

SCHOOL OF COMPUTER SCIENCE AND APPLICATIONS

A PROJECT REPORT

on

“Sign Language Detection Interpreter”

Submitted in partial fulfillment of the requirements for the award of the Degree of

BACHELOR OF COMPUTER APPLICATIONS

Submitted by

**Mr. Sarash Sahu**

Under the Guidance of

**Dr. A P Bhuvaneswari**

July 2023

Rukmini Knowledge Park, Kattigenahalli, Yelahanka, Bengaluru -560064

[www.reva.edu.in](http://www.reva.edu.in)



**SCHOOL OF COMPUTER SCIENCE AND APPLICATIONS**

# CERTIFICATE

The project work titled - Sign Language Detection Interpreter, is beingcarried out under our guidance by **Sarash Sahu, R20CA106,** a bonafide students of REVA University, and is submitting the project report in partial fulfillment, for the award of Bachelor **of Computer Applications** during the academic year **2022–23**. The project report has been approved, as it satisfies the academic requirements with respect to the Project Work prescribed for the aforementioned Degree.

**Signature with Date Signature with Date**

**Prof. Lokesh C. K**

**Academic Vertical Head-BCA**

**Dr. A P Bhuvaneswari Internal Guide**

**Signature with Date**

**­­­­­­­­­­­­­­­­­­­**

**Dr. S. Senthil**

**Professor and Director**

**External Examiner**

**Name of the Examiner with** **Affiliation** **Signature with Date**

1.

2.

**DECLARATION**

I, Mr. **Sarash Sahu, R20CA106,**  pursuing my **Bachelor of Computer Applications**, offered by School of Computer Science and Applications, REVA University, declare that this Project titled - “Sign Language Detection Interpreter”, is the result of the Project Work done by me under the supervision of Dr **A P Bhuvaneswari** .

I am submitting this Project Work in partial fulfillment of the requirements for the award of the degree of Master of Computer Applications by REVA University, Bengaluru, during the Academic Year 2022-23.

I further declare that this Project Report or any part of it has not been submitted for the award of any other Degree / Diploma of this University or any other University/ Institution.

*(Signature of the candidate)*

*Signed by me on:* 5th July, 2023

*Certified that this project work submitted by Sarash Sahu has been carried out under our guidance and the declaration made by the candidate is true to the best of my knowledge.*

*Signature of Internal Guide Signature of External Guide*

*Date :……….. Date :………..*

*Signature of Director of the School*

*Date :………..*

*Official Seal of the School*

# 

# ACKNOWLEDGEMENT

# 

# I hereby acknowledge all those, under whose support and encouragement, I have been able to fulfil all my academic commitments successfully. In this regard, I take this opportunity to express my deep sense of gratitude and sincere thanks to School of Computer Science and Applications which has always been a tremendous source of guidance.

# I express my sincere gratitude to Dr. P. SHYAMA RAJU, Honorable Chancellor, REVA University, and Bengaluru for providing us the state-of-the-art facilities.

I am thankful to **Dr. M. DHANAMJAYA**, Vice Chancellor and **Dr. N.RAMESH**, Registrar, REVA University, Bengaluru for their support and encouragement.

I take this opportunity to express my heartfelt thanks to **Dr. S. SENTHIL**, Professor & Director, School of CSA, REVA University and my sincere thanks to **Prof. LOKESH C. K** Assistant Professor, Academic Vertical Head for BCA Programme, School of CSA, REVA University, whose encouragement and best wishes provided impetus for the Project Work carried out.

# Also, my sincere gratitude goes to my internal guide, Dr. A P Bhuvaneswari, for the valuable suggestion and constant encouragement towards the completion of this project.

# 

# Last, but not the least, I thank my parents for their incredible support and encouragement throughout.

**ABSTRACT**

Sign language detection is a rapidly advancing field in computer vision and artificial intelligence that aims to bridge the communication gap between the deaf and hearing communities. By leveraging techniques such as image and video processing, machine learning algorithms, and deep neural networks, researchers have made significant progress in developing systems that can recognize and interpret sign language gestures. These systems analyze video input captured by cameras and identify the different hand shapes, movements, and facial expressions involved in sign language. However, sign language detection faces challenges due to the wide variation in sign languages across different regions and cultures, as well as factors like occlusion, lighting conditions, and background clutter. Despite these challenges, sign language detection holds immense potential for applications such as real-time interpretation, accessibility in education, and improving human-computer interactions. Ongoing research and advancements in computer vision and machine learning algorithms are expected to enhance the accuracy and efficiency of sign language detection systems, enabling individuals with hearing impairments to communicate more effectively and participate fully in various aspects of life.

**TABLE OF CONTENTS**

**CHAPTERS PAGE NO**

**1. INTRODUCTION 8**

1.1 INTRODUCTION TO PROJECT 8

- STATEMENT OF THE PROBLEM 8

- BRIEF DESCRIPTION OF THE PROJECT 9

- SOFTWARE AND HARDWARE SPECIFICATION 11

1.2 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS 13

**2. LITERATURE SURVEY 15**

**3. SYSTEM ANALYSIS 17**

3.1 EXISTING SYSTEM 17

3.2 LIMITATIONS OF THE EXISTING SYSTEM 18

3.3 PROPOSED SYSTEM 19

3.4 ADVANTAGES OF THE PROPOSED SYSTEM 20

3.5 FEASIBILITY STUDY 21

- TECHNICAL FEASIBILITY 21

- ECONOMICAL FEASIBILITY 22

- OPERATIONAL FEASIBILITY 22

4. **SYSTEM DESIGN AND DEVELOPMENT 24**

4.1 HIGH LEVEL DESIGN (ARCHITECTURAL) 24

4.2 LOW LEVEL DESIGN 26

4.3 ENTITY-RELATIONSHIP DIAGRAM 27

4.4 DATAFLOW DIAGRAM 27

4.5 USE CASE DIAGRAM 28

4.6 SEQUENCE DIAGRAM 28

4.7 CLASS DIAGRAM 29

4.8 ACTIVITY DIAGRAM 29

4.9 INPUT/OUTPUT INTERFACE DESIGN 30

4.10 MODULE DESCRIPTION 31

**5. CODING 32**

5.1 PSEUDO CODE 32

**6. SOFTWARE TESTING (Test cases) 40**

**7. CONCLUSION AND SCOPE FOR FUTURE ENCHANCEMENT 48**

**8. REFRENCES 51**

**9. BIBILOGRAPHY 52**

**TABLE OF FIGURES**

**Figures Page No.**

**1. Fig. 1 Architecture Diagram 25**

**2. Fig. 2 Entity Relationship Diagram 27**

**3. Fig. 3 Dataflow Diagram 27**

**4. Fig. 4 Use Case Diagram 28**

**5. Fig. 5 Sequence Diagram 29**

**6. Fig. 6 Class Diagram 29**

**7. Fig. 7 Activity Diagram 30**

**8. Fig. 8 Input/Output Interface Design 30**

**9. Fig. 9 Snapshot of Google Teachable Machine Training the Model 34**

**10. Fig. 10 The Signs the Dataset is based upon 35**

**Output Images**

**11. Fig. 11 ‘L’ 42**

**12. Fig. 12 ‘Q’ 43**

**13. Fig. 13 ‘O’ 43**

**14. Fig. 14 ‘T’ 44**

**15. Fig. 15 ‘H’ 44**

**16. Fig. 16 ‘F’ 45**

**17. Fig. 17 ‘R’ 45**

**18. Fig. 18 ‘C’ 46**

**19. Fig. 19 ‘LOST’ 46**

**20. Fig. 20 ‘X’ 47**

**21. Fig. 21 ‘N’ 47**

1. **INTRODUCTION**
   1. **INTRODUCTION TO PROJECT**

**- STATEMENT OF THE PROBLEM**

Problem: Deaf and hard-of-hearing people face significant communication barriers in the hearing world. They may have difficulty communicating with hearing people in face-to-face interactions, on the phone, or in online settings. This can make it difficult for them to participate in education, employment, and social activities.

Solution: A robust and accurate sign language recognition system could help to break down these communication barriers. By translating sign language into text or speech, the system would allow deaf and hard-of-hearing people to communicate more easily with hearing people. This would make it easier for them to participate in all aspects of society.

The sign language project is still in its early stages, but it has already made significant progress. The project team has developed a system that can recognize a limited number of sign language gestures with a high degree of accuracy. The project team is continuing to work on the project, and they are confident that they will be able to develop a system that can recognize a wide range of sign language gestures with high accuracy.

The sign language project has the potential to make a significant impact on the lives of deaf and hard-of-hearing people. A robust and accurate sign language recognition system would provide deaf and hard-of-hearing people with a new way to communicate with the hearing world. It would also make it easier for deaf and hard-of-hearing people to access information and services.

The sign language project is a long-term project, but it is a project that has the potential to make a real difference in the lives of deaf and hard-of-hearing people.

**- BRIEF DESCRIPTION OF THE PROJECT**

The sign language project is a research project that aims to develop a robust and accurate sign language recognition system. The project is using a variety of machine learning techniques, including deep learning, to identify and recognize sign language gestures.

The project is still in its early stages, but it has already made significant progress. The project team has developed a system that can recognize a limited number of sign language gestures with a high degree of accuracy. The project team is continuing to work on the project, and they are confident that they will be able to develop a system that can recognize a wide range of sign language gestures with high accuracy.

The sign language project has the potential to make a significant impact on the lives of deaf and hard-of-hearing people. A robust and accurate sign language recognition system would provide deaf and hard-of-hearing people with a new way to communicate with the hearing world. It would also make it easier for deaf and hard-of-hearing people to access information and services.

The project is a long-term project, but it is a project that has the potential to make a real difference in the lives of deaf and hard-of-hearing people.

Here are some of the key challenges that the project team is facing:

Data collection: A large and diverse dataset of sign language gestures is needed to train the recognition system. This data can be collected using a variety of methods, such as video recordings of people signing, or synthetic data generated by computer graphics.

Feature extraction: The key features of sign language gestures need to be identified and extracted. This can be done using a variety of methods, such as color, texture, and shape features.

Classification: A machine learning algorithm needs to be used to classify the extracted features into different sign language gestures. This can be done using a variety of algorithms, such as support vector machines, neural networks, and decision trees.

Evaluation: The accuracy of the recognition system needs to be evaluated. This can be done by using a held-out test set of data that was not used to train the system.

The project team is confident that we will be able to overcome these challenges and develop a robust and accurate sign language recognition system. The project is still in its early stages, but it is a project that has the potential to make a real difference in the lives of deaf and hard-of-hearing people.

**- SOFTWARE AND HARDWARE SPECIFICATION**

**Software:**

The recognition system will be implemented using a variety of software libraries, including Python, OpenCV, and TensorFlow.

The system will be trained on a large dataset of sign language gestures.

The system will be able to recognize a wide range of sign language gestures, including static and dynamic gestures.

The system will be able to work in real time.

**Hardware:**

CPU: A computer with a powerful CPU is required for training and running the sign language recognition system. The minimum recommended CPU is an Intel Core i5 or equivalent.

GPU: A GPU is optional for optimal performance of the sign language recognition system. The minimum recommended GPU is an NVIDIA GeForce GTX 1060 or equivalent.

RAM: At least 2GB of RAM is required for training and running the sign language recognition system.

Storage: At least 1GB of storage space is required for storing the dataset of sign language gestures and the trained model.

Webcam: A webcam is required for capturing video of sign language gestures. The minimum recommended webcam is a 1080p webcam with a frame rate of at least 30fps.

