ABSTRACT

Sign language is the means of communication for hearing-impaired people. There is a challenge for ordinary people to communicate with deaf people, which makes this system helpful in assisting them.

This project aims at implementing computer vision, which can take the signs from the users and convert them into text in real time.

The proposed system contains four modules such as: image capturing, pre-processing, classification, and prediction. By using image processing, the segmentation can be done. Sign gestures are captured and processed using the OpenCV python library. The captured gesture is resized, converted to a greyscale image, and the noise is filtered to achieve prediction with high accuracy. The classification and predication are done using a convolution neural network (CNN).

The converted real-time text appears in the text box. Generated text can be edited if necessary, and there is an option to save it on a clipboard, which can be pasted on any application we want.

The main features of this project are:

- It allows for all people who depend on sign language for effective communication.
- It helps to learn sign language for those who wish to learn sign language.
- It provides accurate sign language conversions in real-time.
- The converted text can be copied to any application.

REFERENCES

Sign Language Translation
Published in: 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)
Conducted by: IEEE
Date of Conference: 06-07 March 2020
Conference Location: Coimbatore, India

CHAPTER 1
INTRODUCTION

1.1 GENERAL INTRODUCTION

The most recent WHO estimate suggests that approximately 466 million people (or 6.1% of the world's population) were living with disabling hearing loss in 2018. This estimate is projected to rise to 630 million by 2030 and to over 900 million by 2050. In India, there are approximately 63 million people who are suffering from significant hearing impairment; this places the estimated prevalence at 6.3% of the Indian population.

This proposed system is a python application using packages like Tkinter, OpenCV, and TensorFlow. Where sign language is captured in real-time from the end user and these signs are evaluated through a CNN deep learning neural network, which was trained using machine learning mechanisms to predict each word and appear in the text field as output, which can be used for various applications.

1.2 GOAL OF THE PROJECT

The main means of communication for deaf people are through text and sign language. So, I am trying to demonstrate how effective it is when we integrate these sources of communication by making a sign-to-text transcript, which is beneficial to deaf people because it allows conversation between people and even modern technologies like voice assistants through sign language regardless of whether or not they are deaf, with the help of computer vision and deep learning CNN model.

Here I am using American Sign Language (ASL) since it is one of the most widely used sign languages in the world.

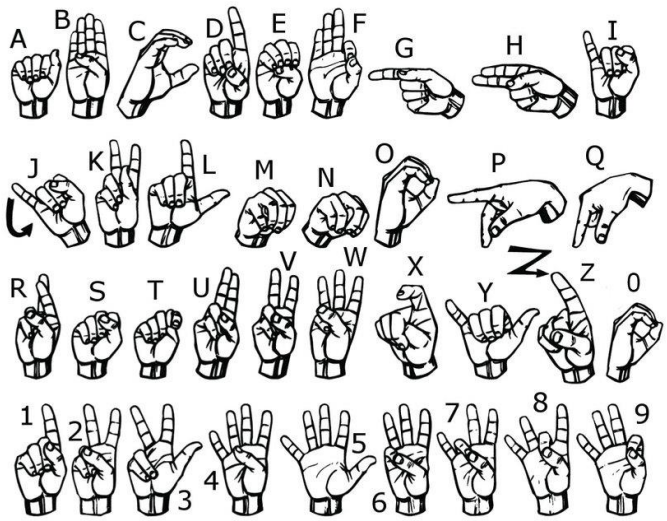


Figure 1: American Sign Language (ASL)

CHAPTER 2

LITERATURE SURVEY

2.1 STUDY OF SIMILAR WORK

There are many types of advanced and modern transcription applications that are available today on various platforms like desktops, smartphones, etc., web applications, hardware devices, or even as an embedded future on various applications like voice assistant systems.

The main leader in this field is Google Translate, which has lots of options and features like selecting translations according to the end user, predicting the language, and translating it through both voice and text transcript.

But there is a lack of a sign to corresponding text translation system even though there are lots of people dependent on it and different sign languages like American Sign Language (ASL), British Australian and New Zealand Sign Language (BANZSL), Indian Sign Language(ISL), etc. are some of the sign languages commonly used.

2.1.1 EXISTING SYSTEM

Existing Sign to text transcript systems are commonly available as prototypes that can't be commercially used with minimal and sometimes no user interface and we can't extract any resulting text for other applications just because it labels on the display view itself.
These Existing systems need to be manually trained whenever we try to use this system and it is not feasible for common users with minimal or no programming skills.
Lack of providing training or adding new training data to the existing system is a concern but there is a risk factor of overfitting and miss judge the outliers on the system which can affect the accuracy. So, research on overcoming this problem is just evolving now may be in the future there will be an optimal solution for this.

