A

Mini-Project Report on

Sign Language Interpreter using Google Teachable Machine

Submitted in partial fulfillment of the requirements
for the degree of
BACHELOR OF ENGINEERING
IN
Computer Science & Engineering
Artificial Intelligence & Machine Learning

by

Adrian Gilbert T. (21106015)
Sanjita Shukla (21106056)
Chirag Sawant (21106058)
Omkar Yadav (21106023)

Under the guidance of

Dr. Jaya Gupta

Department of Computer Science & Engineering
(Artificial Intelligence & Machine Learning)
A. P. Shah Institute of Technology
G. B. Road, Kasarvadavali, Thane (W)-400615
University Of Mumbai

2022-2023

A. P. SHAH INSTITUTE OF TECHNOLOGY

CERTIFICATE

This is to certify that the project entitled "**Sign Language Interpreter using Google Teachable Machine"** is a bonafide work of Adrian Gilbert T. (21106015), Sanjita Shukla (21106056), Chirag Sawant (21106058), Omkar Yadav (21106023) submitted to the University of Mumbai in partial fulfillment of the requirement for the award of **Bachelor of Engineering** in **Computer Science & Engineering (Artificial Intelligence & Machine Learning).**

Dr. Jaya Gupta

Head of Department and Mini Project Guide

A. P. SHAH INSTITUTE OF TECHNOLOGY

Project Report Approval

This Mini project report entitled "**Sign Language Interpreter using Google Teachable Machine**" by **Adrian Gilbert T, Sanjita Shukla, Chirag Sawant and Omkar Yadav** is approved for the degree of ***Bachelor of Engineering*** in ***Computer Science &Engineering***, (AIML) ***2022-23***.

External Examiner: ________________

Internal Examiner: _______________

Place: APSIT, Thane
Date:

Declaration

We declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission hasnot been taken when needed.

Adrian Gilbert T.	Sanjita Shukla	Chirag Sawant	Omkar Yadav
(21106015)	(21106056)	(21106058)	(21106023)

ABSTRACT

Simply put, communication is the act of transferring information from one location, person, or group to another. Every communication has at least one sender, one message, and one recipient. This may appear straightforward, but communication is a very complex subject. The transmission of a message from sender to recipient can be influenced by a wide range of factors, including our emotions, the cultural situation, the communication medium used and even our location. Because of the complexities, good communication skills are highly valued by employers all over the world. According to the World Health Organization (WHO), hearing loss affects approximately 466 million people worldwide, with 34 million of these people being children. Sign language is one of the options available to these people as a mode of communication. American Sign Language (ASL) is the fourth most studied second language at American universities and one of the most widely used languages in the United States. ASL is primarily used by people who are deaf or hard of hearing in North America. In the United States and Canada, there are between 250,000 and 500,000 ASL users, the majority of whom use ASL as their primary language. Knowing ASL allows us to communicate with a diverse range of hearing, hard of hearing, and deaf people, including students in mainstream and deaf school or university programs, as well as deaf or hard of hearing residents and business people in our community. To address this situation, we present a project that will benefit such people. This project proposes a solution based on Machine Learning that will recognize hand gestures and translate them into text. A webcam will be used to capture the region of interest, i.e., recognize the indicated hand motions. The translated text will thus be provided based on the recognized gestures.

Index

CHAPTER 1

INTRODUCTION

1. INTRODUCTION

Communication is the process of exchanging information, views and expressions between two or more persons, in both verbal and non-verbal manner. Hand gestures are the non-verbal method of communication used along with verbal communication. A more organized form of hand gesture communication is known as sign language. In this language each alphabet of the English vocabulary is assigned a sign. A physically disabled person like the deaf and the dumb use this language to communicate with each other. The idea of this project is to design a system that can understand the sign language accurately so that the less fortunate people may communicate with the outside world without the need of an interpreter. Sign language is a more organized and defined way of communication in which every word or alphabet is assigned some gesture. In American Sign Language (ASL) each alphabet of English vocabulary, A-Z, is assigned a unique gesture. Sign language is mostly used by the deaf, dumb or people with any other kind of disabilities. Our aim is to design a Human Computer Interface (HCI) system that can understand sign language accurately so that the signing people may communicate with the non-signing people without the need of an interpreter. No one form of sign language is universal as it varies from region to region and country to country and a single gesture can carry a different meaning in a different part of the world. Various available sign languages are American Sign Language (ASL), British Sign Language (BSL), Turkish Sign Language (TSL), Indian Sign Language (ISL) and many more. There are a total of 26 alphabets in the English vocabulary. Each alphabet may be assigned a unique gesture. In our project, the image of the hand is captured using a simple web camera. The acquired image is then processed and some features are extracted. These features are then used as input to a classification algorithm for recognition. The recognized gesture may then be used to generate speech or text. This project aims at designing a fully functional system with significant improvement from the past works.

CHAPTER 2

LITERATURE SURVEY

2. LITERATURE SURVEY

2.1-HISTORY

Prior to the birth of ASL, sign language had been used by various communities in the United States. In the United States, as elsewhere in the world, hearing families with deaf children have historically employed ad hoc home sign, which often reaches much higher levels of sophistication than gestures used by hearing people in spoken conversation. As early as 1541 at first contact by Francisco Vásquez de Coronado, there were reports that the Indigenous peoples of the Great Plains widely spoke a sign language to communicate across vast national and linguistic lines.