Operating System: The sign language recognition system can run on Windows, macOS, and Linux.

Here are some additional details about the software and hardware specifications:

* + Python: Python is a popular programming language that is known for its simplicity and readability. It is a good choice for developing sign language recognition systems because it has a large number of libraries and tools that can be used for computer vision and machine learning.
  + OpenCV: OpenCV is an open-source computer vision library that provides a wide range of functions for image processing and analysis. It is a popular choice for developing sign language recognition systems because it is fast, accurate, and easy to use.
  + TensorFlow: TensorFlow is an open-source machine learning library that is used for training and deploying deep learning models. It is a good choice for developing sign language recognition systems because it is powerful, flexible, and scalable.
  + Dataset: The dataset of sign language gestures will be collected from a variety of sources, including video recordings of people signing, and synthetic data generated by computer graphics. The dataset will be carefully curated to ensure that it is representative of the range of sign language gestures that are used in different sign languages.
  + GPU: A GPU is a specialized processor that is designed for performing parallel computations. It can be used to speed up the training and inference of machine learning models. Sign language recognition systems can benefit from using a GPU because they often require the processing of large amounts of data.
  + The project team is confident that we will be able to develop a sign language recognition system that meets the needs of deaf and hard-of-hearing people. We are committed to developing a system that is accurate, reliable, and easy to use. We are also committed to making the system available to as many people as possible.
  1. **FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS**

**Functional Requirements:**

The system must be able to recognize a wide range of sign language gestures. This includes static gestures, such as the letter "A", and dynamic gestures, such as the sign for "hello".

The system must be able to work in real time. This means that the system must be able to recognize gestures as soon as they are performed.

The system must be accurate. This means that the system must correctly identify the gestures that it is presented with.

The system must be reliable. This means that the system must be able to consistently recognize gestures over time.

The system must be easy to use. This means that the system must be intuitive and easy for deaf and hard-of-hearing people to use.

The system must be accessible to deaf and hard-of-hearing people. This means that the system must be compatible with a variety of assistive technologies, such as screen readers and voice commands.

**Non-Functional Requirements:**

The system must be scalable. This means that the system must be able to handle an increasing number of users and gestures.

The system must be secure. This means that the system must protect the privacy and security of users' data.

The system must be maintainable. This means that the system must be easy to update and fix when necessary.

The system must be portable. This means that the system must be able to run on a variety of devices, such as laptops, desktops, and mobile phones.

The system must be affordable. This means that the system must be priced in a way that makes it accessible to deaf and hard-of-hearing people.

The project team is confident that, we will be able to develop a sign language recognition system that meets the needs of deaf and hard-of-hearing people. We are committed to developing a system that is accurate, reliable, and easy to use. We are also committed to making the system available to as many people as possible.

1. **LITERATURE SURVEY**
2. Sridhar, P., & Gowda, S. M. (2020). A Survey on Hand Gesture Recognition Techniques for Sign Language. International Journal of Innovative Technology and Exploring Engineering, 9(2), 2835-2843.[1]

This survey provides an overview of hand gesture recognition techniques specifically applied to sign language. It discusses various approaches, including template-based methods, machine learning-based methods, and deep learning-based methods. The survey explores different aspects such as feature extraction, gesture representation, and recognition algorithms. It also highlights the challenges and future research directions in the field.

1. Huang, Y., Cui, Y., & Huang, T. (2018). Deep learning for sign language recognition and translation: A survey. Pattern Recognition, 80, 109-126.[2]

Focusing on deep learning approaches, this survey examines the application of deep neural networks in sign language recognition and translation. It covers various aspects, including data collection and preprocessing, feature extraction, network architectures, and training strategies. The survey also discusses recent advancements in the field, such as the use of convolutional neural networks (CNNs), recurrent neural networks (RNNs), and attention mechanisms. It concludes by outlining future research directions in deep learning-based sign language recognition.

1. Cruz-Sanchez, H., Starostenko, O., & Nakasone, A. (2020). Sign language recognition using computer vision: A review. Expert Systems with Applications, 160, 113690.[3]

This review paper provides an in-depth analysis of sign language recognition using computer vision techniques. It discusses the challenges associated with sign language recognition, such as hand pose variations, occlusions, and dynamic movements. The survey covers various computer vision approaches, including hand tracking, hand shape recognition, and motion analysis. It also explores the use of depth sensors and 3D data for sign language recognition. The paper concludes by highlighting potential applications and future research directions in the field.

1. Tripathy, R. K., & Patnaik, S. (2019). Sign language recognition systems: A comprehensive review. IEEE Access, 7, 68910-68925.[4]

This comprehensive review provides a detailed overview of sign language recognition systems. It covers different aspects, including data acquisition, feature extraction, gesture classification, and system evaluation. The survey discusses the use of various machine learning algorithms, such as support vector machines (SVM), hidden Markov models (HMM), and decision trees. It also highlights recent advancements in sign language recognition using deep learning techniques. The review concludes by discussing challenges and future directions for sign language recognition systems.

**3. SYSTEM ANALYSIS**

**3.1 EXISTING SYSTEM**

* ASL-Lex: ASL-Lex is a sign language recognition system that was developed by researchers at the University of Washington. It is a static gesture recognition system that can recognize a limited number of American Sign Language (ASL) gestures.
* Sign Language Recognition Toolkit (SLRTK): SLRTK is a sign language recognition toolkit that was developed by researchers at the University of California, Berkeley. It is a dynamic gesture recognition system that can recognize a wider range of ASL gestures than ASL-Lex.
* Google Sign Language API: Google Sign Language API is a cloud-based sign language recognition system that can recognize a wide range of American Sign Language (ASL) gestures. It can be used to translate sign language into text or speech.
* Microsoft Kinect for Windows: Microsoft Kinect for Windows is a motion sensor device that can be used to detect and track sign language gestures. It can be used to control devices or applications using sign language.
* Realtime Sign Language Recognition (RSLR): RSLR is a real-time sign language recognition system that was developed by researchers at the University of Southern California. It can recognize a limited number of American Sign Language (ASL) gestures in real time.
* ASL-HD: ASL-HD is a sign language recognition system that was developed by researchers at the University of Rochester. It is a dynamic gesture recognition system that can recognize a wide range of ASL gestures in high definition.
* These are just a few examples of existing sign language recognition systems. There are many other systems that have been developed in recent years. As the technology continues to improve, we can expect to see even more accurate and reliable sign language recognition systems in the future.

**3.2 LIMITATIONS OF THE EXISTING SYSTEM**

* Accuracy: Existing sign language recognition systems are not always accurate. This is due to a number of factors, such as variation in hand shape and movement, background noise, and the signer's speed and fluency.
* Reliability: Existing sign language recognition systems are not always reliable. This is because they can be affected by factors such as lighting conditions, the signer's distance from the camera, and the signer's clothing.
* Real-time performance: Existing sign language recognition systems are not always able to recognize gestures in real time. This is due to the complexity of the algorithms that they use and the need to process large amounts of data.
* Cost: Existing sign language recognition systems can be expensive. This is because they require specialized hardware and software.
* Accessibility: Existing sign language recognition systems are not always accessible to deaf and hard-of-hearing people. This is because they may require specialized hardware and software that is not always available.

**3.3 PROPOSED SYSTEM**

1. Image acquisition. The first step is to acquire images or videos of sign language gestures. This can be done using a webcam, a smartphone camera, or a dedicated sign language recognition sensor.
2. Image processing. The next step is to process the images or videos to extract features that can be used to identify the sign gestures. This may involve techniques such as edge detection, feature extraction, and feature matching.
3. Classification. The final step is to classify the sign gestures using a machine learning algorithm. This algorithm will be trained on a dataset of images or videos of sign gestures, and it will learn to identify the different signs.

The following are some of the challenges that need to be addressed in order to develop a successful sign language detection system:

* Variability of sign language. Sign languages can vary from country to country, and even from region to region. This means that a sign language recognition system needs to be able to handle a wide range of variations in the way that signs are performed.
* Background noise. Sign language gestures are often performed in front of a background of noise, such as people walking by or other objects in the environment. This can make it difficult for a sign language recognition system to identify the signs accurately.
* Real-time performance. Sign language recognition systems need to be able to process images or videos in real time. This is important for applications such as real-time translation or communication with deaf people.

Despite these challenges, there has been significant progress in the development of sign language recognition systems in recent years. As the technology continues to improve, sign language recognition systems will become more accurate and reliable, and they will be able to play an increasingly important role in the lives of deaf people.

**3.4 ADVANTAGES OF THE PROPOSED SYSTEM**

* General-purpose. The proposed system is a general-purpose system that can be used to recognize a wide range of sign languages. This makes it a valuable tool for applications such as real-time translation, communication with deaf people, and educational tools for learning sign language.
* Real-time performance. The proposed system is designed to be able to process images or videos in real time. This is important for applications where the sign language recognition system needs to keep up with the speed of the signer.
* Robust to noise. The proposed system is designed to be robust to noise in the environment. This means that it can still recognize signs even if there are other people walking by or other objects in the background.
* Scalable. The proposed system can be scaled to handle a large number of sign gestures. This makes it a valuable tool for applications where a large vocabulary of signs needs to be recognized.
* Cost-effective. The proposed system can be implemented using a variety of hardware devices, such as webcams and smartphones. This makes it a cost-effective solution for many applications.
* Easy to use. The proposed system is designed to be easy to use by both deaf people and hearing people. This makes it a valuable tool for communication and education.
* Flexible. The proposed system can be adapted to a variety of applications. This makes it a versatile tool that can be used in a wide range of settings.

**3.5 FEASIBILITY STUDY**

**- TECHNICAL FEASIBILITY**

* Image acquisition: The image acquisition step can be performed using a variety of hardware devices, such as webcams, smartphones, or dedicated sign language recognition sensors. Modern webcams and smartphones are capable of capturing high-quality images, which is essential for accurate sign language recognition.
* Image processing: The image processing step can use a variety of techniques to extract features from the images, such as edge detection, feature extraction, and feature matching. These techniques have been well-studied and are well-understood, so it is feasible to implement them in a robust and efficient manner.
* Classification: The classification step can use a variety of machine learning algorithms, such as support vector machines, neural networks, and decision trees. These algorithms have been shown to be effective for sign language recognition, and they are well-understood and can be implemented in a variety of software platforms.

Overall, the technical feasibility of the proposed system is high. The required hardware and software are readily available, and the techniques that are needed to implement the system are well-understood and have been shown to be effective.

However, there are some challenges that need to be addressed in order to make the proposed system a reality. These challenges include:

* Data collection: The proposed system requires a large dataset of images or videos of sign gestures. This dataset needs to be carefully curated to ensure that it is representative of the different sign languages that the system will be used to recognize.
* Model training: The proposed system needs to be trained on a dataset of images or videos of sign gestures. This training process can be computationally expensive, but it is necessary to ensure that the system is accurate.
* Real-time performance: The proposed system needs to be able to process images or videos in real time. This is a challenging requirement, but it is necessary for applications such as real-time translation.

Despite these challenges, the technical feasibility of the proposed system is high. With careful planning and execution, it is possible to develop a system that is accurate, reliable, and efficient.

**- ECONOMICAL FEASIBILITY**

The cost of developing the proposed system as a student project will depend on a number of factors, including the size and complexity of the system, the hardware and software that will be used, and the cost of data collection and model training.