2.1.2 DRAWBACK OF EXISTING SYSTEM

- Lack of a good Graphical User interface
- Only prototypes are available
- Absence of full-fledged application
- Lack of availability for common people
- Absence of this feature as an integrated unit in modern technologies
- Language support is minimal
- Dependent on external environment for sign capture
- Not produce accurate results on some existing system according to external environments
- Some existing system is purely dependent on training data and it always need to be trained manually

CHAPTER 3
OVERALL DESCRIPTION

3.1 PROPOSED SYSTEM

The research paper entitled "Sign Language Translation" (2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)) conducted by IEEE at Coimbatore, India on March 06-07, 2020) presents a good general idea of how to implement one of the methods, i.e., by using live sign image capture, reducing background noise, evaluating using a conventional neural network(CNN) and generating results, which is one of the basic and easy ways to implement on various platforms.

By inspiring this paper, which basically focuses on implementation, I am trying to implement how this system can be commercialised to the end user as a fully-fledged application which can be accessed by ordinary people.

The process complexity of this proposed system is hidden from the end user. Only the required transcript text will appear as the output.

3.2 FEATURES OF PROPOSED SYSTEM

The aim of the proposed system is to translate American Sign Language (ASL) to text by using CNN and deep learning algorithms with the help of Tensorflow and Keras. Apart from that It shows real-time identification of which character is provided according to text on display itself as a label using OpenCV and this character forms into a text message on the text field provided. This text field can be further edited and can be copied using the copy to clipboard option using TKinter provided in this application itself, which can be pasted on various applications.

3.3 FUNCTIONS OF PROPOSED SYSTEM

There are basically four main functions for this proposed system:

1. **Image capturing**
   Python OpenCV library can be used to capture sign gestures from computer's internal camera. The dataset for various signs are collected. To predict gestures with high accuracy, around 25 images are collected for each sign.

2. **Pre-processing**
   The primary focus of the system is to support detecting gestures in dynamic background condition. To achieve this, the frames are pre-processed and converted to gray scale image and then background subtraction algorithm is used using Gaussian blur to reduce noise. The camera first captures around 10 frames to identify the background and compares current frame with previous frame. If a hand is detected, then the background is subtracted and only the hand gesture is converted to gray scale and transfers to model for classification and prediction.

3. **Classification**
   After collecting and processing the image dataset, they have to be classified. Convolutional neural network is used to analyse and classify visual imagery. It is

widely applied in image recognition and classification, natural language processing, etc. CNN is regularized versions of multilayer perceptron.

4. **Prediction**
   The CNN model evaluates the real time input from end user and predicts the output.

3.4 REQUIREMENTS SPECIFICATION

1. **Accuracy**
   The proposed system should be accurate on generating results based on given inputs.

2. **Speed**
   The proposed system should be in real time for generating results.

3. **Flexible**
   The proposed system should be flexible to new updates and patches in near future.

4. **Good Interface**
   The proposed system should maintain good interface even after upgradations.

3.5 FEASIBILITY STUDY

Feasibility Study in Software Engineering is a study to evaluate feasibility of proposed project or system. Feasibility study is one of stage among important four stages of Software Project Management Process. As name suggests feasibility study is the feasibility analysis or it is a measure of the software product in terms of how much beneficial product development will be for the organization in a practical point of view. Feasibility study is carried out based on many purposes to analyse whether software product will be right in terms of development, implantation, contribution of project to the organization etc.
In our proposed system the product is feasibility can be achieved in all four aspects Technical Operational, Economical and Behavioural.

3.5.1 TECHNICAL FEASIBILITY

In Technical Feasibility current resources both hardware software along with required technology are analysed/assessed to develop project. This technical feasibility study gives report whether there exists correct required resources and technologies which will be used for project development. Along with this, feasibility study also analyses technical skills and capabilities of technical team, existing technology can be used or not, maintenance and up-gradation is easy or not for chosen technology etc. In this proposed system technical feasibility is achieved according to above criteria.

3.5.2 OPERATIONAL FEASIBILITY

In Operational Feasibility degree of providing service to requirements is analysed along with how much easy product will be to operate and maintenance after deployment. Along with this other operational scopes are determining usability of product, determining suggested

solution by software development team is acceptable or not etc. The Operational feasibility can be ensured by the proposed system.

3.5.3 ECONOMICAL FEASIBILITY

Economic feasibility the most important and frequently used method for evaluating the effectiveness of the proposed system. It is very essential because the main goal of the proposed system is to have economically better results along with increased efficiency. Cost benefit analysis is usually performed for the expected from the proposed system. Since the organization is well equipped with the required hardware, the project was found to be economically feasible and the users who possess a device supports Windows operating system can easily use it.

3.5.4 BEHAVIORAL FEASIBILITY

The proposed system satisfies behavioral feasibility because the system is providing with good and minimalistic GUI which can easily be understand for any end users and its encapsulates the conversion procedure from the users. Hence it’s easier to operate the system with ease.

