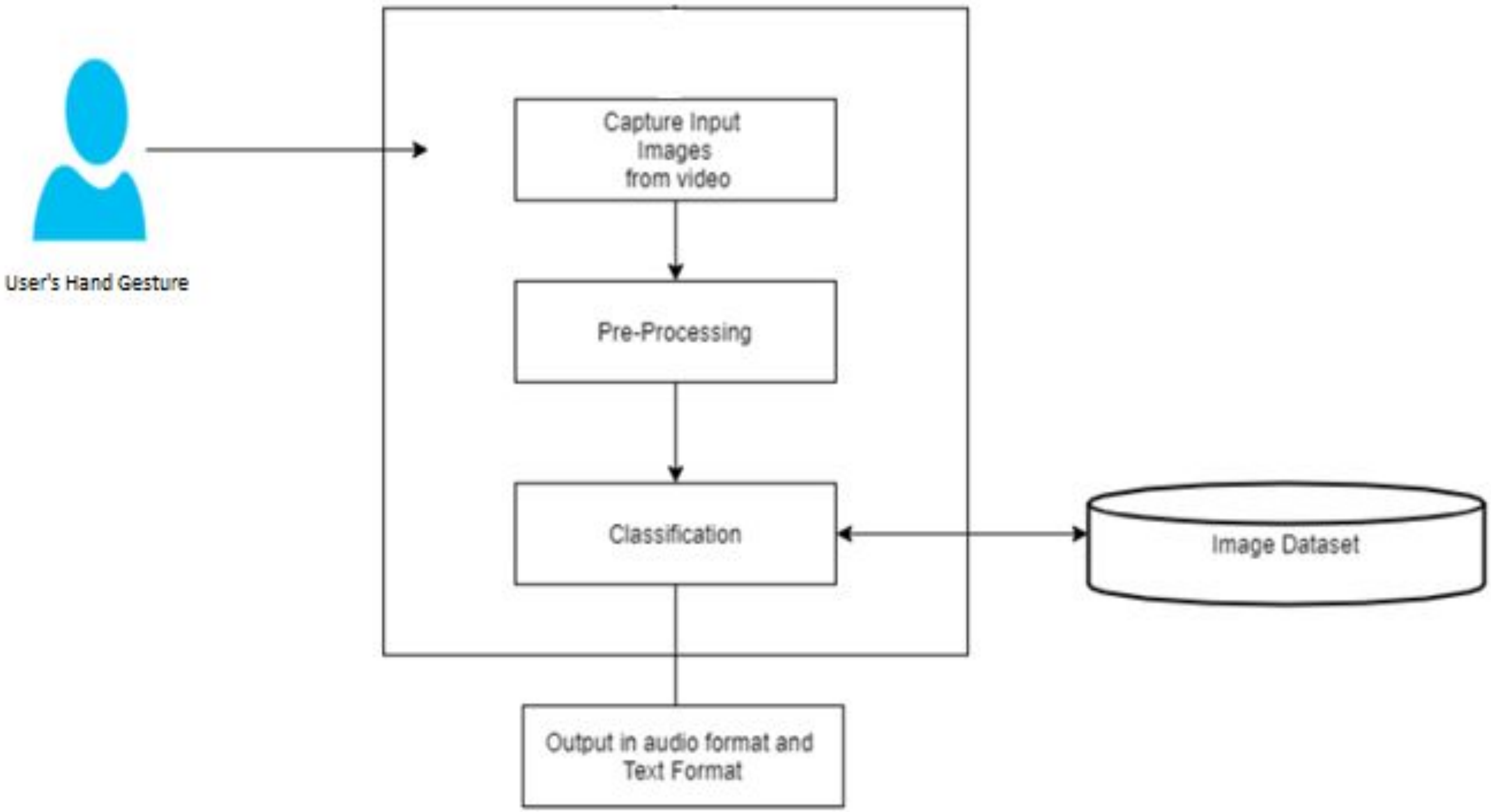
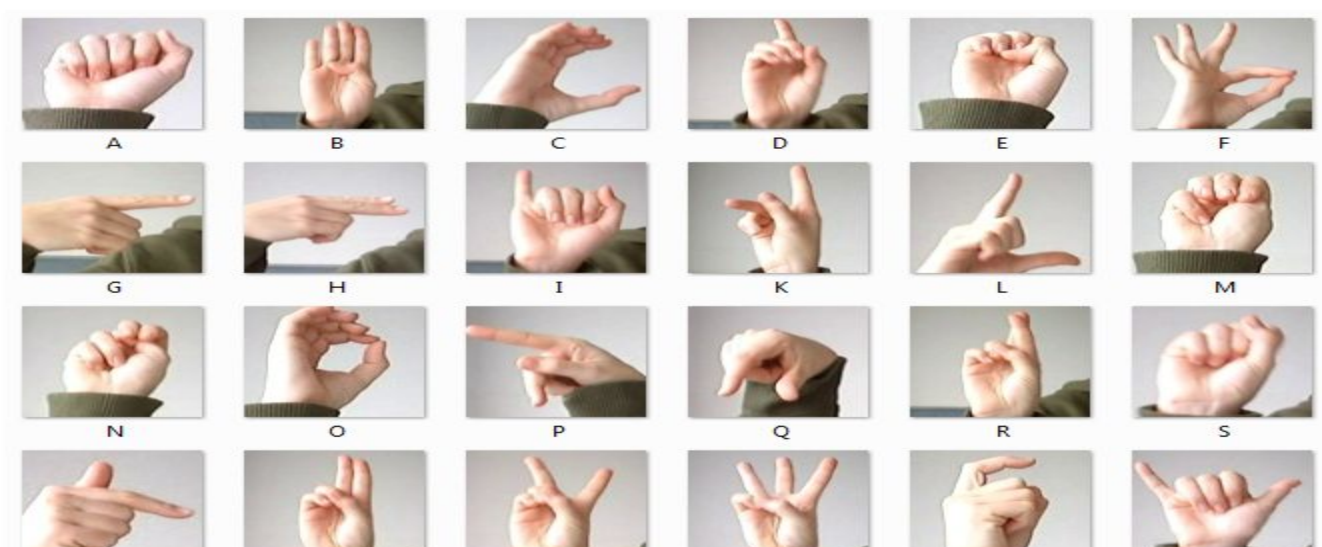