Due to intermarriage in the original community of English settlers of the 1690s, and the recessive nature of genetic deafness, Chilmark had a high 4% rate of genetic deafness. MVSL was used even by hearing residents whenever a deaf person was present, and also in some situations where spoken language would be ineffective or inappropriate, such as during church sermons or between boats at sea.

ASL is thought to have originated in the American School for the Deaf (ASD), founded in Hartford, Connecticut, in 1817. Originally known as The American Asylum, At Hartford, For The Education And Instruction Of The Deaf And Dumb, the school was founded by the Yale graduate and divinity student Thomas Hopkins Gallaudet. Sign production can often vary according to location. Signers from the South tend to sign with more flow and ease. Native signers from New York have been reported as signing comparatively quicker and sharper. Sign production of native Californian signers has also been reported as being fast. Research on that phenomenon often concludes that the fast-paced production for signers from the coasts could be due to the fast-paced nature of living in large metropolitan areas. That conclusion also supports how the ease with which Southern sign could be caused by the easygoing environment of the South in comparison to that of the coasts. Sign production can also vary depending on age and native language. For example, sign production of letters may vary in older signers. Slight differences in finger spelling production can be a signal of age. Additionally, signers who learned American Sign Language as a second language vary in production. For Deaf signers who learned a different sign language before learning American Sign Language, qualities of their native language may show in their ASL production. Some examples of that varied production include fingerspelling towards the body, instead of away from it, and signing certain movement from bottom to top, instead of top to bottom. Hearing people who learn American Sign Language also have noticeable differences in signing production. The most notable production difference of hearing people learning American Sign Language is their rhythm and arm posture.

2.2-LITERATURE REVIEW

[1] Sign Language Recognition and Translation: A Multidisciplined Approach from the Field of Artificial Intelligence, Becky Sue Parton.

This paper reviews significant projects in the field beginning with finger-spelling hands such as "Ralph" (robotics), CyberGloves (virtual reality sensors to capture isolated and continuous signs), camera-based projects such as the CopyCat interactive American Sign Language game (computer vision), and sign recognition software (Hidden Markov Modeling and neural network systems). Avatars such as "Tessa" (Text and Sign Support Assistant; three-dimensional imaging) and spoken language to sign language translation systems such as Poland's project entitled "THETOS" (Text into Sign Language Automatic Translator, which operates in Polish; natural language processing) are addressed. The application of this research to education is also explored. The "ICICLE" (Interactive Computer Identification and Correction of Language Errors) project, for example, uses intelligent computer-aided instruction to build a tutorial system for deaf or hard-of-hearing children that analyzes their English writing and makes tailored lessons and recommendations. Finally, the article considers synthesized sign, which is being added to educational material and has the potential to be developed by students themselves.

[2] Sign language recognition using image based hand gesture recognition techniques, Ashish S. Nikam, Aarti G. Ambekar.

This paper introduced software which presents a system prototype that is able to automatically recognize sign language to help deaf and dumb people to communicate more effectively with each other or normal people. Pattern recognition and Gesture recognition are the developing fields of research. Being a significant part in nonverbal communication hand gestures are playing key role in our daily life. Hand Gesture recognition system provides us an innovative, natural, user friendly way of communication with the computer which is more familiar to the human beings. By considering in mind the similarities of human hand shape with four fingers and one thumb, the software aims to present a real time system for recognition of hand gesture on basis of detection of some shape-based features like orientation, Centre of mass centroid, fingers status, thumb in positions of raised or folded fingers of hand.

[3] ML Based Sign Language Recognition System.

This paper reviews different steps in an automated sign language recognition (SLR) system. Developing a system that can read and interpret a sign must be trained using a large dataset and the best algorithm. As a basic SLR system, an isolated recognition model is developed. The model is based on vision-based isolated hand gesture detection and recognition. Assessment of ML-based SLR model was conducted with the help of 4 candidates under a controlled environment. The model made use of a convex hull for feature extraction and KNN for classification. The model yielded 65% accuracy.

[4] Deep learning-based sign language recognition system for static signs.

This paper deals with robust modeling of static signs in the context of sign language recognition using deep learning-based convolutional neural networks (CNN). In this research, total 35,000 sign images of 100 static signs are collected from different users. The efficiency of the proposed system is evaluated on approximately 50 CNN models. The results are also evaluated on the basis of different optimizers, and it has been observed that the proposed approach has achieved the highest training accuracy of 99.72% and 99.90% on colored and grayscale images, respectively. The performance of the proposed system has also been evaluated on the basis of precision, recall and *F*-score. The system also demonstrates its effectiveness over the earlier works in which only a few hand signs are considered for recognition.

[5] American Sign Language: Facts & Related Content, Erik Drasgow.

American Sign Language (ASL), visual-gestural language used by most of the deaf community in the United States and Canada. ASL is a natural language with a structure quite different from spoken English. It is not a manual-gestural representation of spoken English, nor is it pantomime. Instead, ASL is a full language, with all of the properties of spoken natural languages, but one that has developed independently of and differently from English. The role of ASL in the education of deaf students has been characterized by conflict and controversy. This situation has existed throughout the history of deaf education in the United States.

[6] Sign Language Interpreter Using Machine Learning, S. Anthoniraj, Vn Ganashree, Gangineni Divya Sai, B R Navya.