CHAPTER 4

OPERATING ENVIRONMENT

4.1 HARDWARE REQUIREMENTS

1. Processor: Dual Core 1.60 GHz or higher
2. Hard disk: 500 GB
3. RAM: 4GB
4. Monitor: 17" Color Monitor or higher
5. Mouse: Microsoft
6. Keyboard: Microsoft multimedia keyboard

4.2 SOFTWARE REQUIREMENTS

1. Operating System: Windows 8.1 Pro or higher
2. Framework: Microsoft .Net Framework
3. Environment: PyCharm Community Edition 2022.2
4. Language: Python 3.8, Open CV, and Tensor Flow 2.3.0
5. Documentation: Microsoft Word 2010 or higher

4.3 TOOLS AND PLATFORMS

4.3.1 PYCHARM

PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web, and data science development.

4.3.2 PYTHON 3.8

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

4.3.3 TKINTER

The tkinter package ("Tk interface") is the standard Python interface to the Tcl/Tk GUI toolkit. Both Tk and tkinter are available on most Unix platforms, including macOS, as well as on Windows systems.
Running python -m tkinter from the command line should open a window demonstrating a simple Tk interface, letting you know that tkinter is properly installed on your system, and also showing what version of Tcl/Tk is installed, so you can read the Tcl/Tk documentation specific to that version.
Tkinter supports a range of Tcl/Tk versions, built either with or without thread support. The official Python binary release bundles Tcl/Tk 8.6 threaded.

Tkinter is not a thin wrapper, but adds a fair amount of its own logic to make the experience more pythonic

4.3.4 OPEN CV

Open CV (*Open Source Computer Vision Library*) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez (which was later acquired by Intel). The library is cross-platform and free for use under the open-source Apache 2 License. Starting with 2011, OpenCV features GPU acceleration for real-time operations.

4.3.5 DEEP LEARNING

Deep learning is a machine learning technique that teaches computers to do what comes naturally to humans: learn by example. Deep learning is a key technology behind driverless cars, enabling them to recognize a stop sign, or to distinguish a pedestrian from a lamppost. It is the key to voice control in consumer devices like phones, tablets, TVs, and hands-free speakers. Deep learning is getting lots of attention lately and for good reason. It's achieving results that were not possible before.
In deep learning, a computer model learns to perform classification tasks directly from images, text, or sound. Deep learning models can achieve state-of-the-art accuracy, sometimes exceeding human-level performance. Models are trained by using a large set of labeled data and neural network architectures that contain many layers.

4.3.6 CNN

In deep learning, a **convolutional neural network** (**CNN/ConvNet**) is a class of deep neural networks, most commonly applied to analyze visual imagery. Now when we think of a neural network we think about matrix multiplications but that is not the case with ConvNet. It uses a special technique called Convolution. Now in mathematics **convolution** is a mathematical operation on two functions that produces a third function that expresses how the shape of one is modified by the other.

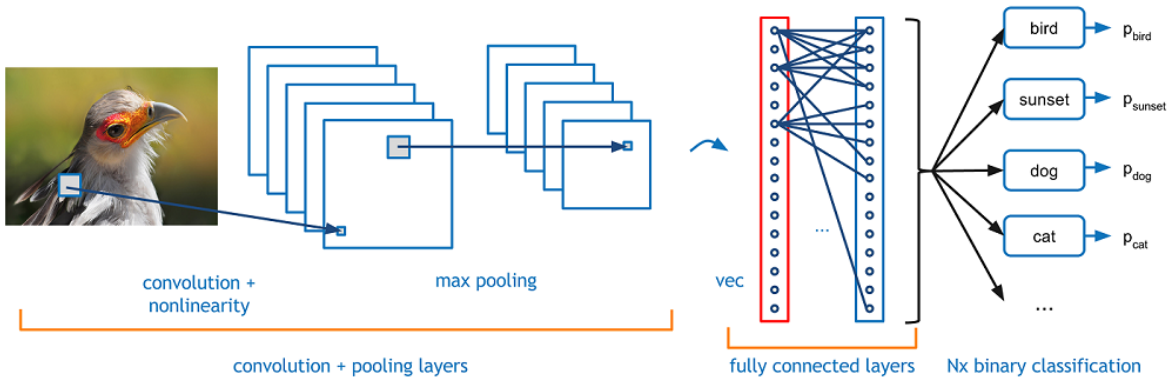


Figure 2: CNN

Convolutional neural networks are composed of multiple layers of artificial neurons. Artificial neurons, a rough imitation of their biological counterparts, are mathematical functions that calculate the weighted sum of multiple inputs and outputs an activation value. When you input an image in a ConvNet, each layer generates several activation functions that are passed on to the next layer.
The first layer usually extracts basic features such as horizontal or diagonal edges. This output is passed on to the next layer which detects more complex features such as corners or combinational edges. As we move deeper into the network it can identify even more complex features such as objects, faces, etc.