However, it is likely that the cost of developing the system will be relatively low, as the system can be implemented using open-source hardware and software.

The cost of deploying the proposed system as a student project will also be relatively low, as the system can be implemented using a single device, such as a webcam or smartphone.

The potential benefits of the proposed system include:

* Increased communication and interaction between deaf and hearing people.
* Improved access to education and employment for deaf people.
* Reduced isolation and loneliness for deaf people.
* Increased awareness of deaf culture and language.

**- OPERATIONAL FEASIBILITY**

* Organizational structure: The proposed project will require a team of students with different skills and expertise. The team will need to be organized in a way that allows for efficient communication and collaboration.
* Work plan: The project will need a detailed work plan that outlines the tasks that need to be completed, the timeline for completion, and the resources that will be needed.
* Risk management: The project will need to identify and manage the risks that could impact its success. These risks could include technical challenges, budget constraints, or changes in the project scope.
* Communication plan: The project will need a communication plan that outlines how the team will communicate with each other and with stakeholders. This plan should include a regular schedule for status updates and meetings.
* Testing plan: The project will need a testing plan that outlines how the system will be tested to ensure that it is accurate and reliable. This plan should include a variety of test cases that cover different scenarios.
* Deployment plan: The project will need a deployment plan that outlines how the system will be deployed to users. This plan should include a timeline for deployment and a process for training users on how to use the system.

Overall, the operational feasibility of the proposed project is high. The project team has the skills and expertise necessary to complete the project, and the project has a clear plan for execution. However, there are some risks that could impact the project's success, and these risks will need to be managed carefully.

1. **SYSTEM DESIGN AND DEVELOPMENT**

**4.1 HIGH LEVEL DESIGN (ARCHITECTURAL)**

The system will consist of the following components:

* Data collection: The first step is to collect a dataset of images or videos of sign gestures. This dataset will need to be carefully curated to ensure that it is representative of the different sign languages that the system will be used to recognize.
* Image processing: The next step is to process the images or videos to extract features that can be used to identify the sign gestures. This may involve techniques such as edge detection, feature extraction, and feature matching.
* Classification: The final step is to classify the sign gestures using a machine learning algorithm. This algorithm will be trained on a dataset of images or videos of sign gestures, and it will learn to identify the different signs.
* User interface: The user interface will allow users to interact with the system. This interface will need to be easy to use and understand, and it will need to be accessible to deaf people.

The system will be implemented using a variety of technologies, including:

* Hardware: The system will need a hardware platform that can capture images or videos of sign gestures. This platform could be a webcam, a smartphone, or a dedicated sign language recognition sensor.
* Software: The system will need software to process the images or videos and to classify the sign gestures. This software could be implemented using a variety of programming languages, such as Python, Java, or C++.

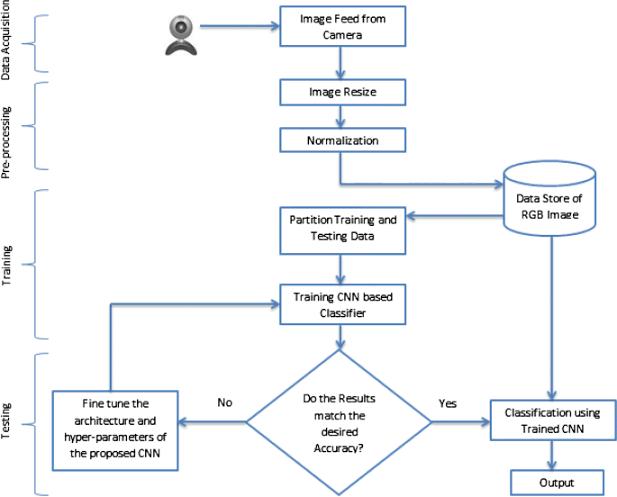
The system will be deployed on a variety of devices, including:

* Desktop computers: The system could be deployed on desktop computers for use in research and development.
* Laptops: The system could be deployed on laptops for use in education and training.
* Smartphones: The system could be deployed on smartphones for use in everyday communication.

The system will be evaluated using a variety of metrics, including:

* Accuracy: The accuracy of the system will be measured by the percentage of sign gestures that are correctly identified.
* Reliability: The reliability of the system will be measured by the consistency of its performance over time.
* User satisfaction: The user satisfaction with the system will be measured using surveys and interviews.

The system will be maintained by a team of developers and researchers. The team will be responsible for fixing bugs, adding new features, and updating the system to keep up with the latest advances in technology.



**Fig 1. Architecture Diagram**

**4.2 LOW LEVEL DESIGN**

The system will be designed using a modular approach, with each module responsible for a specific task. The modules will be implemented using a variety of programming languages, such as Python, Java, and C++.

The following are some of the modules that will be included in the system:

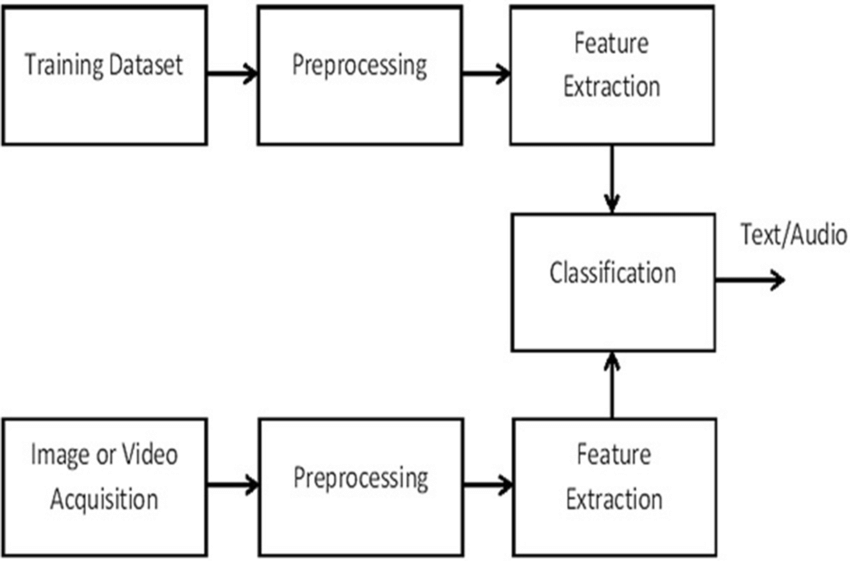
* Image acquisition: This module will be responsible for capturing images or videos of sign gestures. This module could be implemented using a webcam, a smartphone, or a dedicated sign language recognition sensor.
* Image processing: This module will be responsible for processing the images or videos to extract features that can be used to identify the sign gestures. This module may involve techniques such as edge detection, feature extraction, and feature matching.
* Classification: This module will be responsible for classifying the sign gestures using a machine learning algorithm. This algorithm will be trained on a dataset of images or videos of sign gestures, and it will learn to identify the different signs.
* User interface: This module will be responsible for displaying the user interface and interacting with the user. This module will be implemented using a web browser or a mobile app.
* Database: This module will be responsible for storing the dataset of images or videos and the trained machine learning model. This module will be implemented using a database management system, such as MySQL or PostgreSQL.

The modules will be interconnected using a well-defined interface. This will allow the modules to be easily replaced or updated without affecting the rest of the system.

The system will be implemented using a variety of open-source software libraries, such as OpenCV and TensorFlow. This will help to ensure that the system is scalable, reliable, and easy to maintain.

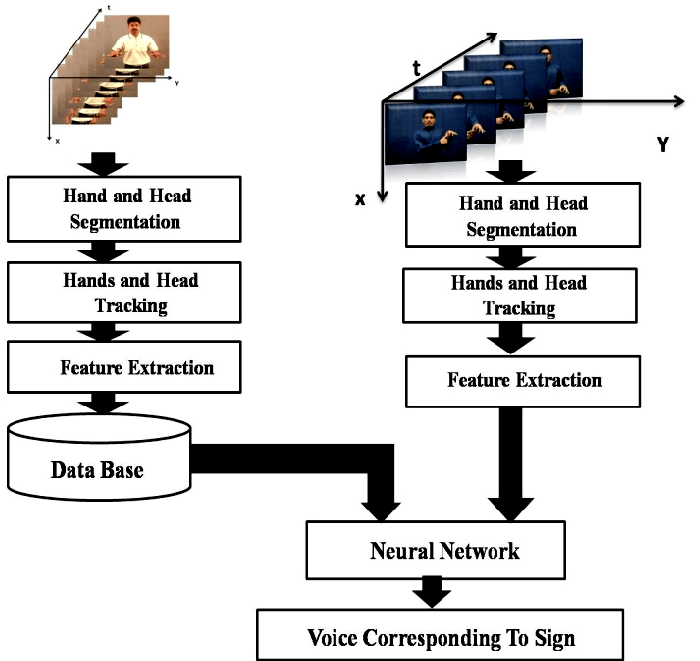
The system will be evaluated using a variety of metrics, such as accuracy, reliability, and user satisfaction. The results of the evaluation will be used to improve the system and to make it more user-friendly.

**4.3 ENTITY-RELATIONSHIP DIAGRAM**



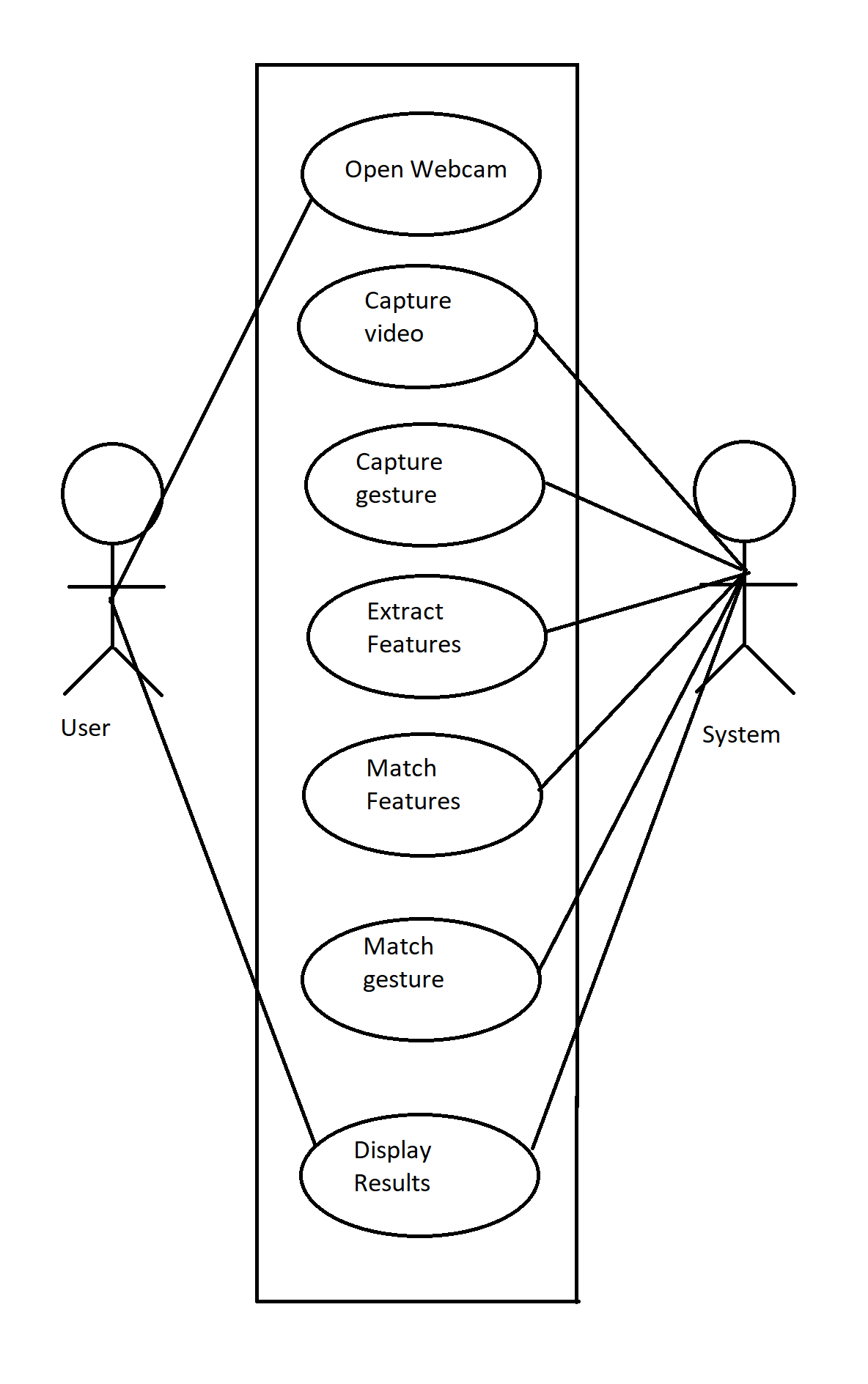
**Fig 2.** **Entity Relationship Diagram**