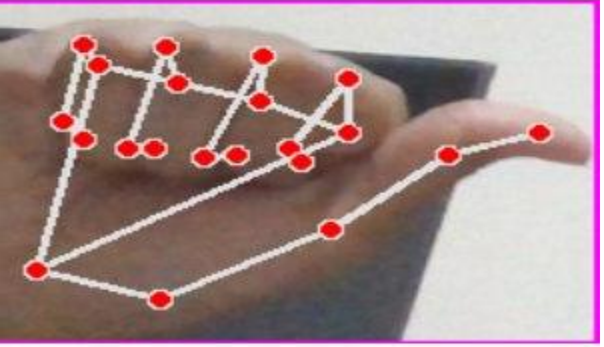
Fig. System Architecture

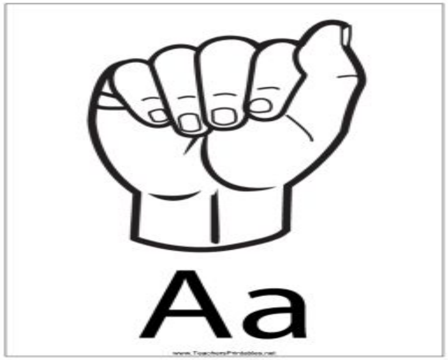
Data Collection

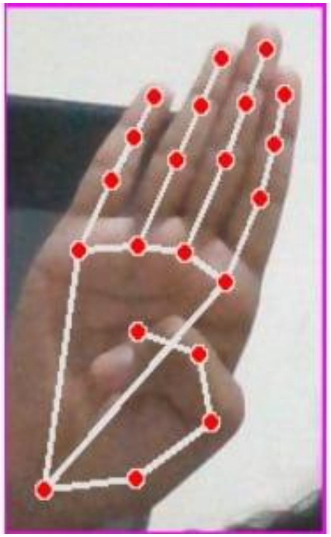
The first step is to collect a large dataset of sign language images. This dataset should include a wide variety of sign language gestures from different sign languages and performed by different people.

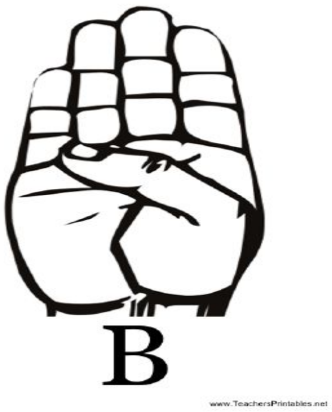
- With the help of Google Media Pipe we created our own data set consisting of 500 images per sign through different angles and lighting with different background and etc.