4.3.7 TENSORFLOW 2 AND KERAS

TensorFlow 2 is an end-to-end, open-source machine learning platform. You can think of it as an infrastructure layer for differentiable programming. It combines four key abilities:

- Efficiently executing low-level tensor operations on CPU, GPU, or TPU.
- Computing the gradient of arbitrary differentiable expressions.
- Scaling computation to many devices, such as clusters of hundreds of GPUs.
- Exporting programs ("graphs") to external runtimes such as servers, browsers, mobile and embedded devices.

Keras is the high-level API of TensorFlow 2: an approachable, highly-productive interface for solving machine learning problems, with a focus on modern deep learning. It provides essential abstractions and building blocks for developing and shipping machine learning solutions with high iteration velocity.

Keras empowers engineers and researchers to take full advantage of the scalability and cross-platform capabilities of TensorFlow 2: you can run Keras on TPU or on large clusters of GPUs, and you can export your Keras models to run in the browser or on a mobile device.

4.3.8 JUPYTER NOTEBOOK

Jupyter Notebook (formerly IPython Notebook) is a web-based interactive computational environment for creating notebook documents. Jupyter Notebook is built using several open-source libraries, including IPython, ZeroMQ, Tornado, jQuery, Bootstrap, and MathJax. A Jupyter Notebook document is a browser-based REPL containing an ordered list of input/output cells which can contain code, text (using Markdown), mathematics, plots and rich media. Underneath the interface, a notebook is a JSON document, following a versioned schema, usually ending with the ".ipynb" extension.

4.3.9 NVIDIA CUDA

CUDA (or **Compute Unified Device Architecture**) is a parallel computing platform and application programming interface (API) that allows software to use certain types of graphics processing units (GPUs) for general purpose processing, an approach called general-purpose computing on GPUs (GPGPU). CUDA is a software layer that gives direct access to the GPU's virtual instruction set and parallel computational elements, for the execution of compute kernels.

CUDA is designed to work with programming languages such as C, C++, and Fortran. This accessibility makes it easier for specialists in parallel programming to use GPU resources, in contrast to prior APIs like Direct3D and OpenGL, which required advanced skills in graphics programming. CUDA-powered GPUs also support programming frameworks such as OpenMP, OpenACC and OpenCL and HIP by compiling such code to CUDA.

4.3.10 CANVA

Canva is an Australian graphic design platform, used to create social media graphics, presentations, posters, documents and other visual content The app includes templates for users to use. The platform is free to use and offers paid subscriptions such as Canva Pro and Canva for Enterprise for additional functionality. In 2021, Canva launched a video editing tool. Users can also pay for physical products to be printed and shipped. It has announced it intends to compete with Google and Microsoft in the office software category, with website and whiteboard products.

CHAPTER 5
DESIGN

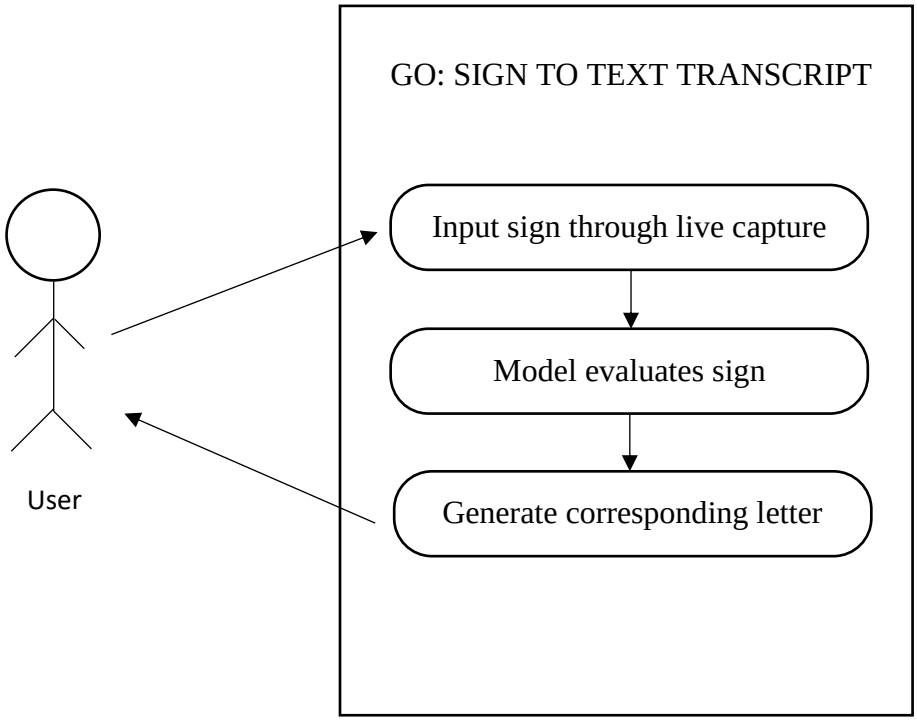
5.1 SYSTEM DESIGN

System design is the process of defining the architecture, modules, and data for a system to satisfy specified requirements. It is the phase where the SRS document is converted into a format that can be implemented and decides how the system will operate. The purpose of design phase is to plan a solution for problem specified by the requirements. System design aims to identify the modules that should be in the system, the specification of those modules and how they interact with each other to produce the result. The goal of the design process is to produce a model for or representation of a system can be used later to build. The produced model is called design of the system.