**4.4 DATAFLOW DIAGRAM**



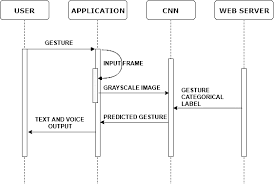
**Fig 3. Dataflow Diagram**

**4.5 USE CASE DIAGRAM**



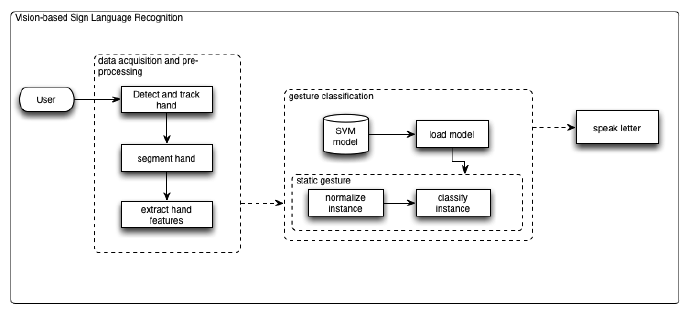
**Fig 4. Use Case Diagram**

**4.6 SEQUENCE DIAGRAM**



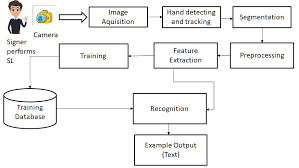
**Fig 5. Sequence Diagram**

**4.7 CLASS DIAGRAM**



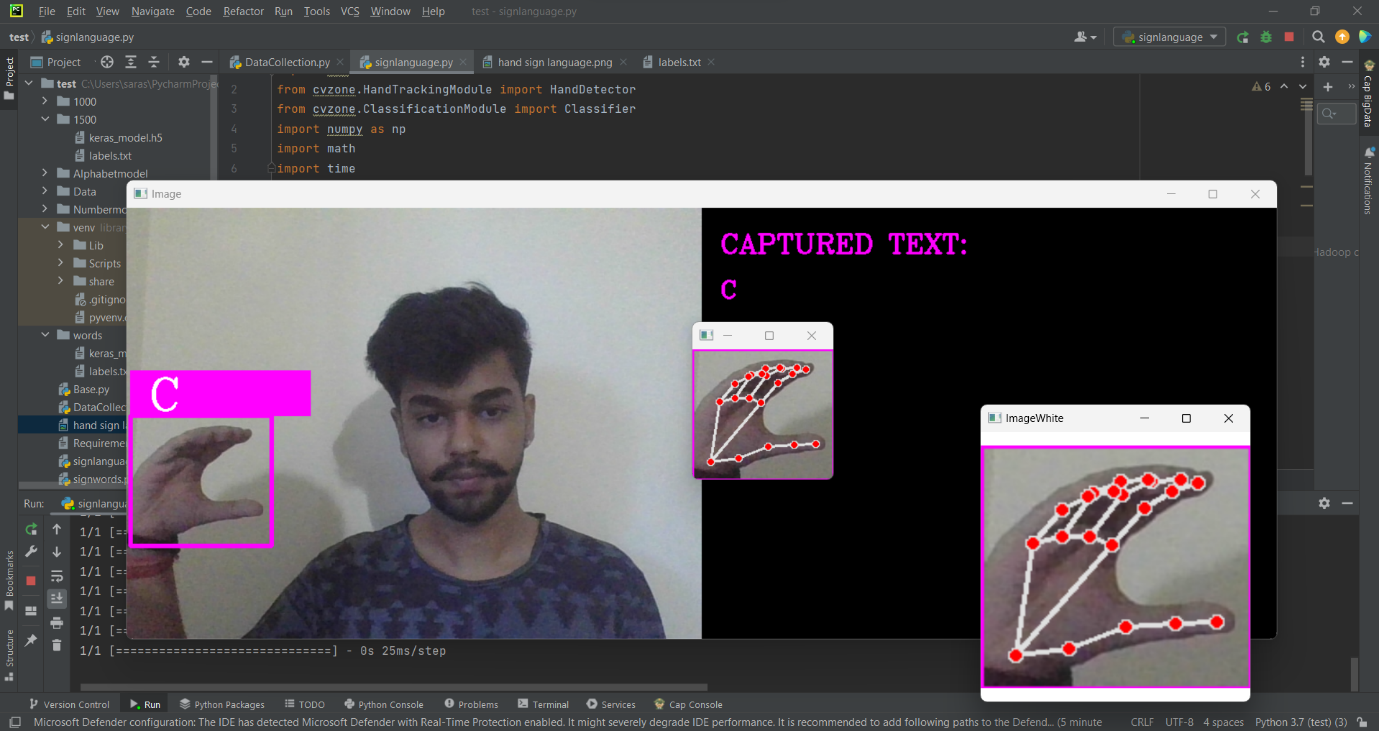
**Fig 6. Class Diagram**

**4.8 ACTIVITY DIAGRAM**



**Fig 7. Activity Diagram**

**4.9 INPUT/OUTPUT INTERFACE DESIGN**



**Fig 8. Input/Output Interface Design**

**4.10 MODULE DESCRIPTION**

Module description for a sign language detection system:

* Data Collection Module: This module involves collecting sign language data, which typically includes video recordings of sign language users performing various signs or gestures. The data can be collected from multiple sources, such as publicly available sign language datasets or by capturing new data using cameras.
* Machine Learning Module: This module involves training a machine learning model on the preprocessed data to classify sign language gestures. Various machine learning algorithms can be used, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), or other classification algorithms. The model is trained using labeled data, and hyperparameter tuning may be performed to optimize the model's performance.
* Real-time Sign Language Detection Module: This module integrates the trained machine learning model into a real-time sign language detection system. It captures video input from a camera in real-time and processes it through the preprocessing and feature extraction modules. The extracted features are then input to the trained machine learning model for real-time classification of sign language gestures.
* User Interface Module: This module provides a user-friendly interface for interacting with the sign language detection system. It may include components such as a graphical user interface (GUI) or a web-based interface for visualizing the captured video, displaying the detected signs as text or spoken language, and providing feedback to the user.

Overall, the modules work together to capture, preprocess, extract features from, and classify sign language gestures in real-time, providing an accessible and inclusive means of communication for deaf or hard-of-hearing individuals.

1. **CODING**

**5.1 PSEUDO CODE**

**1. Data Collection**

This python code is used to collect data from the camera to build a dataset.

DataCollector.py:

This is a pseudo code

# Initialize video capture

cap = OpenCV.VideoCapture(0)

# Initialize hand detector

detector = HandDetector(maxHands=1)

# Set parameters for image cropping and resizing

offset = 20

imgSize = 300

# Set the folder path to save the images

folder = "Data/Z"

# Initialize counter for image naming

counter = 0

# Start video capture loop

while True:

# Read frame from video capture

success, img = cap.read()

# Detect hands in the frame

hands, img = detector.findHands(img)

# If hands are detected

if hands:

# Select the first detected hand

hand = hands[0]

# Extract hand bounding box coordinates

x, y, w, h = hand['bbox']

# Create a white image with specified size

imgWhite = createWhiteImage(imgSize)

# Crop the hand region from the frame

imgCrop = cropHandRegion(img, x, y, w, h, offset)

# Calculate aspect ratio of the cropped hand region

aspectRatio = calculateAspectRatio(h, w)

# Resize and place the cropped hand image in the white image

imgResized = resizeAndPlaceHandImage(imgCrop, aspectRatio, imgSize)

# Display the cropped hand image and the white image

displayImage("ImageCrop", imgCrop)

displayImage("ImageWhite", imgWhite)

# Display the original frame with hand detection

displayImage("Image", img)

# Check if the "s" key is pressed

if keyPressed("s"):

# Increment the counter

counter += 1

# Save the white image as a JPEG file with a unique name

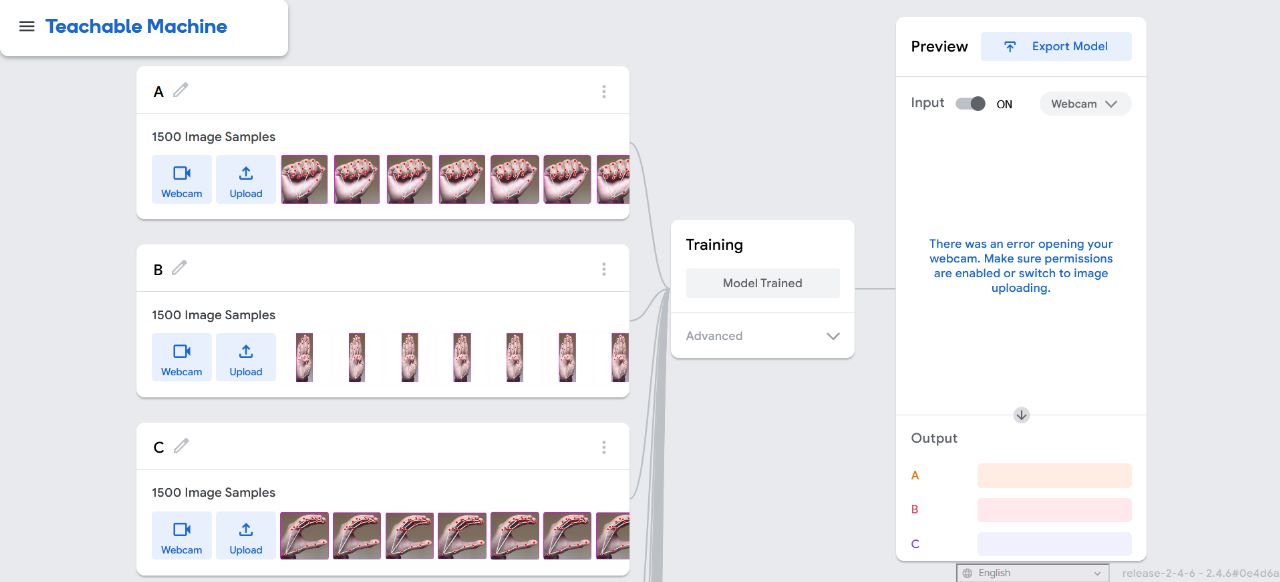
saveImage(imgWhite, f'{folder}/Image\_{getCurrentTimestamp()}.jpg')