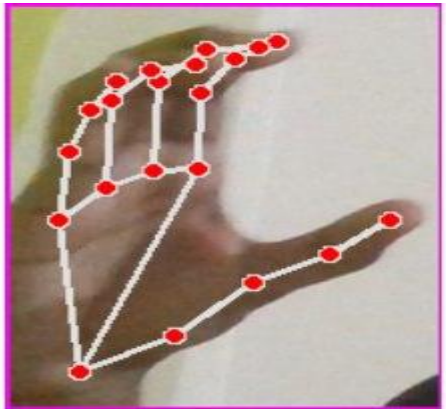


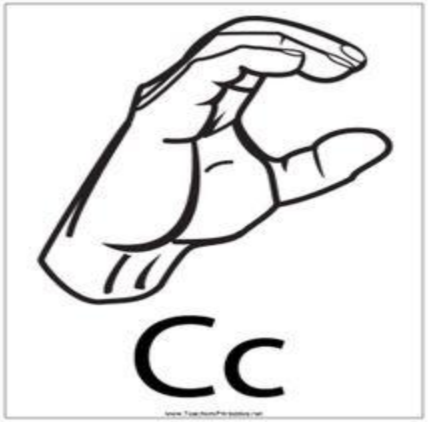


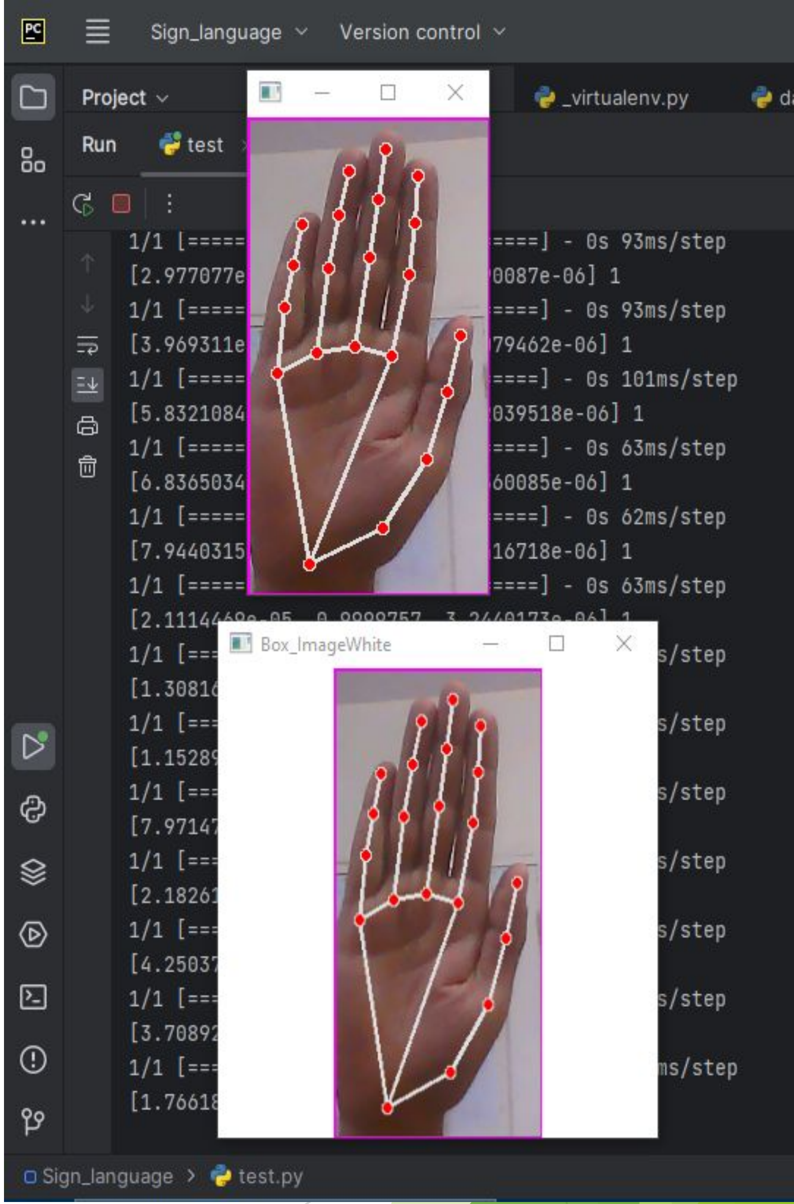








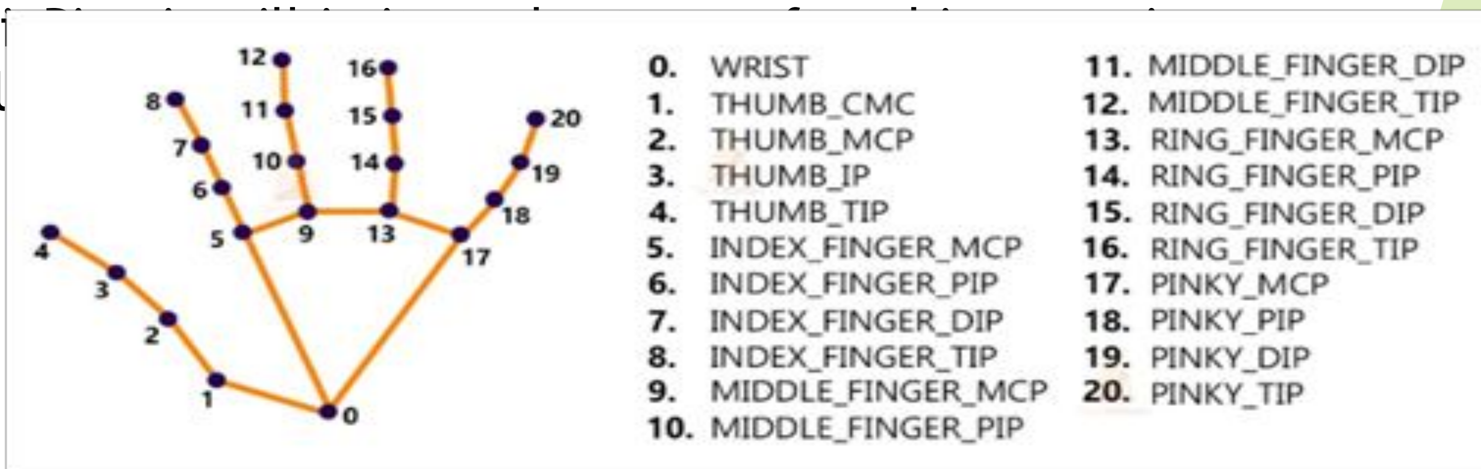




Photos of data set collections

Google's MediaPipe Hand Tracking

MediaPipe is the most straightforward method for developers and academics to create machine learning (ML) apps across the web, cloud, edge, and mobile platforms. It is a node-based framework designed for scholars, students, and professionals that want code release research, engineering, and production-ready machine learning applications. Multipurpose (video, audio, and sensor) pipelines to conduct machine learning may be built using it. This implies that learning sign language has other benefits beyond its use in communication, such as its application in interactions between people and machines. Unfortunately, Medi
publ



Training of Dataset

Once the dataset has been generated, and all the necessary images have been collected, the next step involves developing a machine learning model capable of predicting hand signs and converting them into a file format compatible with easy integration into a mobile application. There are various methods to generate this model. One option is to create custom Python scripts utilizing different machine learning models. Alternatively, OpenCV can be employed for model development.