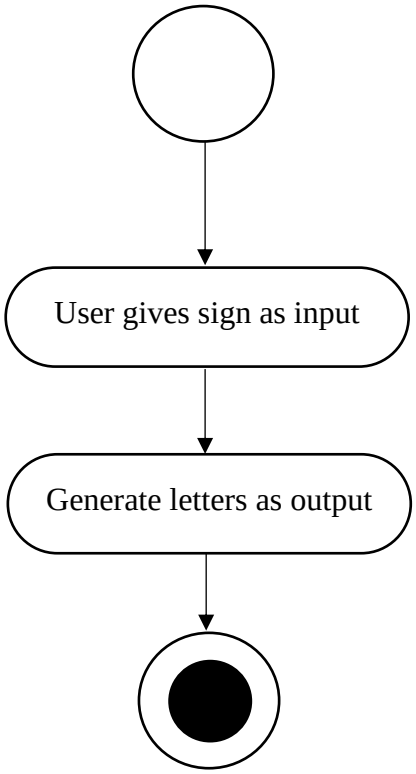
5.2 PROGRAM DESIGN

- As a user, I want to learn one of the widely used sign languages American Sign Language (ASL). So that I can implement it as my sign language in my system.
- As a user, I want to explore the dataset. So that I can use it for sign detection.
- As a user, I want to learn OpenCV. So that I can capture live signs from the user end.
- As a user, I want to learn dimension tables. So that I can know about the filter size, output size, etc. of an image.
- As a user, I want to obtain the partitioned dataset. So that I can get the test images and train images.
- As a user, I want to learn CNN architecture. So that I can explore the layers and description.
- As a user, I want to train the model using the dataset. So that I can start testing.
- As a user, I want to implement the algorithm by using a trained model. So that I can extract only the sign language and respond with the corresponding letter.
- As a user, I want to append corresponding letters to the text field. So that generated letters can form a sentence.
- As a user, I want to implement copy to clipboard button. So that I can copy generated sentences to desired applications by using a button.
- As a user, I want to implement a reset to defaults button. So that I can start generating letters from the beginning.