Communication with the rest of the world is very difficult for hearing-impaired people and hence they are isolated, in order to overcome these circumstances, we present a project that is helpful to such people. This project presents a solution with the use of Machine Learning that will identify hand gestures and translate it into speech/text and the vice versa that is it convert from speech/text to gesture. Communication is the key point for interaction and we humans use different languages through which we share our views and opinions. But for certain people i. e., the hearing impaired or the deaf and dumb they are not able communicate well because of their disability. According to World Health Organization (WHO) it is estimated that hearing loss affects approximately four hundred sixty-six million people globally with thirty-four million of these being children. A webcam will be used to capture the region of interest i.e., recognize the hand motion as well as the gesture that the hand is indicating. Based on the recognized gestures, the recorded soundtrack will be played.

[7] Sign Language Recognition Systems: A Decade Systematic Literature Review, Ankita Wadhawan & Parteek Kumar.

This is the first identifiable academic literature review of sign language recognition systems. It provides an academic database of literature between the duration of 2007–2017 and proposes a classification scheme to classify the research articles. Three hundred and ninety six research articles were identified and reviewed for their direct relevance to sign language recognition systems. One hundred and seventeen research articles were subsequently selected, reviewed and classified. Each of 117 selected papers was categorized on the basis of twenty five sign languages and were further compared on the basis of six dimensions (data acquisition techniques, static/dynamic signs, signing mode, single/double handed signs, classification technique and recognition rate). The Systematic Literature Review and classification process was verified independently.

CHAPTER 3

PROBLEM STATEMENT

3. PROBLEM STATEMENT

It is vital in our contemporary society to socialise with all people, whether for enjoyment or for some other reason. Every human being needs communication. People with hearing and/or speech disabilities, on the other hand, require a different mode of communication than vocal communication. They converse with each other via sign language. Dumb individuals communicate via hand signs, therefore normal people have difficulty recognising their language through signs. As a result, technologies that recognise various signs and deliver the respective information tied to these signs to ordinary people are crucially needed. Sign language is difficult to understand and learn, and not everyone understands what the sign language movements represent. It additionally demands time to learn sign language because there is no reliable, portable tool for recognising sign language. Hearing or speech disabled people who know Sign Language require a translator who also knows Sign Language to effectively communicate their thoughts to others. Existing sign language recognition systems confront a number of obstacles, including low accuracy rates, a restricted vocabulary, and difficulty recognising signs produced by diverse people with different signing styles and speeds. Furthermore, many sign language recognition systems are not portable and necessitate costly hardware installations, limiting their accessibility and usability. As a result, the goal of this project is to overcome these issues by developing an accurate, robust, and portable sign language recognition system that can recognise sign language movements performed by people with hearing loss in real-time. With a high recognition rate, the system will use powerful computer vision and machine learning algorithms to accurately read and translate sign language motions into text or speech. The successful completion of this project will contribute to the improvement of communication and inclusion of people with hearing loss, allowing them to speak more effectively with the hearing population and engage fully in a variety of social and professional contexts. It will also have potential uses in sectors such as education, healthcare, and accessibility, positively impacting the lives of people with disabilities. It will also have potential uses in sectors such as education, healthcare, and accessibility, improving the lives of people with hearing loss and increasing inclusion in society.

CHAPTER 4

EXPERIMENTAL SETUP

4.1 Hardware Setup

1) Processor:

As for the CPU, a multi-core CPU, such as Intel Core i5 or higher (or equivalent from other manufacturers), would be sufficient for most small to medium-sized sign language detection projects. The specific CPU requirements may vary depending on the size of the dataset and complexity of the model, but a modern CPU with good processing power and multiple cores should be capable of handling the training workload. The system we used was a laptop with an AMD Ryzen 6800h processor. It's important to note that larger datasets and more complex models may require more powerful CPUs to ensure efficient and timely training. If you have a particularly large dataset or plan to train a complex model, you may need to consider a more powerful CPU, such as an Intel Core i7 or higher, or even a dedicated workstation with high-end CPUs like Intel Xeon or AMD Threadripper processors.

2) Ram:

As a general guideline, a minimum of 4 GB of RAM would be sufficient for smaller datasets and simpler models. However, for larger datasets and more complex models, the system may demand more RAM to ensure smooth and efficient training. When training machine learning models, the dataset is loaded into memory during the training process, and the model's weights and parameters are also stored in the memory. Larger datasets and more complex models require more memory to store and process the data efficiently. If the dataset is particularly large or if there may be plans to train deep learning models with many layers or parameters, then 16 GB, 32 GB, or even higher amounts of RAM may be needed for optimal performance. For our system,

3) Graphics Processing Unit (GPU):

While not strictly necessary, having access to a GPU can significantly speed up the training process, especially for complex models. Google Teachable Machine supports training with GPUs, and a dedicated GPU, such as NVIDIA GeForce or AMD Radeon, with at least 4 GB of VRAM would be beneficial for faster model training. For our system,

4) Storage:

There will always be a need for adequate storage space to store the dataset, trained model files, and any other necessary files. The size of the dataset and model files will depend on the number of sign language gestures, resolution and duration of the images, and model architecture used. A minimum of 100 GB of free storage space is recommended, though the actual requirement will vary depending on the specific use case. For our system,

5) High-speed Internet Connection:

Google Teachable Machine is a web-based platform, so a fast and stable internet connection for uploading and downloading datasets, model files, and for using the platform smoothly will be needed.