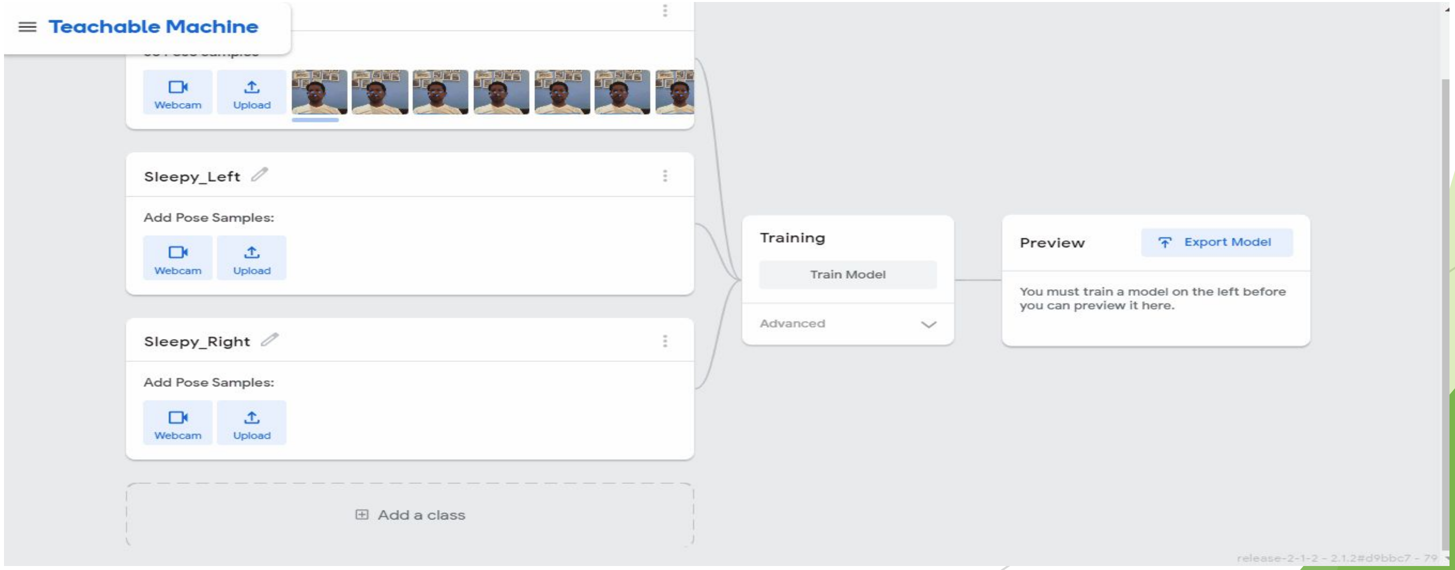
In our case, we utilized Teachable Machine, a web-based application that facilitates the creation of effective machine learning models. The Teachable Machine platform offers three options for model creation: picture classification mode, speech classification, and pose classification. For our sign language recognition system, we employed the picture classification mode. We categorized the images corresponding to different sign language gestures into distinct classes and used Teachable Machine to construct the machine learning model.

Additionally, Teachable Machine allows for the incorporation of additional accuracy adjustments before the model generation process begins. This flexibility ensures the refinement of the machine learning model to enhance its performance.