5.3 USECASE DIAGRAM

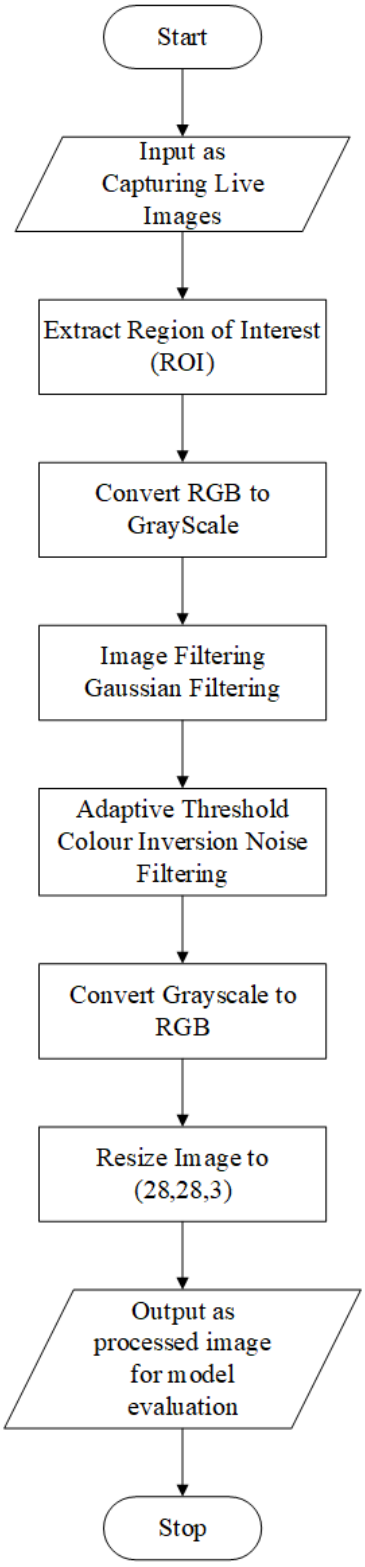


5.4 ACTIVITY DIAGRAM

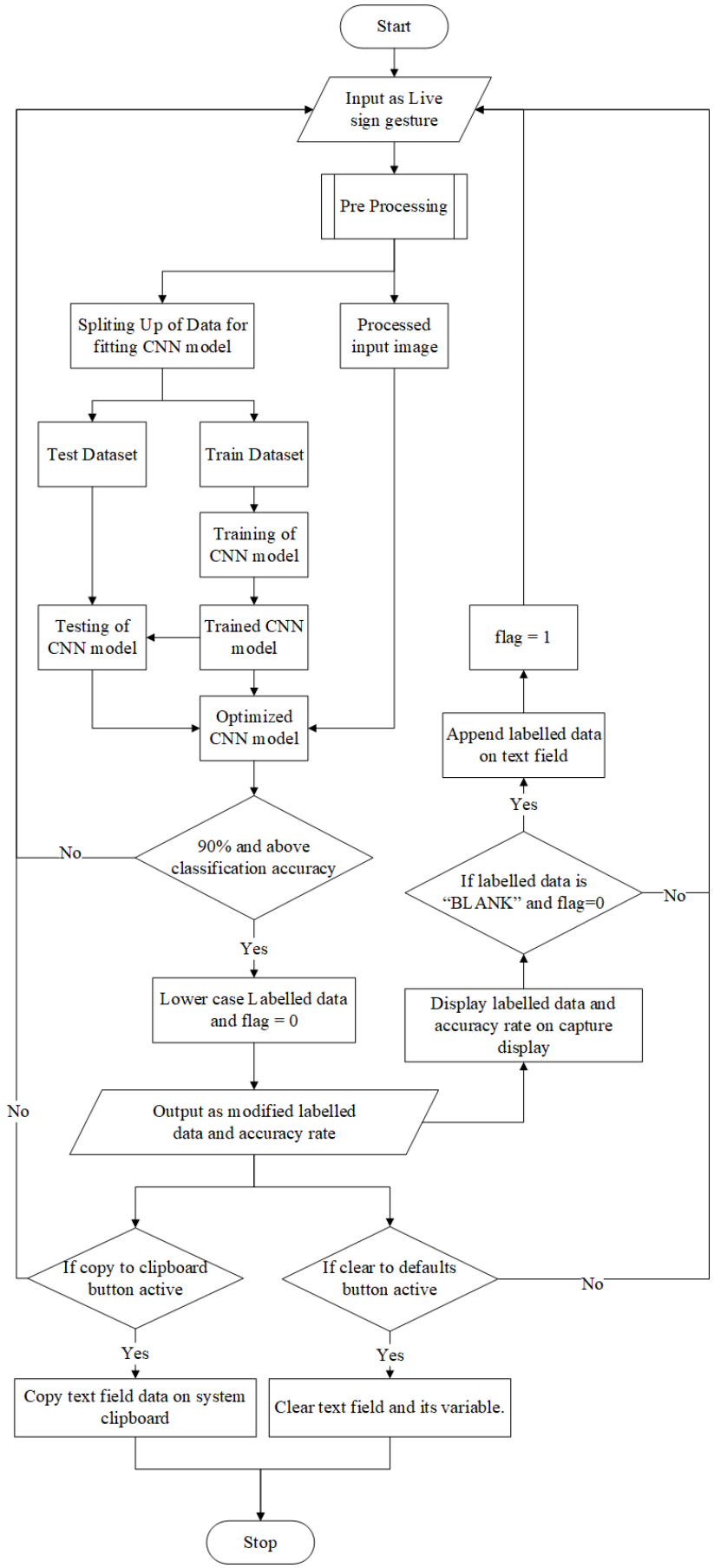


5.5 PROPOSED PROJECT PIPELINE

5.5.1 PIPELINE FOR PRE PROCESSING



5.5.2 PIPELINE FOR PROPOSED SYSTEM



5.6 MODEL ARCHITECTURE

5.6.1 LAYERED VIEW

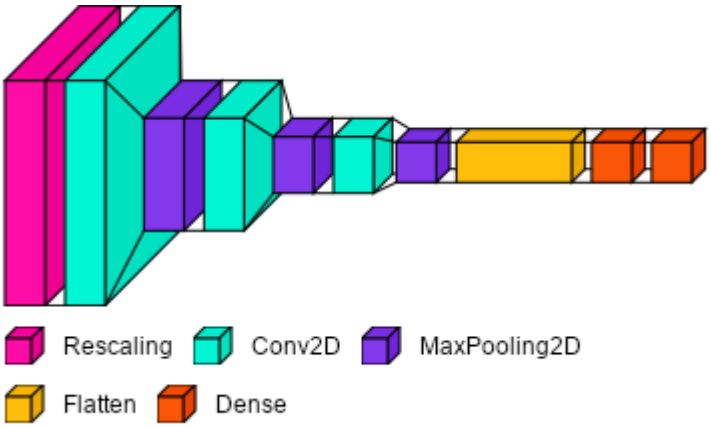


Figure 3: Layered view of model

5.6.2 MODEL SUMMARY

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 rescaling_1 (Rescaling)     (None, 28, 28, 3)         0