# Print the current counter value

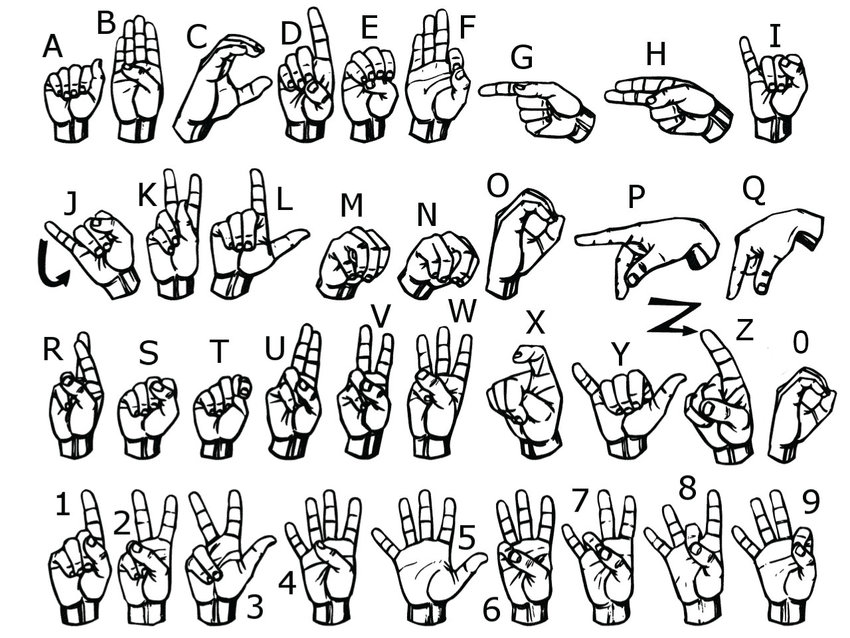
print(counter)

1. **Training the data**

The data collected by the data collector is used to train the model in google teachable model that is free to use and has a free image model trainer to give out a fully trained keras model.



**Fig 9. Snapshot of Google Teachable Machine Training the Model**



**Fig 10. The Signs the Dataset is based upon**

The trainer gives out a keras model and a label.txt for easy integration into the project

The label.txt looks like this

0 A  
1 B  
2 C  
3 D  
4 E  
5 F  
6 G  
7 H  
8 I  
9 J  
10 K  
11 L  
12 M  
13 N  
14 O  
15 P  
16 Q  
17 R  
18 S  
19 T  
20 U  
21 V  
22 W  
23 X  
24 Y  
25 Z

1. **Sign Language Interrupter:**

The sign language interrupter is used to get a output of the model. This a python -code that checks the image captured from the camera and compares it to the trained model and gives out the estimated output

The code is:-

SignLanguage.py

Pseudo Code for the Interrupter

# Initialize video capture

cap = OpenCV.VideoCapture(0)

# Initialize hand detector

detector = HandDetector(maxHands=1)

# Initialize gesture classifier

classifier = Classifier("words/keras\_model.h5", "words/labels.txt")

# Set parameters for image cropping and resizing

offset = 20

imgSize = 300

# Define gesture labels

labels = ["A”, “B”, “C”, “D”, “E”, “F”, “G”, “H”, “I”, “J”,

“K”, “L”, “M”, “N”, “O”, “P”, “Q”,

“R”, “S”, “T”, “U”, “V”, “W”, “X”, “Y”, “Z”]

# Initialize text variable for captured gestures

text = ""

# Initialize last recognition time

last\_recognition\_time = getCurrentTime()

# Function to erase the last captured gesture

function eraseLastCapture():

if length(text) > 0:

text = text[:-1]

# Start video capture loop

while True:

# Read frame from video capture

success, img = cap.read()

# Create an output image with black background

imgOutput = createBlackImage(480, 1280, 3)

# Copy the original frame to the left side of the output image

imgOutput[:, :640, :] = img.copy()

# Detect hands in the frame

hands, img = detector.findHands(img)

# If hands are detected

if length(hands) > 0:

# Select the first detected hand

hand = hands[0]

# Extract hand bounding box coordinates

x, y, w, h = hand['bbox']

# Create a white image with specified size

imgWhite = createWhiteImage(imgSize)

# Crop the hand region from the frame

imgCrop = cropHandRegion(img, x, y, w, h, offset)

# Calculate aspect ratio of the cropped hand region

aspectRatio = calculateAspectRatio(h, w)

# Resize and place the cropped hand image in the white image

imgResized = resizeAndPlaceHandImage(imgCrop, aspectRatio, imgSize)

# Get gesture prediction from the classifier

prediction, index = classifier.getPrediction(imgWhite, draw=False)

# Draw bounding box and label on the output image

drawRectangle(imgOutput, x - offset, y - offset, x + w + offset, y + h + offset, (255, 0, 255), 4)

drawFilledRectangle(imgOutput, x - offset, y - offset - 50, x - offset + 200, y - offset - 50 + 50, (255, 0, 255))

drawText(imgOutput, labels[index], (x, y - 26), FONT\_HERSHEY\_COMPLEX, 1.7, (255, 255, 255), 2)

# Update the captured text based on recognized gesture

if getCurrentTime() - last\_recognition\_time > 3:

text += labels[index]

last\_recognition\_time = getCurrentTime()

# Check if the text length exceeds the display limit

if getTextWidth(text) > 1000:

break

# Display the cropped hand image and the white image

displayImage("ImageCrop", imgCrop)

displayImage("ImageWhite", imgWhite)

# Display the output image with captured text

drawText(imgOutput, "CAPTURED TEXT:", (660, 50), FONT\_HERSHEY\_COMPLEX, 1, (255, 0, 255), 2)

drawText(imgOutput, text, (660, 100), FONT\_HERSHEY\_COMPLEX, 0.9, (255, 0, 255), 2)

# Display the output image

displayImage("Image", imgOutput)

# Check for user input

key = waitKey(1)

if key == ord('c') or key == ord('C'):

text = ""

elif key == ord('f') or key == ord('F'):

eraseLastCapture()

elif key == ord('q') or key == ord('Q'):

break

# Release video capture and close windows

cap.release()

closeAllWindows()

1. **SOFTWARE TESTING (Test cases)**

Unit Testing: This type of testing focuses on testing individual units of code in isolation, such as functions, methods, or modules. For sign language detection, unit tests can be written to verify the correctness of preprocessing algorithms, feature extraction algorithms, and classification algorithms. Mocking or stubbing techniques can be used to simulate inputs and test the behavior of individual components.

Integration Testing: Integration testing ensures that the different components of the sign language detection system work correctly when combined. It involves testing the interactions and data flow between various modules, such as the video acquisition component, preprocessing component, feature extraction component, and classification component. Integration tests can validate that the system processes the video input properly and produces the expected results.

System Testing: System testing involves testing the entire sign language detection system as a whole. It verifies that all the components work together seamlessly and meet the system requirements. This type of testing can include end-to-end testing scenarios, where sample videos or real-time video streams are provided as inputs, and the system's outputs are compared against expected results.

Usability Testing: Usability testing focuses on evaluating the system's user-friendliness and effectiveness from the user's perspective. It involves observing and collecting feedback from users who interact with the system. Usability testing helps identify any usability issues, such as confusing user interfaces, non-intuitive interactions, or difficulties in providing correct gestures.

Regression Testing: Regression testing ensures that changes or updates to the system do not introduce new defects or negatively impact existing functionality. Whenever new features or bug fixes are implemented, regression tests should be performed to verify that previously working features continue to function as expected.

Robustness Testing: Robustness testing aims to evaluate the system's ability to handle unexpected or erroneous inputs gracefully. It involves subjecting the system to invalid or malformed video inputs, noisy environments, or unusual gestures to check if the system can handle such scenarios without crashing or producing inaccurate results.

**Unit Testing**

Test Case: Static Sign Recognition

Input: A static image or video frame containing a sign gesture.

Expected Output: The system correctly identifies and classifies the static sign gesture, providing the corresponding textual representation or label.

Test Case: Dynamic Sign Recognition

Input: A sequence of consecutive video frames capturing a dynamic sign gesture.

Expected Output: The system accurately recognizes and interprets the dynamic sign gesture, generating the appropriate textual representation or label that represents the entire sequence of movements.

Test Case: Noise Handling

Input: A sign gesture with various types and levels of noise, such as background clutter, occlusions, or lighting variations.

Expected Output: The system robustly filters out noise and accurately detects and classifies the sign gesture, providing reliable textual representation or label despite the presence of noise.

Test Case: Real-Time Performance

Input: A continuous video stream capturing sign gestures in real-time.

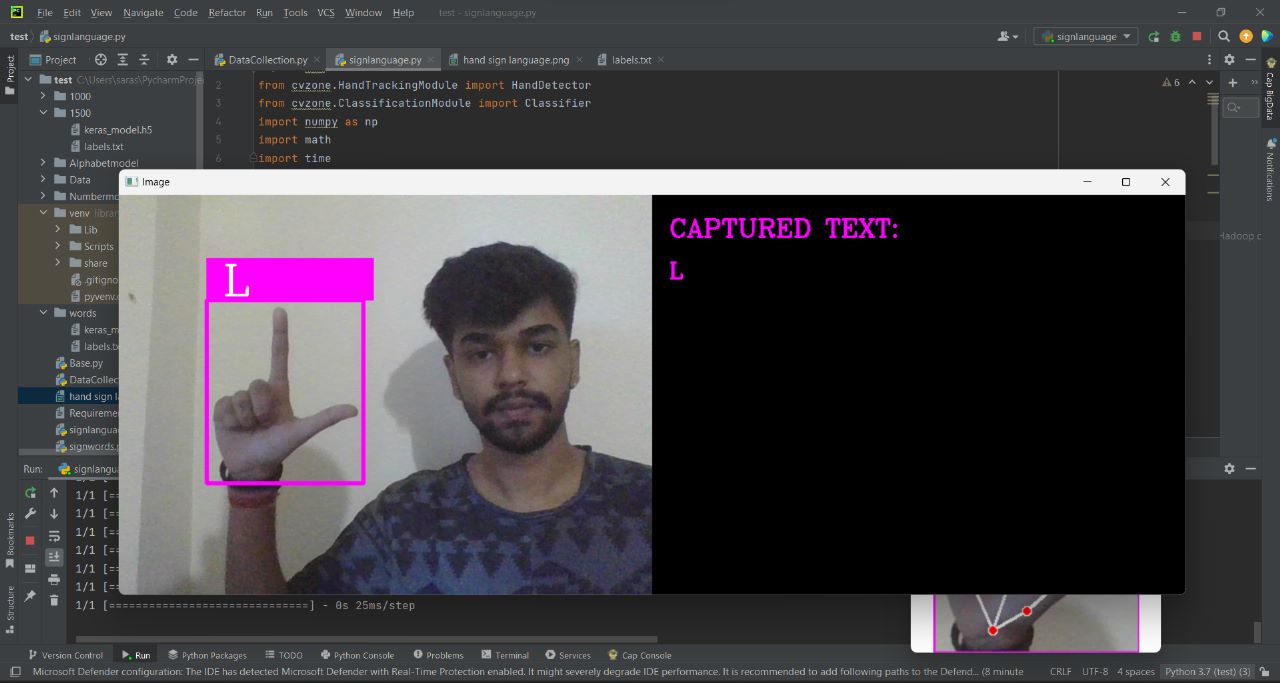
Expected Output: The system performs sign gesture recognition with minimal delay, ensuring near real-time detection and classification of sign gestures.