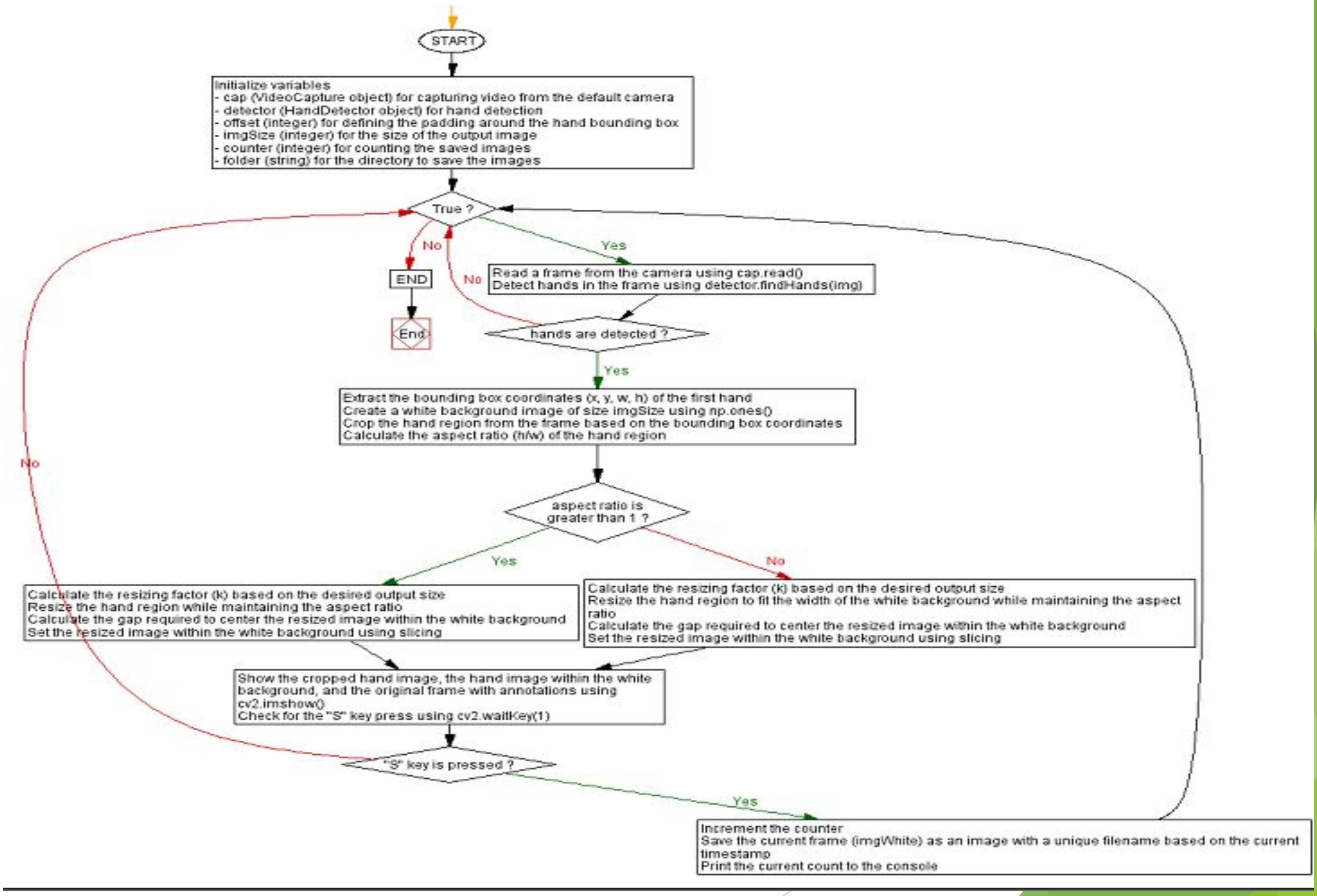
Teachable Machine

- Teachable Machine uses a simple drag-and-drop interface.
- You start by collecting images or sounds for your two target classes.
- For example, you could collect images of cats and dogs.
- Once you have your data, you label each image or sound with its
- corresponding class.
- Finally, you train the model on your labeled data. This involves the model l
- earning to identify the patterns that differentiate your classes.
- Finally, you train the model on your labeled data. This involves the mode
- learning to identify the patterns that differentiate your classes

Teachable Machine is a great tool for beginners because it removes the complexity of coding. The intuitive interface allows you to focus on collecting data and training your model.



Data Collection Algorithm:



Testing Algorithm:

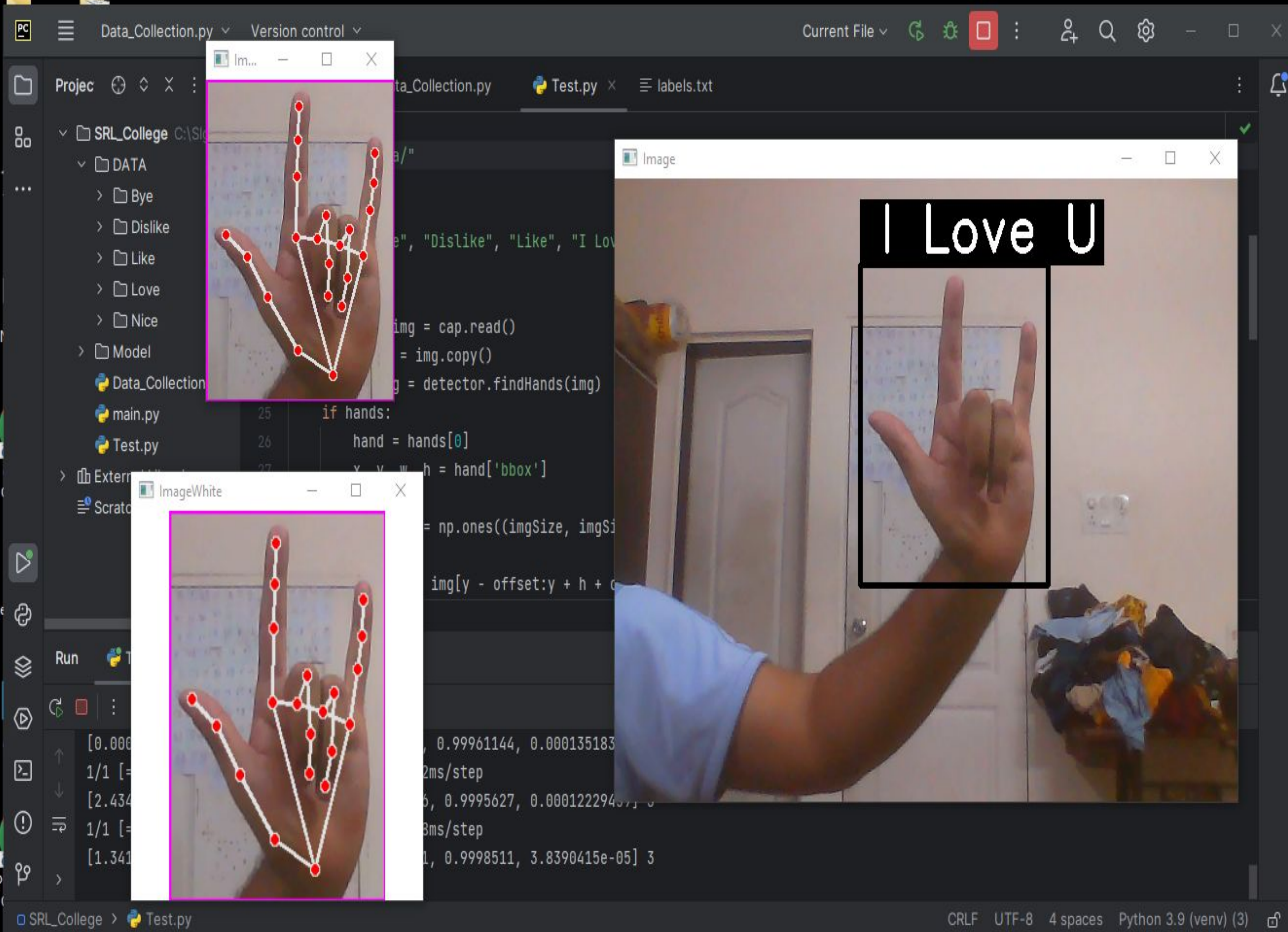
```
START

Initialize variables
- cap (VideoCapture object) for capturing video from the default camera
- detector (HandDetector object) for hand detection
- classifier (Classifier object) for hand gesture classification
- offset (integer) for defining the padding around the hand bounding box
- imgSize (integer) for the size of the output image
- counter (integer) for counting
- folder (string) for the directory to save data
- labels (list) for gesture labels
WHILE True:
Read frame from the camera using cap.read()
Create a copy of the frame as imgOutput
Detect hands in the frame using detector.findHands(img)
IF hands are detected:
Get the first hand detected as hand
Extract the bounding box coordinates (x, y, w, h) of the hand

Create a white background image of size imgSize
Crop the hand region from the frame based on the bounding box coordinates
Calculate the aspect ratio of the hand region
IF aspect ratio is greater than 1:
Calculate the resizing factor (k) based on the desired output size
Resize the hand region while maintaining the aspect ratio
Calculate the gap required to center the resized image within the white background
Set the resized image within the white background using slicing
ELSE:
Calculate the resizing factor (k) based on the desired output size
Resize the hand region to fit the width of the white background while maintaining the aspect
ratio
Calculate the gap required to center the resized image within the white background
Set the resized image within the white background using slicing
Use the classifier to get the prediction and index of the gesture from the resized hand image
Draw a filled rectangle above the hand region with the predicted gesture label
Draw a rectangle around the hand region
Display the cropped hand image, the hand image within the white background, and the original
frame with annotations
Check for any key press (waitKey(1))
END

End
```