 conv2d (Conv2D)             (None, 28, 28, 16)        448

 max_pooling2d (MaxPooling2D  (None, 14, 14, 16)       0
 )

 conv2d_1 (Conv2D)           (None, 14, 14, 32)        4640

 max_pooling2d_1 (MaxPooling  (None, 7, 7, 32)         0
 2D)

 conv2d_2 (Conv2D)           (None, 7, 7, 64)          18496

 max_pooling2d_2 (MaxPooling  (None, 3, 3, 64)         0
 2D)

 flatten (Flatten)           (None, 576)               0

 dense (Dense)               (None, 128)               73856

 dense_1 (Dense)             (None, 37)                4773

=================================================================
Total params: 102,213
Trainable params: 102,213
Non-trainable params: 0
_________________________________________________________________
```

5.6.3 MODEL ACCURACY

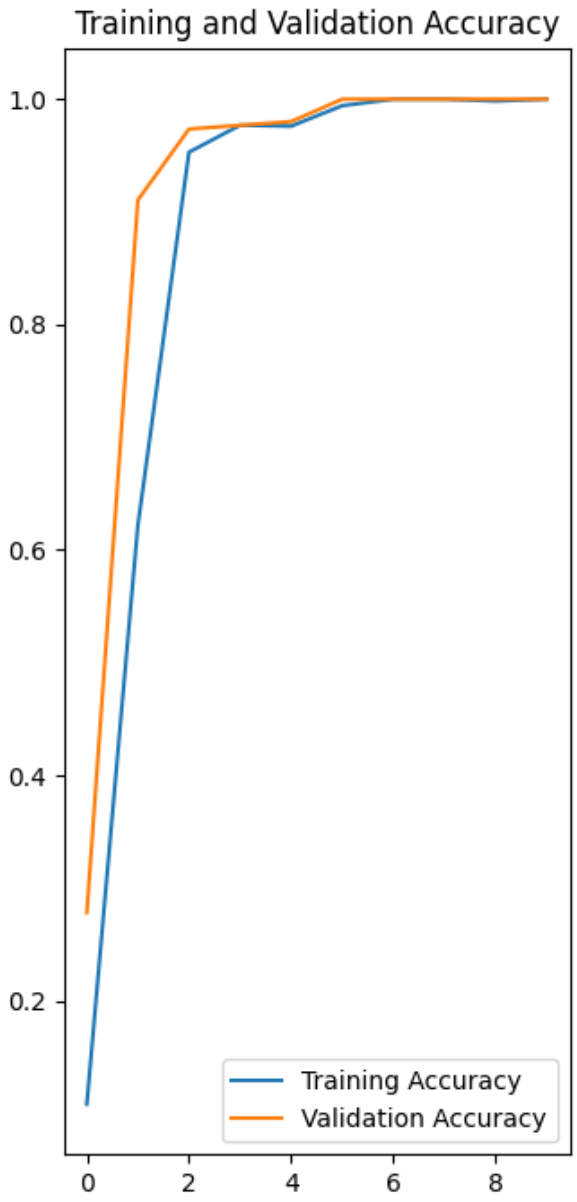


Figure 4: Accuracy Graph

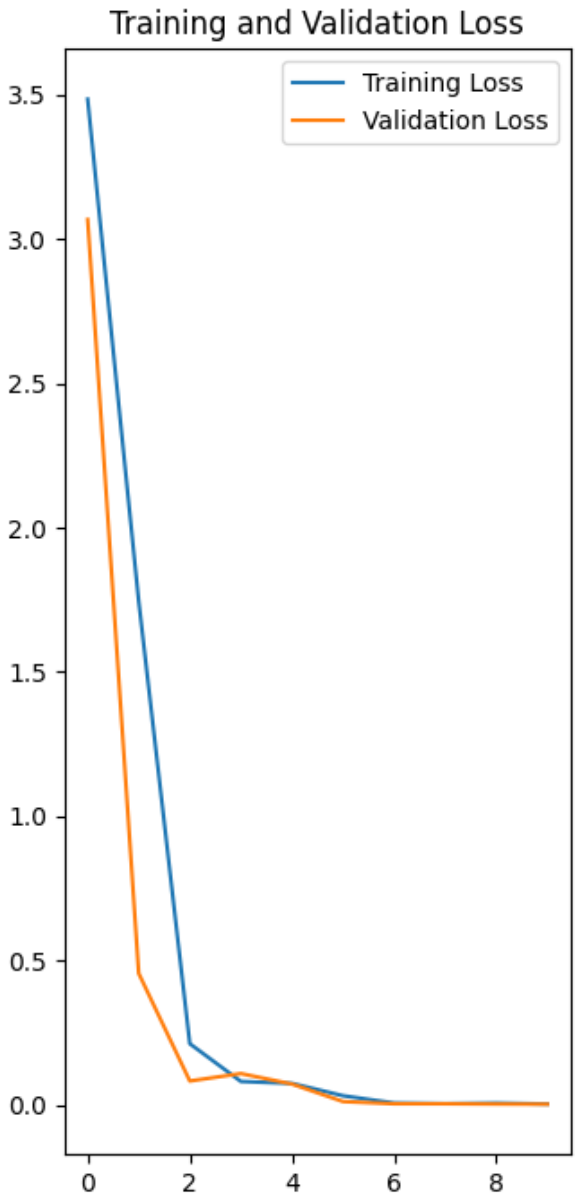


Figure 5: Loss Graph

5.7 INPUT DESIGN

Input is the process of converting user inputs computer-based format. The project requires a set of information from the user to prepare a report. In the order, when organized input data are needed. Input data is collected and organized into groups of similar data. The goal behind designing input data is to make the data entry easy and make it free from logical error. So, the input screens in the system should be really flexible and faster to use. The user need not to input any data manually, we are using the video data directly from the web camera.

Objectives: -

- To produce a cost-effective method of input.
- To achieve the highest possible level of accuracy.
- To ensure that the input is acceptable and understandable.
- To make clutter free screens.
- The prevention of irrelevant data entry.
- To make a user-friendly input screen.

Here in our system, 'GO: Sign Language to Text Transcript', input screens ensure the reliability and accuracy of the system. The input design determines whether the user can interact directly with the computer. With input design, we can say that it is more user friendly as compared to the existing manual system containing paper operations.

5.8 OUTPUT DESIGN

Outputs are the most important direct source of information to the user. Efficient and eligible output design should improve the system's relationship with the user, Output design generally deals with the results generated by the system i.e., letters corresponding to sign gestures. These generated texts can be further used from text field and can be visible to end user. The end users will not actually operate the interior model of system or enter new training dataset to the system, but they will use the output from the system. The system provides a comprehensive output screen with graphical representations of each letter with confidence percentage for user to better understand the provided data.

CHAPTER 6

FUNCTIONAL AND NON FUNCTIONAL REQUIREMENTS

6.1 FUNCTIONAL REQUIREMENT

In software engineering, a functional requirement defines a function of a software system or its component. A function is described as set of inputs, the behaviour, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Generally functional requirements are expressed in the form of "system must do requirement".

6.2 NON FUNCTIONAL REQUIREMENT

A non-functional requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours. Some of the non-functional requirements are mentioned below.

Usability: The system shall have a clean interface with only needed features, clear terminology and tools tips where necessary. Letters for corresponding sign shall be specified in clear way.