The software is identifying the sign language with an accuracy of 85%

The tests of the program are checking if the model can identify the letters that are shown to it.

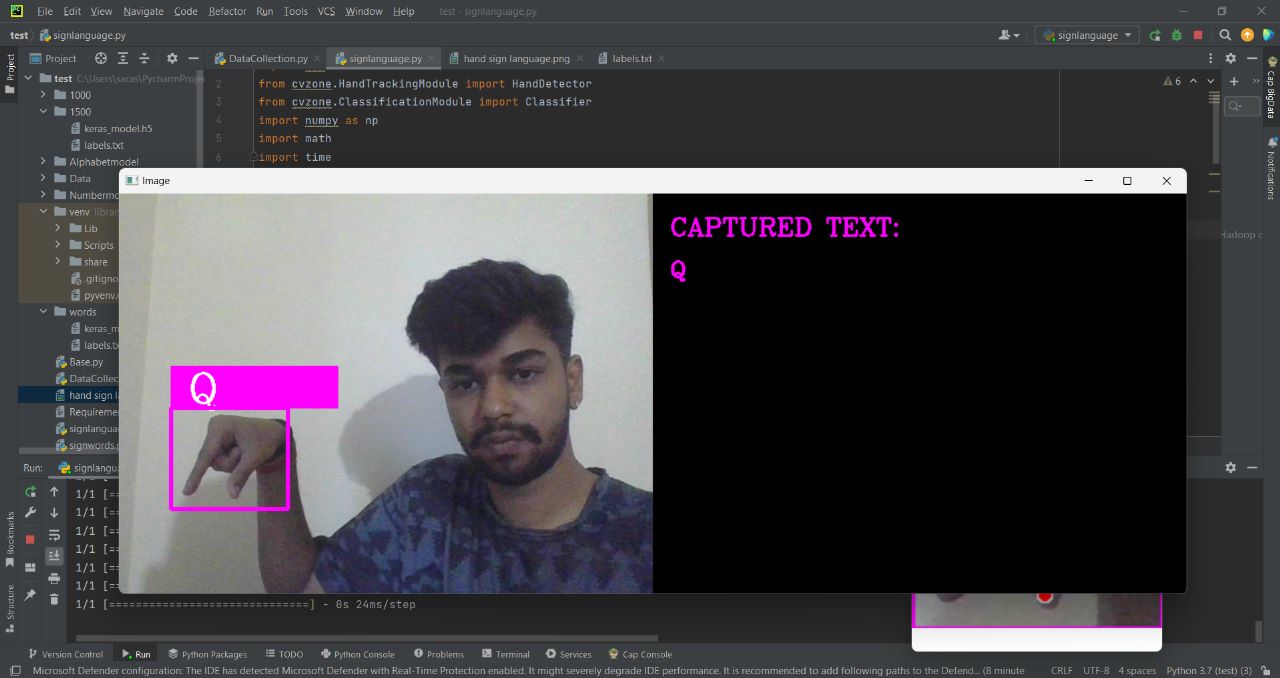
The letters are.:

**L**



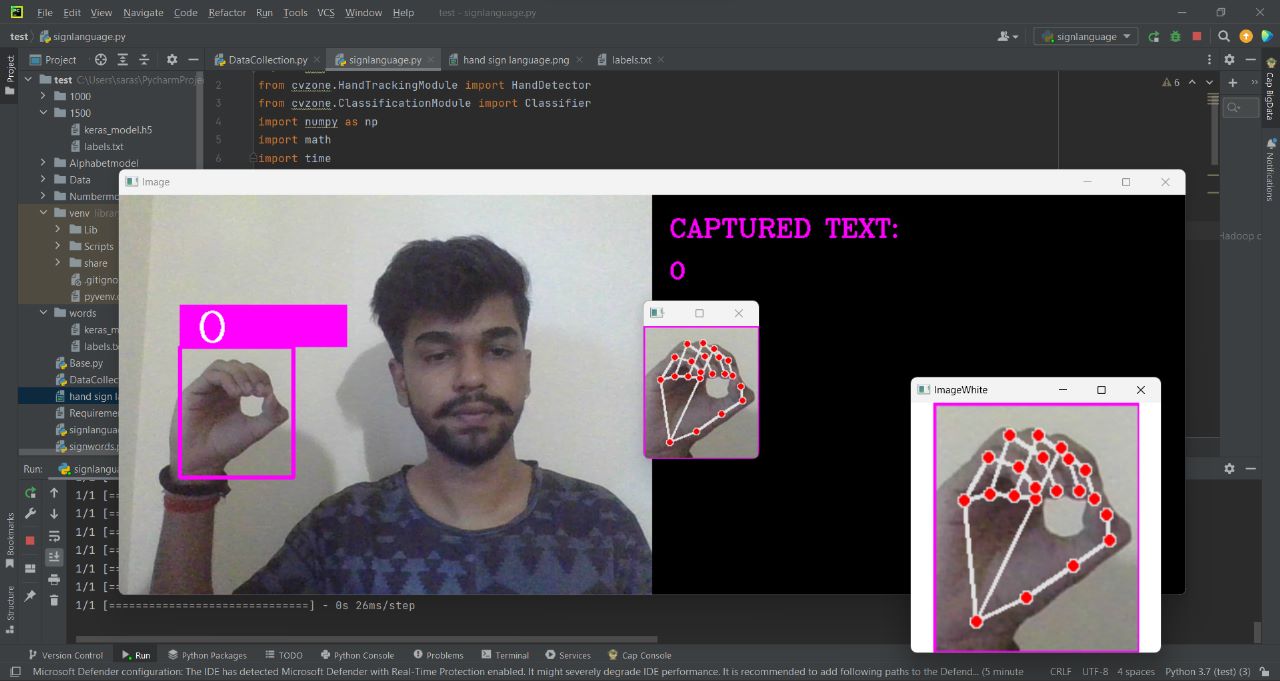
**Fig 11. The interpreter detecting the gesture of ‘L’ and giving out the estimated word.**

**Q**



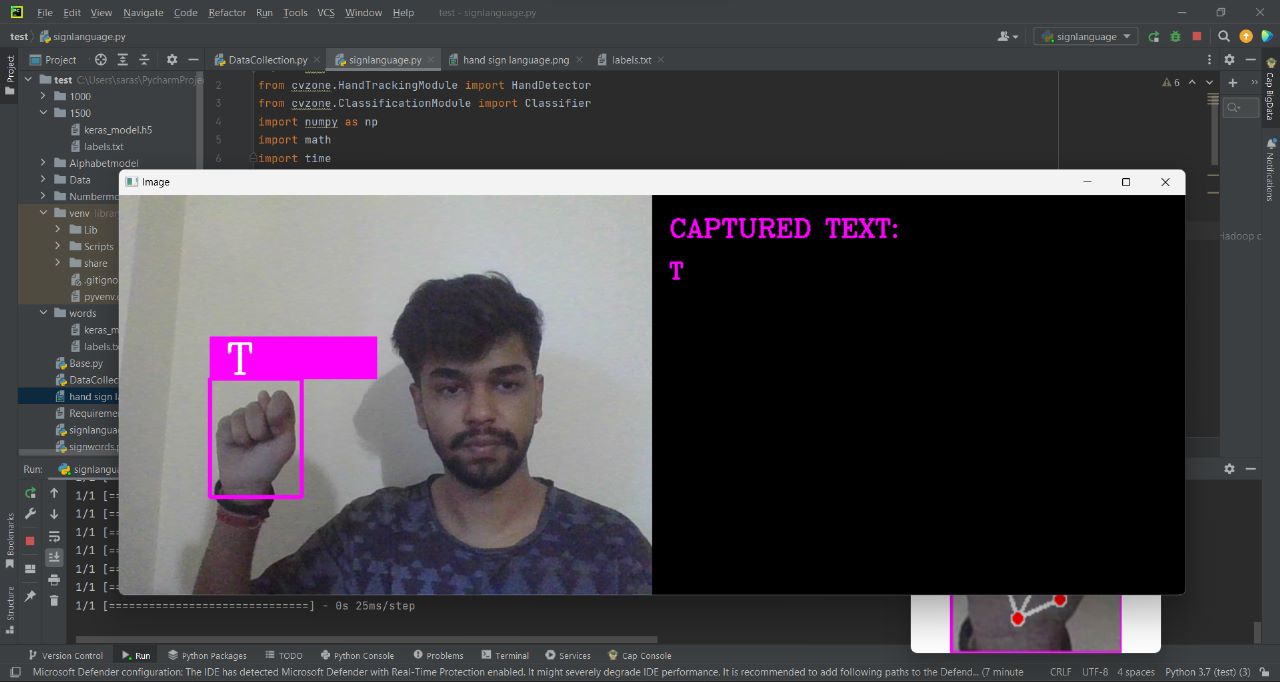
**Fig 12. The interpreter detecting the gesture of ‘Q’ and giving out the estimated word.**

**O**



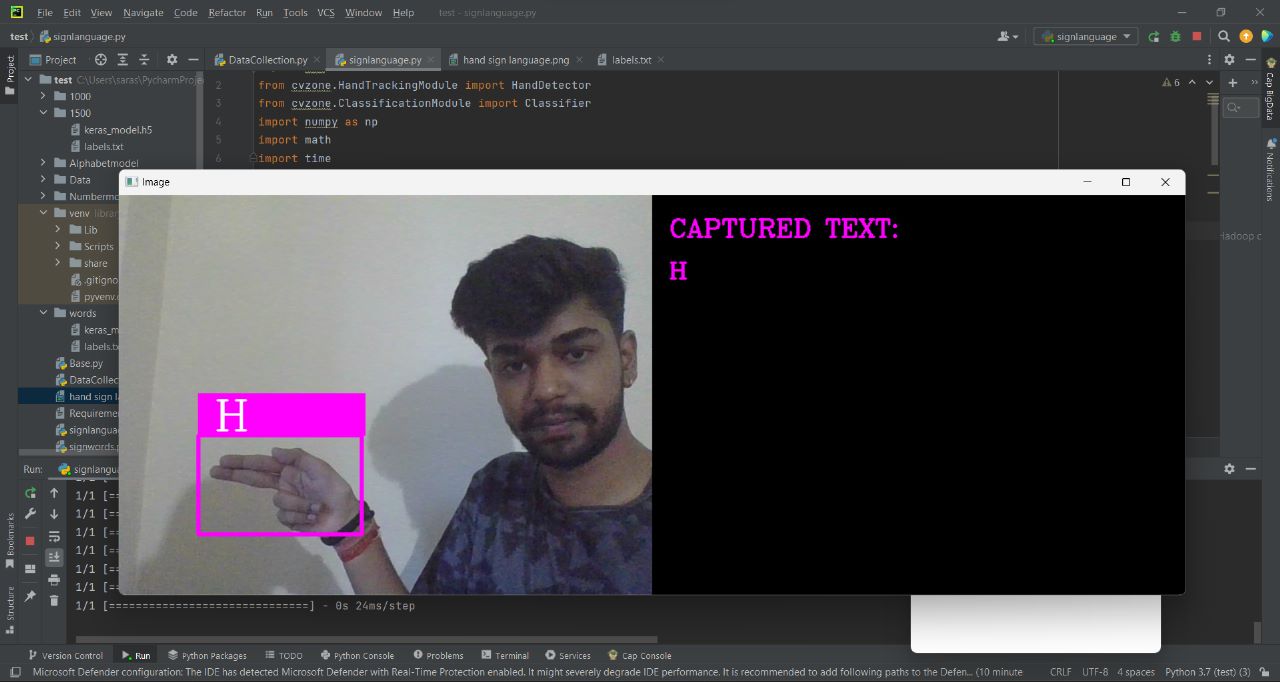
**Fig 13. The interpreter detecting the gesture of ‘O’ and giving out the estimated word.**