Results

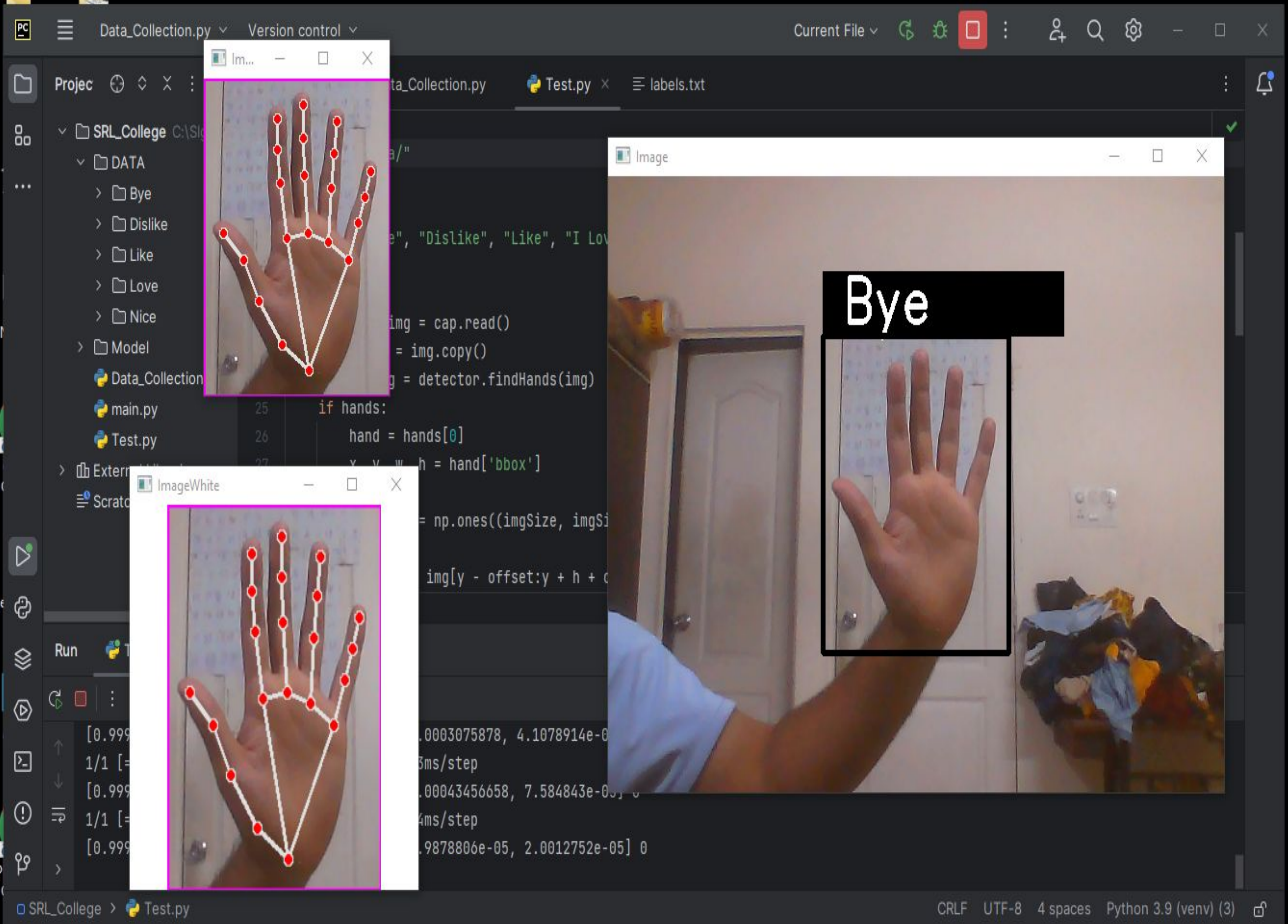
- We highlight the model's obtained accuracy and other pertinent metrics as we provide the results of its performance on an independent test set. We examine performance variances over a range of hand motions and difficult situations, such as various illumination conditions or hand orientations. We also highlight the benefits and advancements made by our proposed model by contrasting its performance with baseline methods or other hand motion detection algorithms.