6) Webcam:

A webcam is an essential hardware component for a sign language detection system using Google Teachable Machine. Considering factors such as image quality, compatibility, connection interface, mounting options, and budget when choosing a webcam for your sign language detection system is crucial. For our system,

Display Monitor: A display monitor with a resolution of 1920x1080 or higher is recommended for a comfortable visual interface when interacting with the sign language detection system.

4.2 Software Setup

1) Linux/Windows OS:

Google Teachable Machine is a web-based platform, so it can be accessed through a web browser on multiple operating systems, including Windows, macOS, and Linux. Therefore, any modern computer running one of these operating systems would be sufficient to use Google Teachable Machine for creating a sign language detection system.

2) Appropriate Python Interpreter:

As for the Python interpreter, you can use the latest stable version of Python, which is currently Python 3.9 or higher. Additionally, certain Python libraries are also needed to be installed which will be mentioned ahead.

3) OpenCV Python:

OpenCV is a Python library that allows you to perform image processing and computer vision tasks. It provides a wide range of features, including object detection, face recognition, and tracking.

4) CV zone:

Computer Vision Zone is a one stop computer vision platform that provides premium resources for learning computer vision techniques.

5) NumPy:

NumPy offers comprehensive mathematical functions, random number generators, linear algebra routines, Fourier transforms, and more

6) TensorFlow:

TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks.

7) Keras:

Keras is an open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library. Up until version 2.3, Keras supported multiple backends, including TensorFlow, Microsoft Cognitive Toolkit, Theano and PlaidML.

8) Pillow:

Python Imaging Library is a free and open-source additional library for the Python programming language that adds support for opening, manipulating, and saving many different image-file formats. It is available for Windows, Mac OS X and Linux.

9) Google Teachable Machine:

Teachable Machine is a web-based tool that makes creating machine learning models fast, easy, and accessible to everyone.

CHAPTER 5
BLOCK DIAGRAM

5.1 Block diagram of proposed system:

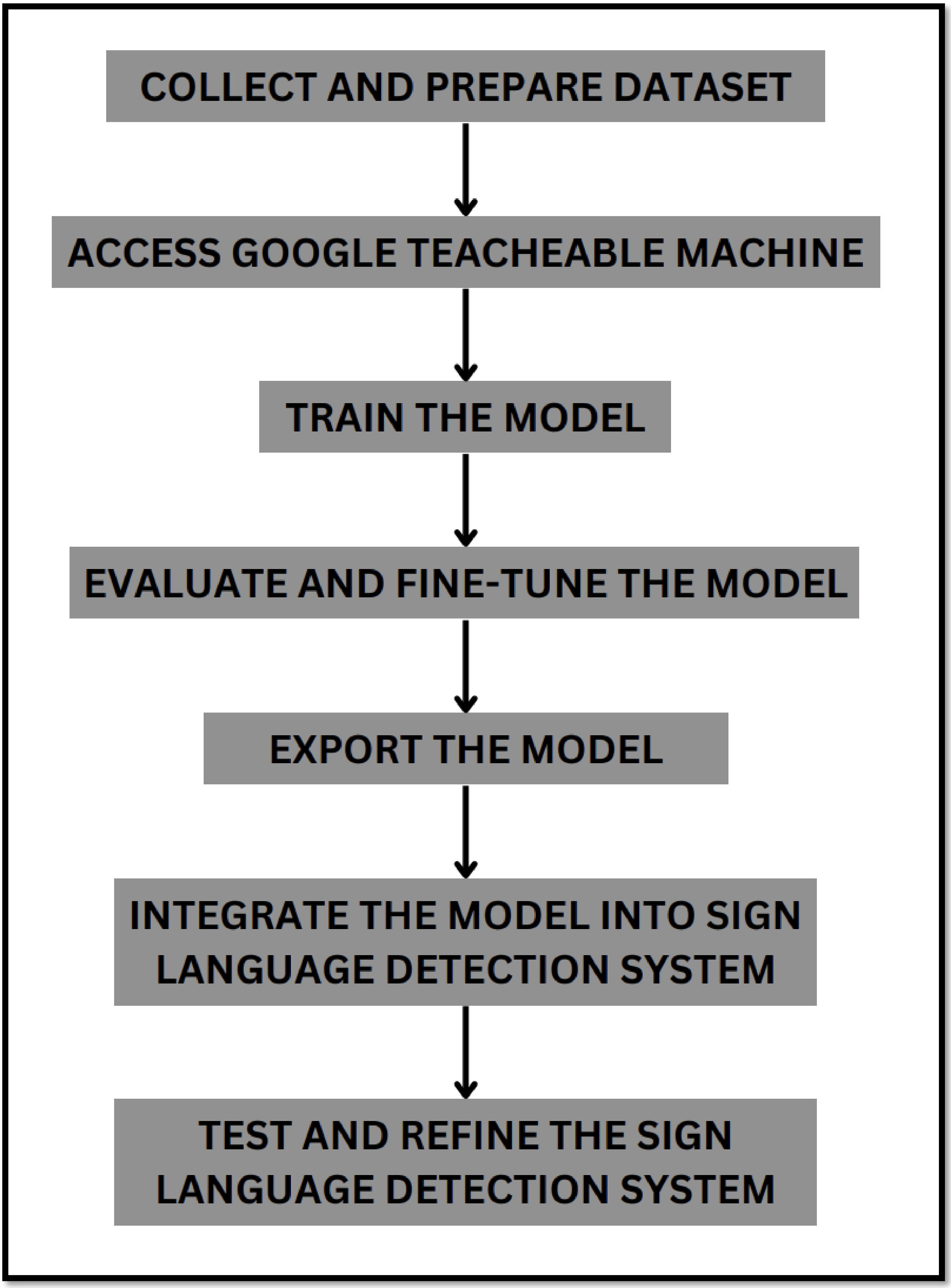

Fig 1.1 Block Diagram

5.2 Description of Block Diagram:

Step 1: Collect and Prepare Dataset:

Gather a dataset of sign language gestures, which includes images of different sign gestures performed by various sign language users. Label the dataset, assigning appropriate class labels to each sign gesture to indicate the corresponding sign being performed. Split the dataset into training, validation, and testing sets for model training and evaluation.

Step 2: Access Google Teachable Machine:

Go to the Teachable Machine website (https://teachablemachine.withgoogle.com/) and click on the "Get Started" button to access the platform. Choose the "Image Project" or "Audio Project" based on the type of data you have collected (images or videos for sign language gestures).

Upload your dataset to Teachable Machine, assigning the class labels to the corresponding sign language gestures.

Step 3: Train the Model:

Use the Teachable Machine platform to train your sign language detection model using the uploaded dataset. Experiment with different model architectures and hyperparameters to optimize model performance. Monitor the training process and iterate as needed to achieve satisfactory model accuracy.

Step 4: Evaluate and Fine-tune the Model:

Evaluate the trained model using the validation and testing sets to assess its accuracy and generalization performance. Fine-tune the model by adjusting model parameters or collecting more data, if necessary, to improve model accuracy.

Step 5: Export the Model:

Once satisfied with the model's performance, export the trained model from Teachable Machine in a format suitable for your target platform (e.g., TensorFlow, TensorFlow.js, or a custom format). Follow the instructions provided by Teachable Machine to export and download the model files.

Step 6: Integrate the Model into the Sign Language Detection System:

Incorporate the exported model into your sign language detection system, using the appropriate libraries or frameworks for your target platform (e.g., TensorFlow.js for web-based applications or TensorFlow for Python-based applications). Implement the necessary preprocessing steps, such as hand detection and tracking, in your system to prepare input data for the model. Utilize the exported model to perform real-time sign language gesture recognition on the captured video or image data in your system.

Step 7: Test and Refine the Sign Language Detection System:

Test the sign language detection system with real-world sign language gestures to assess its accuracy and performance.Refine the system as needed based on user feedback and evaluation results, such as improving hand detection, tracking, or gesture recognition accuracy.

CHAPTER 6

Proposed System & Implementation

6.1 Flowchart

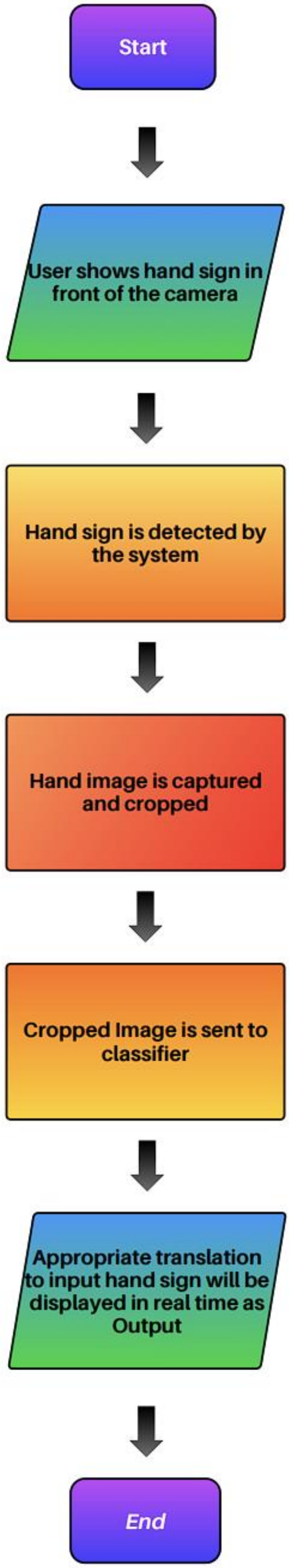

Fig 2.1 Flowchart

6.2 Description of Flowchart

Step 1: (Setup and Calibration)

Calibrate the system according to the environment, this may involve adjusting camera settings, hand positioning or lighting conditions to optimize system performance.

Step 2: (Capture Sign Language Gestures)

Position the camera in front of the user who is signing, ensuring that the hands are clearly visible within the camera's field of view. Begin capturing sign language gestures by starting the video stream in the sign language detection system software.

Step 3: (Hand Detection and Tracking)

The system uses computer vision techniques to detect and track the user's hands in real-time from the video stream. The system employs methods such as hand position cropping, hand shape analysis and deep learning-based object detection to accurately locate and track the user's hands.

Step 4: (Hand Gesture Recognition)

Once the hands are detected and tracked, the system applies machine learning algorithms to recognize the sign language gestures being performed by the user. The system compares the detected hand gestures against a pre-trained dataset of sign language gestures to determine the most likely sign being performed. The system uses techniques such as feature extraction, pattern recognition, or neural network-based classification to achieve accurate sign language gesture recognition.