**T**



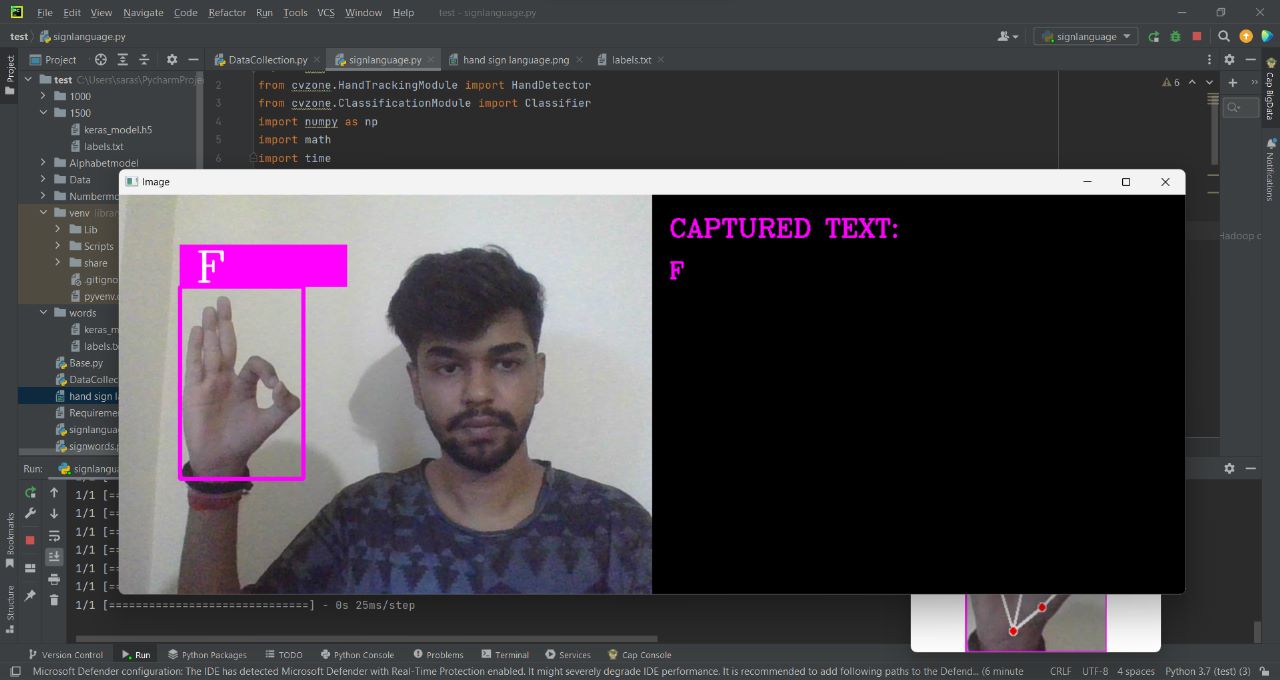
**Fig 14. The interpreter detecting the gesture of ‘T’ and giving out the estimated word.**

**H**



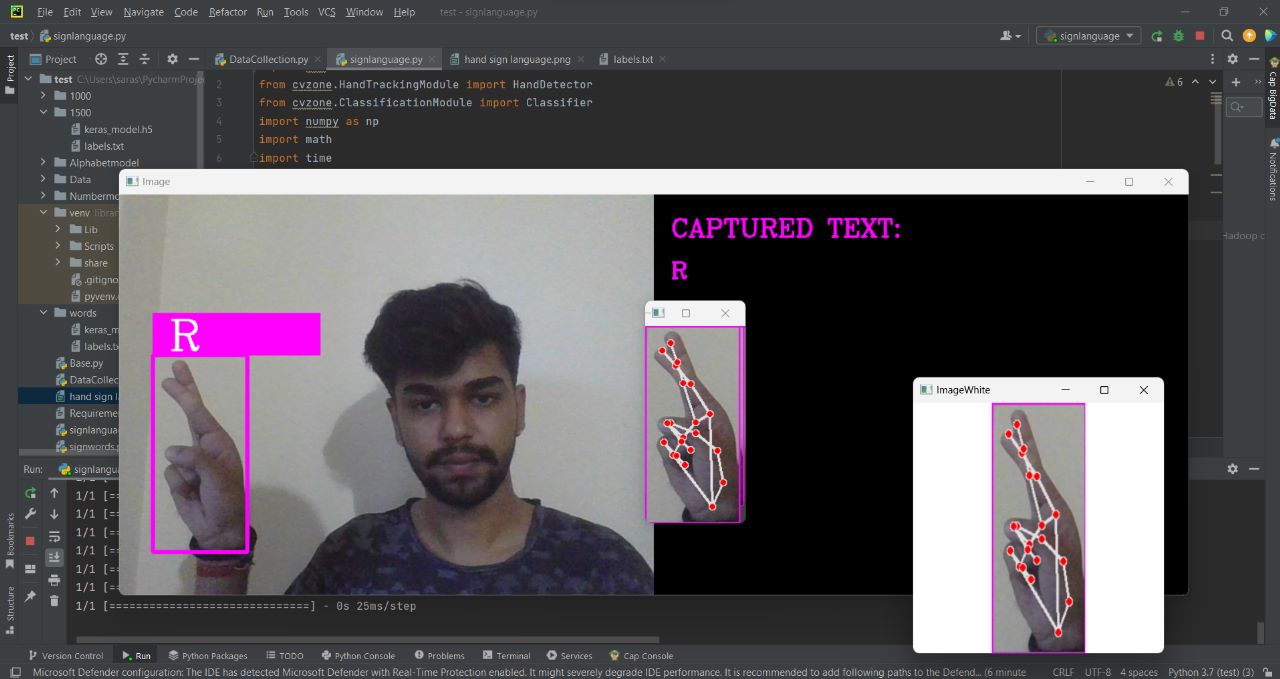
**Fig 15. The interpreter detecting the gesture of ‘H’ and giving out the estimated word.**

**F**



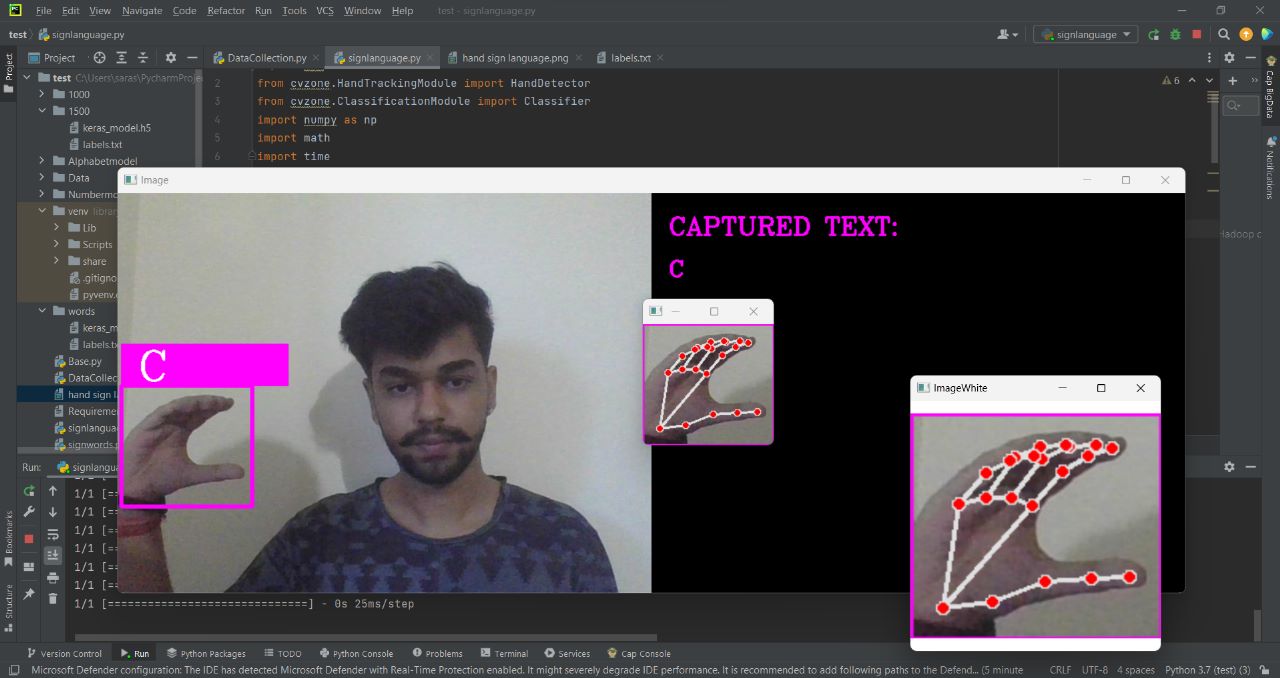
**Fig 16. The interpreter detecting the gesture of ‘F’ and giving out the estimated word.**

**R**



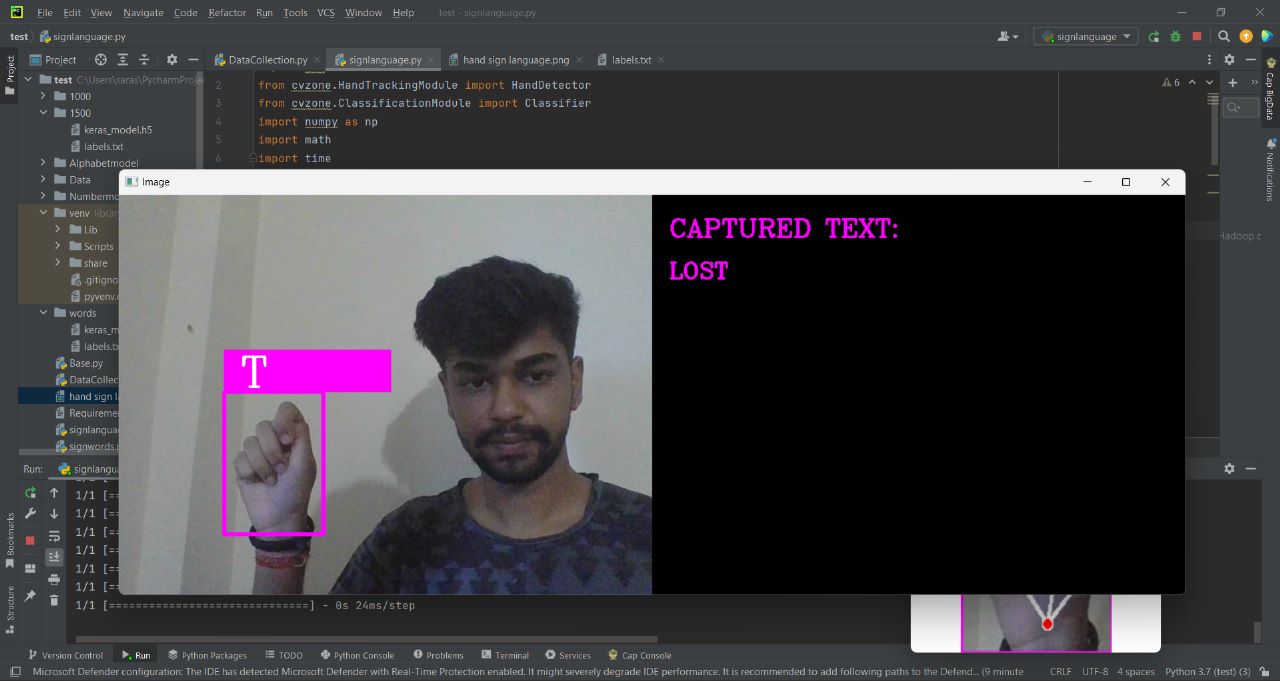
**Fig 17. The interpreter detecting the gesture of ‘R’ and giving out the estimated word.**