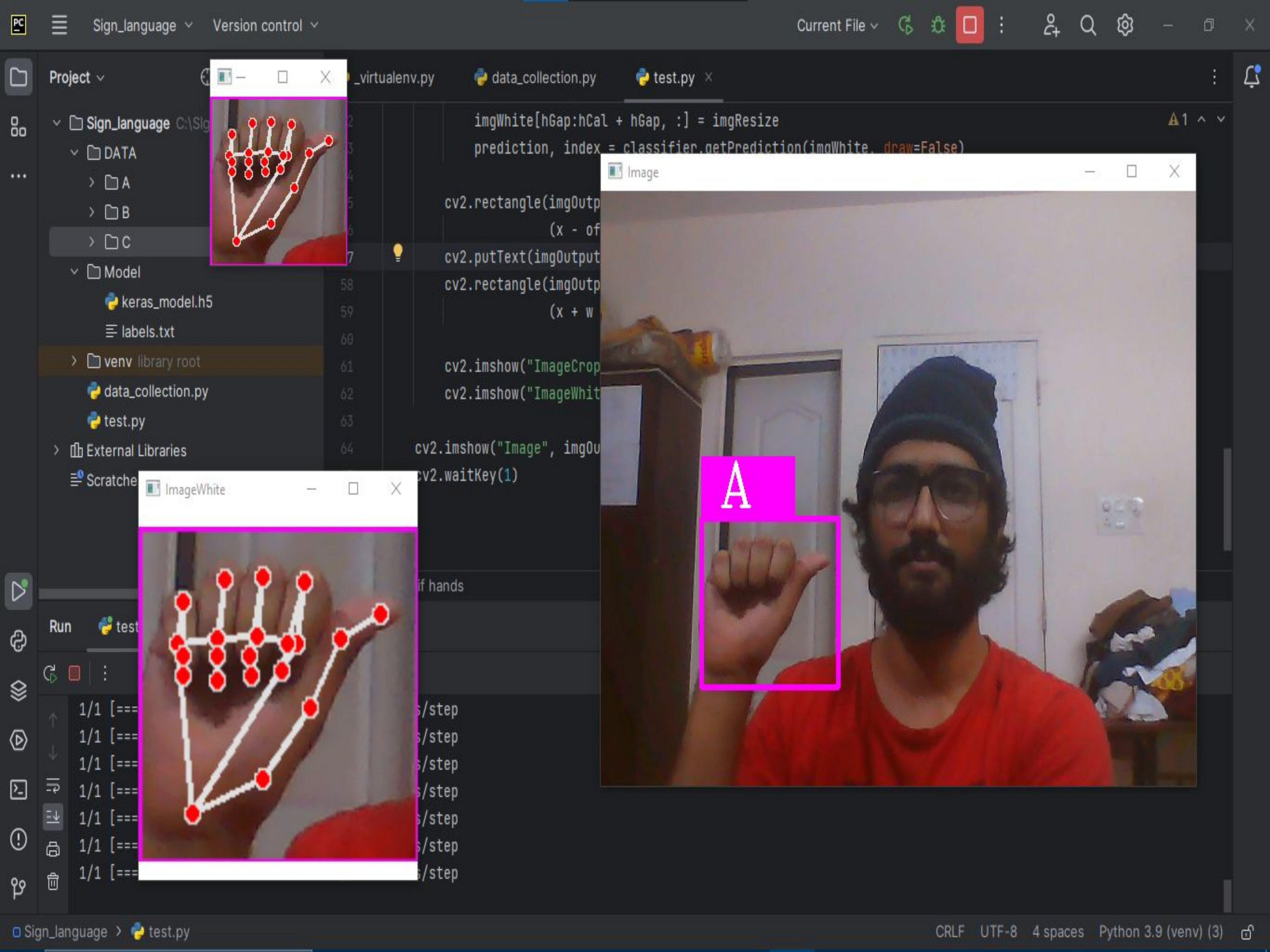
- The created system can instantly recognize American Sign Language alphabets. Google Media Pipe has been used in the system's development. The Google Teachable Machine was used to train the model. On the newly constructed dataset, which consists of 4500 photos overall and 300 images for each alphabet, it has been trained using transfer learning.

- The system's output is based on the confidence level, and that level is currently 85.45% on average. By expanding the dataset, the system's confidence level can be raised, improving the system's capacity for recognition. As a result, the system's performance is enhanced.



Data_Collection.py
Version control
Current File
Projec
SRL_College
DATA
Bye
Dislike
Like
Love
Nice
Model
Data_Collection
main.py
Test.py
Im...
ta_Collection.py
Test.py
labels.txt
Image
I Love U
e", "Dislike", "Like", "I Lov
img = cap.read()
= img.copy()
g = detector.findHands(img)
if hands:
hand = hands[0]
h = hand['bbox']
= np.ones((imgSize, imgSi
img[y - offset:y + h +
ImageWhite
Run
, 0.99961144, 0.000135183
2ms/step
6, 0.9995627, 0.00012229
3ms/step
1, 0.9998511, 3.8390415e-05] 3
SRL_College
Test.py
CRLF
UTF-8
4 spaces
Python 3.9 (venv) (3)