Step 5: (Output and Translation)

The system generates output based on the recognized sign language gestures, which includes real-time text transcription. The output is displayed to the user or transmitted to a recipient, depending on the intended use of the system.

That's it! Following these steps, users can interact with the sign language detection system to communicate using sign language, enabling effective communication for individuals who are deaf or hard of hearing.

6.3 Implementation

System output screen displaying translation to the input images as number 5:

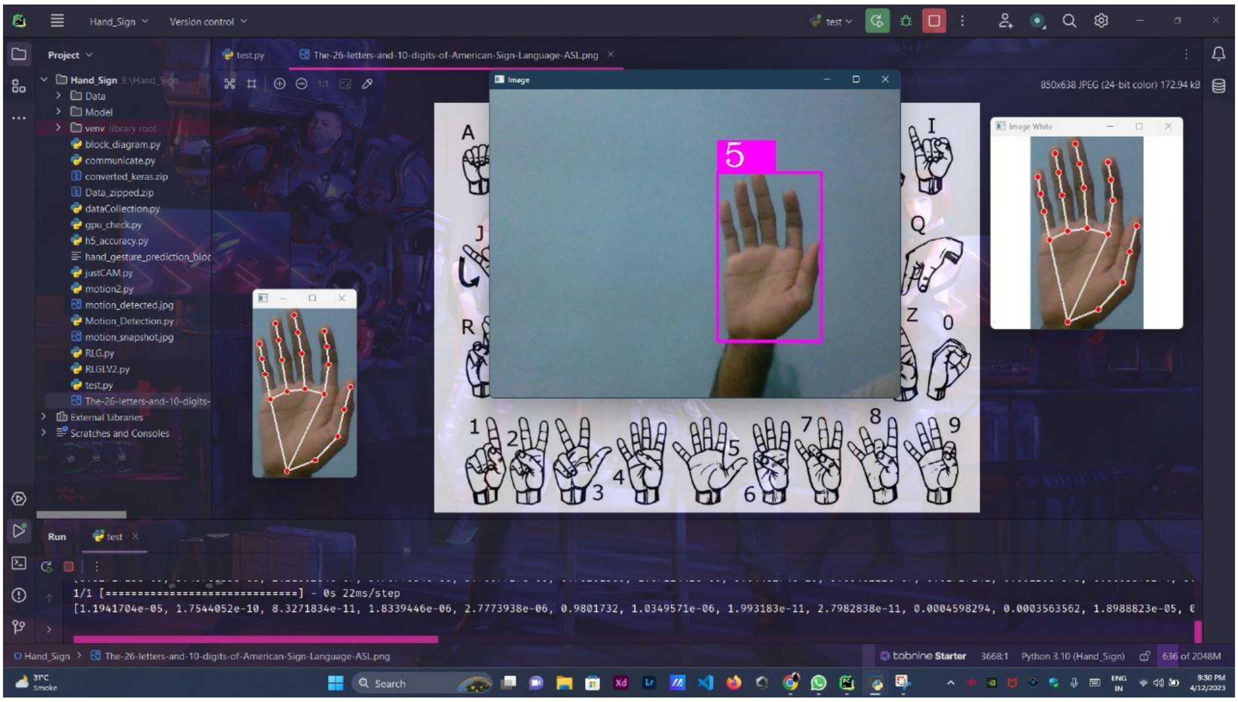

Fig 3.1 Output 5

System output screen displaying translation to the input image as number 2:

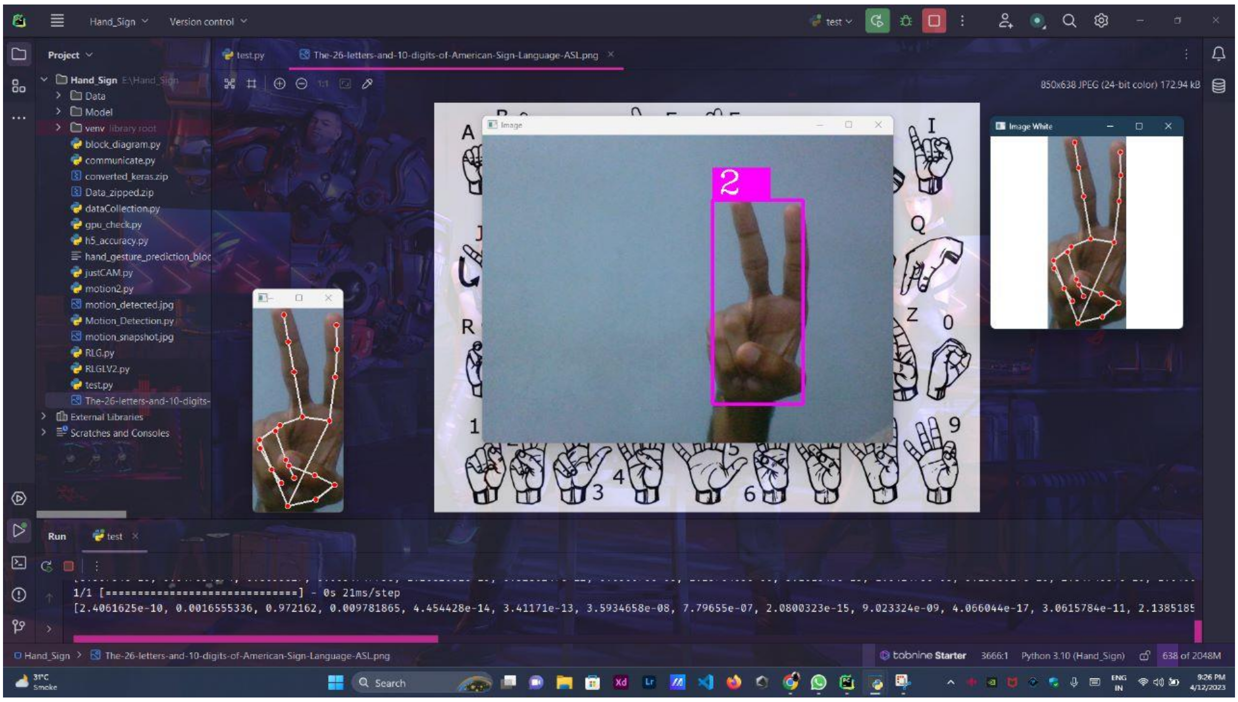

Fig 3.2 Output 2

System output screen displaying translation to the input image as alphabet C:

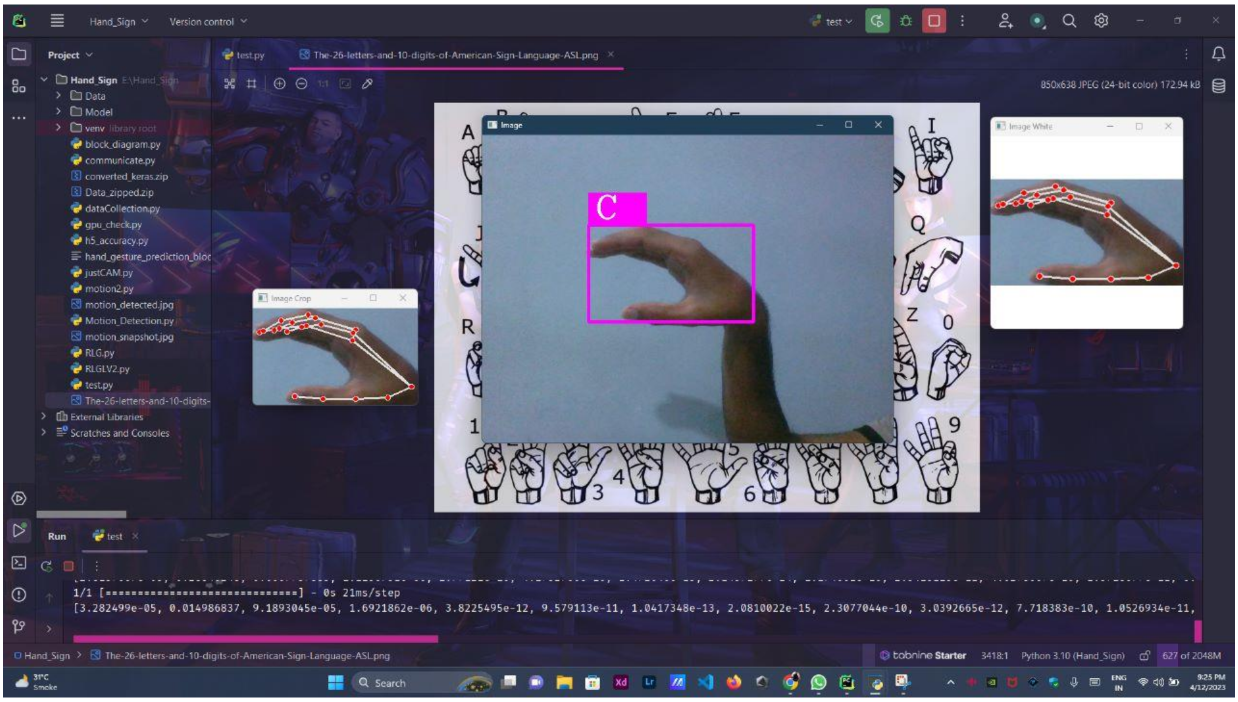

Fig 3.3 Output C

System output screen displaying translation to the input image as alphabet Y:

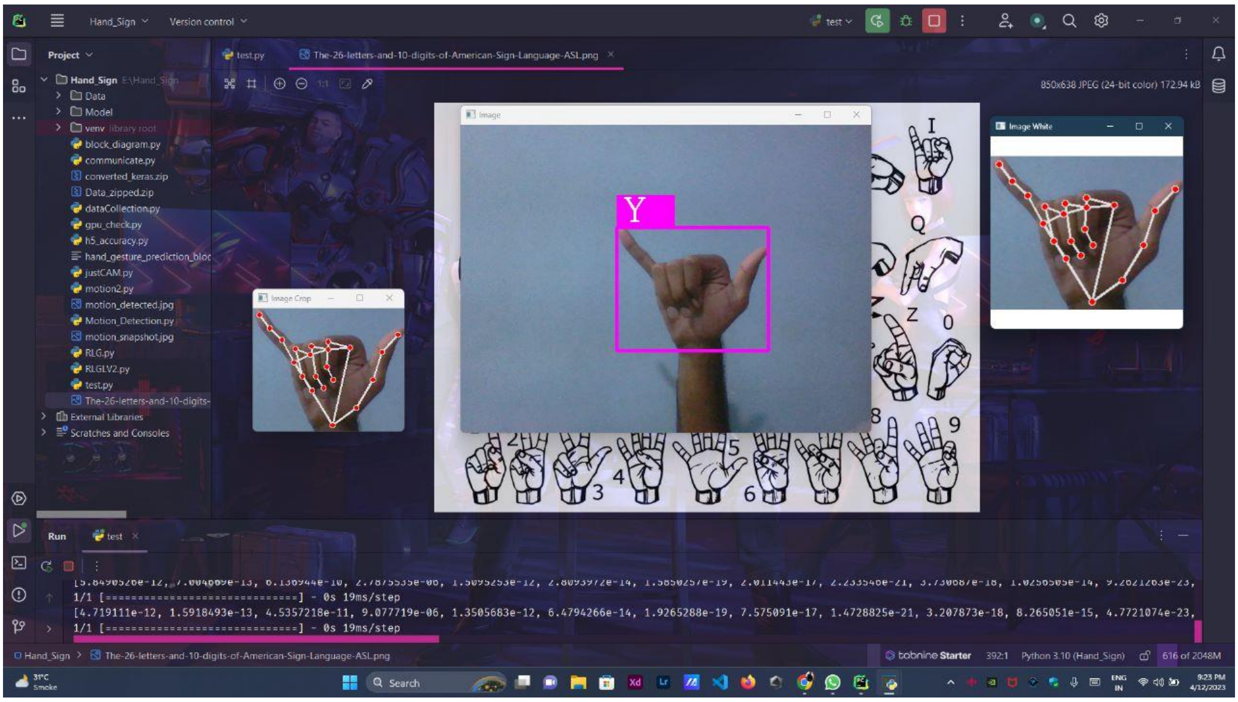

Fig 3.4 Output Y

CHAPTER 7

Conclusion

Conclusion

In conclusion, the development of a sign language detection system using Google Teachable Machine can be a promising approach to bridge the communication gap between individuals who are deaf or hard of hearing and those who are not. This project demonstrated the feasibility of using computer vision techniques to capture sign language gestures through a webcam and training a machine learning model to recognize and classify these gestures in real-time. The system requirements, included a computer with a compatible operating system (such as Windows, macOS, or Linux), a webcam with good image quality and appropriate mounting options, and sufficient RAM to handle the dataset and model complexity, which identified as key considerations for successful implementation. The use of Python and relevant libraries for additional tasks, such as data pre-processing and model evaluation, were also necessary with respect to this particular project's requirements. The project highlights the potential of Google Teachable Machine as a user-friendly platform for creating custom machine learning models without extensive coding or technical expertise. However, it's important to note that the accuracy and performance of the sign language detection system will depend on the quality and size of the dataset used for training, as well as the complexity of the gestures and the model architecture.

Further improvements and optimizations could include collecting a larger and more diverse dataset, refining the model architecture, incorporating feedback from users to improve the system's accuracy, and exploring options for real-world deployment on different platforms or devices. In summary, the development of a sign language detection system using Google Teachable Machine has the potential to enhance accessibility and inclusivity for individuals who are deaf or hard of hearing. The project provides valuable insights into the system requirements, challenges, implementation and future directions for similar projects, and encourages further research and development in this area to make sign language communication more accessible and inclusive for all.

References

Research Papers

[1] Parton, Becky Sue. "Sign language recognition and translation: A multidisciplined approach from the field of artificial intelligence." *Journal of deaf studies and deaf education* 11, no. 1 (2006): 94-101.

[2] Nikam, Ashish S., and Aarti G. Ambekar. "Sign language recognition using image-based hand gesture recognition techniques." In *2016 online international conference on green engineering and technologies (IC-GET)*, pp. 1-5. IEEE, 2016.

[3] Deshpande, Aditi, Ansh Shriwas, Vaishnavi Deshmukh, and Shubhangi Kale. "Sign Language Recognition System using CNN." In *2023 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE)*, pp. 906-911. IEEE, 2023.

[4] Wadhawan, Ankita, and Parteek Kumar. "Deep learning-based sign language recognition system for static signs." *Neural computing and applications* 32 (2020): 7957-7968.

[5] Kalsh, Er Aditi, and N. S. Garewal. "Sign language recognition system." International Journal of Computational Engineering Research (IJCER) 3, no. 6 (2013).

Useful Links

[6] https://www.nidcd.nih.gov/health/american-sign-language#:~:text=It%20is%20the%20primary%20language,some%20hearing%20people%20as%20well.

[7] https://www.geeksforgeeks.org/project-idea-sign-language-translator-speech-impaired/

[8] https://en.wikipedia.org/wiki/American_Sign_Language

[9] https://teachablemachine.withgoogle.com/

[10] https://docs.opencv.org/4.7.0/d6/d00/tutorial_py_root.html

[11] https://github.com/cvzone/cvzone

[12] https://numpy.org/learn/

[13] https://www.youtube.com/watch?v=wa2ARoUUdU8

[14] https://www.jetbrains.com/pycharm/learn/

[15] https://github.com/harshbg/Sign-Language-Interpreter-using-Deep-Learning

[16] https://blog.jetbrains.com/pycharm/2022/06/start-studying-machine-learning-with-pycharm/#:~:text=What%20tools%20are%20used%20in,Conda%2C%20and%20Jupyter%20Notebook%20integrations