**C**



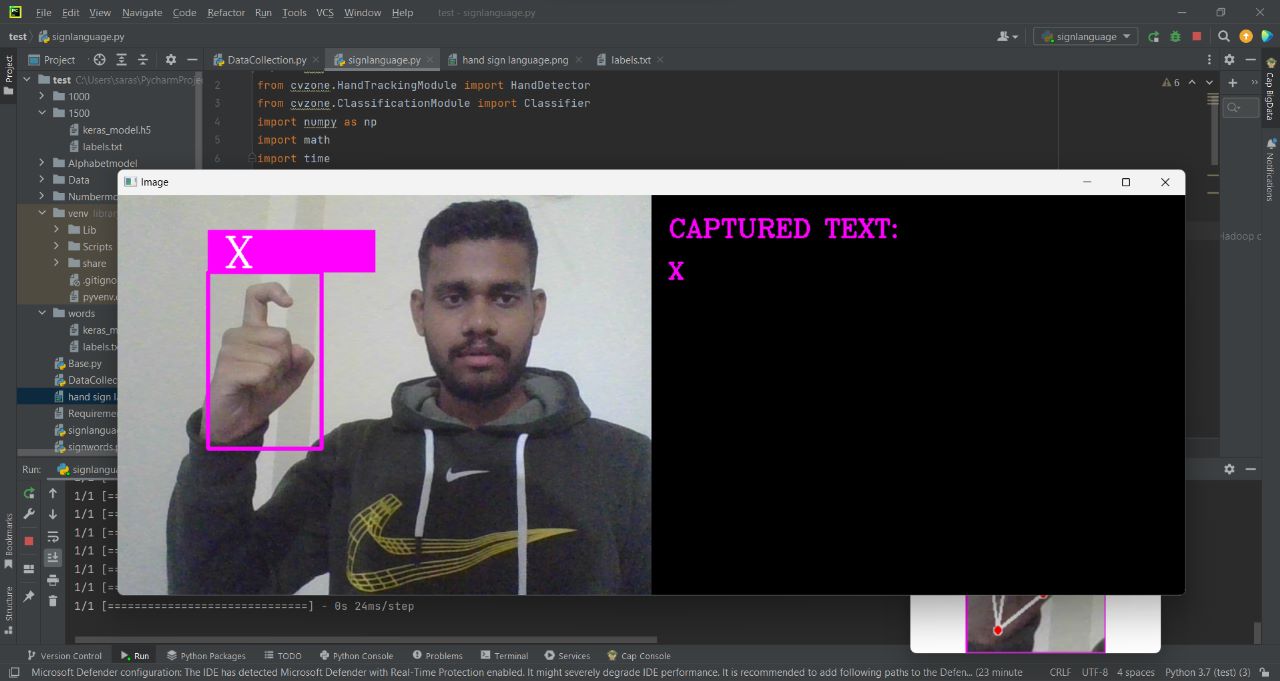
**Fig 18. The interpreter detecting the gesture of ‘C’ and giving out the estimated word.**

**MAKING THE WORD ‘LOST’ WITH THE HELP OF THE INTERRUPTER**



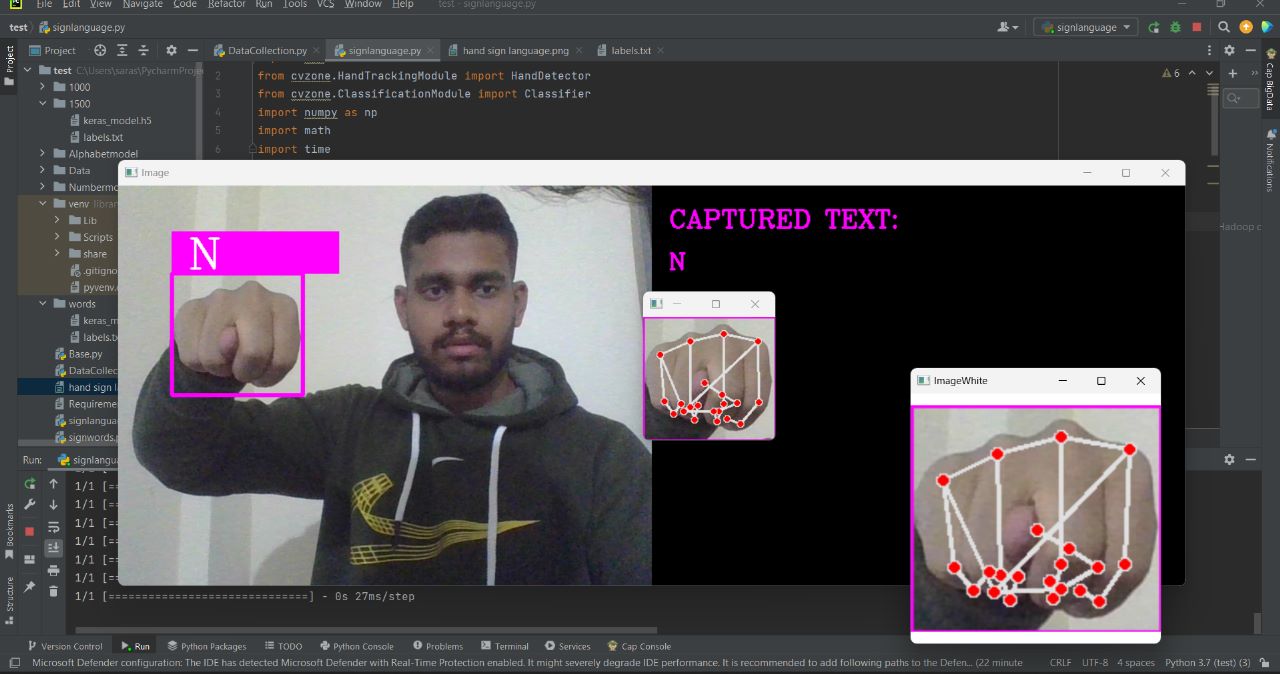
**Fig 19. The interpreter combining the letters of ‘L’ ‘O’ ‘S’ ‘T’ and giving out the word ‘LOST’**

**X**



**Fig 20. The interpreter detecting the gesture of ‘X’ and giving out the estimated word.**

**N**



**Fig 21. The interpreter detecting the gesture of ‘N’ and giving out the estimated word.**

**7. CONCLUSION AND SCOPE FOR FUTURE ENCHANCEMENT**

The proposed project is a significant undertaking, but it has the potential to make a real difference in the lives of deaf people. The system would provide a valuable tool for communication and education, and it would help to break down barriers between deaf and hearing people.

The project would require a team of skilled and dedicated developers and researchers. The team would need to be able to work together effectively and to overcome the challenges that are inherent in developing a complex system like this one.

The project would also require funding. The cost of developing and deploying the system would be significant, but the potential benefits of the system would be worth the investment.

If the project is successful, it could have a profound impact on the lives of deaf people. The system could help to improve communication, access to education and employment, and quality of life for deaf people.

The project would also be a valuable contribution to the field of sign language recognition. The system would be used to gather data and to develop new techniques for sign language recognition. This would help to advance the field and to make sign language recognition more accurate and reliable.

The proposed project is a challenging but worthwhile undertaking. The system would have a positive impact on the lives of deaf people and would help to advance the field of sign language recognition.

Here are some additional thoughts on the conclusion:

* The project would be a valuable contribution to the field of sign language recognition. The system would be used to gather data and to develop new techniques for sign language recognition. This would help to advance the field and to make sign language recognition more accurate and reliable.
* The project would also be a valuable tool for education. The system could be used to teach sign language to hearing people, and it could also be used to provide deaf people with access to educational resources.
* The project would help to break down barriers between deaf and hearing people. The system would provide a way for deaf and hearing people to communicate with each other, and it would help to create a more inclusive society.

**SCOPE FOR FUTURE ENCHANCEMENT**

The proposed project has the potential to be enhanced in a number of ways. Here are a few ideas:

* Improve accuracy: The accuracy of the system could be improved by using a larger dataset of images or videos of sign gestures. The dataset could also be more diverse, to include sign gestures from different sign languages and from different signers.
* Reduce latency: The latency of the system could be reduced by using a faster image processing algorithm. The system could also be deployed on a cloud computing platform, which would allow it to scale to handle more users and to process images or videos more quickly.
* Increase robustness: The robustness of the system could be increased by using a more robust machine learning algorithm. The algorithm could also be trained on a dataset of images or videos that includes sign gestures that are performed in different ways.
* Extend to other languages: The system could be extended to recognize sign gestures from other sign languages. This would require a larger dataset of images or videos of sign gestures from different sign languages.
* Add new features: The system could be enhanced with new features, such as the ability to translate sign language into text or speech. The system could also be used to provide feedback to users on their sign language skills.

The future enhancement of the proposed project is limited only by the imagination of the developers and researchers. The system has the potential to be a powerful tool for communication and education, and it could be used to make a real difference in the lives of deaf people.

Here are some additional thoughts on the scope for future enhancement:

* The system could be enhanced to recognize sign language gestures in real time. This would allow the system to be used in applications such as real-time translation and communication with deaf people.
* The system could be enhanced to recognize sign language gestures from different angles. This would allow the system to be used in applications such as video conferencing and distance learning.
* The system could be enhanced to recognize sign language gestures from different distances. This would allow the system to be used in applications such as public speaking and classroom instruction.

The future enhancement of the proposed project is a promising area of research. The system has the potential to make a real difference in the lives of deaf people, and it could be used to create a more inclusive society.

**8. REFRENCES**

[1] Sridhar, P., & Gowda, S. M. (2020). A Survey on Hand Gesture Recognition Techniques for Sign Language. International Journal of Innovative Technology and Exploring Engineering, 9(2), 2835-2843

[2] Huang, Y., Cui, Y., & Huang, T. (2018). Deep learning for sign language recognition and translation: A survey. Pattern Recognition, 80, 109-126

[3] Cruz-Sanchez, H., Starostenko, O., & Nakasone, A. (2020). Sign language recognition using computer vision: A review. Expert Systems with Applications, 160, 113690

[4] Tripathy, R. K., & Patnaik, S. (2019). Sign language recognition systems: A comprehensive review. IEEE Access, 7, 68910-68925

**8. BIBILOGRAPHY**

[www.google.com](http://www.google.com)

[www.googleteachablemachine.com](http://www.googleteachablemachine.com)

[www.stackoverflow.com](http://www.stackoverflow.com)

[www.github.com](http://www.github.com)

[www.python.org](http://www.python.org)

[www.tensorflow.org](http://www.tensorflow.org)

[www.keras.com](http://www.keras.com)