Data_Collection.py
Version control
Current File
Project
SRL_College
DATA
Bye
Dislike
Like
Love
Nice
Model
Data_Collection
main.py
Test.py
Test.py
labels.txt
"Dislike", "Like", "I Lov
img = cap.read()
= img.copy()
= detector.findHands(img)
if hands:
hand = hands[0]
h = hand['bbox']
= np.ones((imgSize, imgSi
img[y - offset:y + h +
Image
Bye
ImageWhite
Run
.0003075878, 4.1078914e-0
1/1 [=
3ms/step
.00043456658, 7.584843e-05
4ms/step
.9878806e-05, 2.0012752e-05] 0
SRL_College
Test.py
CRLF
UTF-8
4 spaces
Python 3.9 (venv) (3)


Sign_language
Version control
Current File
Project
_virtualenv.py
data_collection.py
test.py
Sign_language
DATA
A
B
C
Model
keras_model.h5
labels.txt
venv library root
data_collection.py
test.py
External Libraries
imgWhite[hGap:hCal + hGap, :] = imgResize
prediction, index = classifier.getPrediction(imgWhite, draw=False)
cv2.rectangle(imgOutp
cv2.putText(imgOutput
cv2.rectangle(imgOutp
cv2.imshow("ImageCrop
cv2.imshow("ImageWhit
cv2.imshow("Image", imgOu
cv2.waitKey(1)
if hands
Image
A
ImageWhite
Run
/step
Sign_language › test.py
CRLF UTF-8 4 spaces Python 3.9 (venv) (3)


I Love U
DELL
AMISH
AMISH

Conlusion

Sign Language is a tool to reduce the communication gap between deaf-mute people and normal people. This system which is proposed above gives the methodology which aims to do the same as the two-way communication is possible. This method proposed here facilitates the conversion on the sign into text for now and with improvements will convert text to speech. This overcomes the requirement of a translator since real time conversion is used. The system acts as a voice of the person who is deaf-mute. This project is a step towards helping specially challenged people. This will be further enhanced by making it more user friendly, efficient, portable, compatible for more signs and as well as dynamic signs. This can be further improved so as to make it compatible for the mobile phones using the built-in camera of the phone. We can increase the distance at which it can be used by using a longer trans- receiver module or over Wi-Fi.

Future Works

A potential system for further improvement that builds on the present system and connects to the Google Translate text-to-speech API via gTTS (Google Text-to-Speech), a Python library, and a CLI utility to convert text in to speech. After that, it is translated into a number of regional languages, creating a multilingual system. So that the system can communicate in both directions—i.e., translate from plain English to sign language and vice versa—the image processing part needs to be improved. We try to identify signs that involve movement. In addition, our attention is directed to the conversion of a sequence of movements to text, words, and sentences, and finally to audible speech. We are working to integrate our system into products like AR/VR glasses like in Apple vision Pro were in we can Launch our system and use it for translations

Applications

1. Assistive Technology: The system can be used as assistive technology to help individuals with hearing and speech disabilities communicate more effectively in daily life, including in schools, workplaces, and social settings.

2. Translation Services through VR/AR glasses: The system can be used to translate sign language into other languages, making it easier for individuals with help of VR classes like Apple Vsison Pro were are progarm cna be launched and used..

3. Customer Service: The system can be used in customer service settings to provide support to individuals with hearing and speech disabilities, such as through chatbots or virtual assistants.

In summary, the Machine Learning-based Sign Language Recognition System has numerous real-life applications that can significantly improve the lives of individuals with hearing and speech disabilities, promoting inclusivity and accessibility in various aspects of daily life.

References

1. T. Shanableh, "Two-Stage Deep Learning Solution for Continuous Arabic Sign Language Recognition Using Word Count Prediction and Motion Images," IEEE Transactions.[2023]

2. M. Faisal, M. Alsulaiman, M. Mekhtiche, B. M. Abdelkader, M. Algabri, T. B. S. Alrayes, G. Muhammad, H. Mathkour, W. Abdul, Y. Alohali, M. Al-Hammadi, H. Altaheri, and T. Alfakih, "Enabling Two-Way Communication of Deaf Using Saudi Sign Language," IEEE.[2023]

3. C. O. Sosa-Jiménez, H. V. Ríos-Figueroa, and A. L. Solís-González-Cosío, "A Prototype for Mexican Sign Language Recognition and Synthesis in Support of a Primary Care Physician[2022].

4. S. M. Kamal, Y. Chen, S. Li, X. Shi, and J. Zheng, "Technical Approaches to Chinese Sign Language Processing: A Review," in IEEE Transactions[2019]..

5. Harditya, Arya. (2020). Indonesian Sign Language (BISINDO) As Means to Visualize Basic Graphic Shapes Using Teachable Machine. 10.2991/assehr.k.201202.045.

Reference

1. Caixia Wu, Chong Pan,YufengJin, Shengli Sun, and Guangyi Shi Shaoxing, "Improvement of Chinese Sign Language Translation System based on Collaboration of Arm and Finger Sensing Nodes", At The 6th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems June 19-22, 2016.

2. Poonam Chavan 1, Prof. Tushar Ghorpade2, Prof. Puja Padiya, "Indian Sign Language to Forecast Text using Leap Motion Sensor and RF Classifier" at 2016 Symposium on Colossal Data Analysis and Networking (CDAN).

3. Ms.Manisha D. Raut, Ms. Pallavi Dhok, Mr. Ketan Machhale, Ms.Jaspreet Manjeet Hora, "A System forRecognition of Indian SignLanguage for Deaf People using Otsu's Algorithm", International Research Journal of Engineering and Technology (IRJET).

4. C. Swapna, Shabnam Shaikh "Literature survey on hand gesture recognition for video processing" International Journal of Advanced Research in Computer Sci.

Thank You