Efficiency: The system shall respond to different sign languages with ease and effective manner.

Portability: The system shall be independent of the specific technological platform used to implement it.

Reliability: Reliability defined as a measure of the time between failures occurring in system, so that the system shall operate without any failures for a particular period of time.

Availability: Availability measures the percentage of time the system is in its operational state so that the system be available for use 24 hours per day and 365 days per year.

CHAPTER 7

TESTING

7.1 TESTING STRATEGIES

Software Testing is the process of executing a program or system with the intent of finding errors. Testing involves any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results. The scope of software testing includes examination of code as well as execution of that code in various environments and conditions as well as examining the quality aspects of code: does it do what it is supposed to Do and do what it needs to do. Testing helps not only to uncover errors introduced during coding, but also locates errors committed during the previous phases.

Testing Objectives Include:
- Testing is a process of executing a program with the intent offending an error.
- A good test case is one that has a probability of finding an as yet undiscovered error.

Testing Principles:
- All tests should be traceable to end user requirements
- Tests should be planned long before testing begin
- Testing should begin on a small scale and progress towards testing in large
- Exhaustive testing is not possible.
- To be most effective testing should be conducted by an independent third party.

Implementation is the stage of the project where the theoretical design is turned into a working system. At this stage the main workload, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned and controlled, it can cause chaos and confusion.

Testing helps not only to uncover errors introduced during coding, but also locates errors committed during the previous phases. Thus the aim of testing is to uncover requirements, design or coding errors in the program. Software Testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding, Testing present interesting anomaly for the software engineer.

7.2 UNIT TESTING

This is the first of testing. In this different modules are tested against the specification produces during the design of the modules. It refers to the verification of single program module in an isolated environment. Unit testing focuses on the modules independently of one another to locate errors.

In our project we test each module and each forms individually. Each forms have tested using appropriate values. The input screens need to be designed very carefully and logically. While entering data in the input forms, proper validation checks are done.

7.3 INTEGRATION TESTING

Integration testing (sometimes called **integration and testing**, abbreviated **I&T**) is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing is conducted to evaluate the compliance of a system or component with specified functional requirements. It occurs after unit testing and before system testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

In our project we used incremental top down approach in which, testing is done by integrating two or more modules that are logically related to each other and then tested for proper functioning of the application. Then the other related modules are integrated incrementally and the process continues until all the logically related modules are integrated and tested successfully i.e. from testing is done from prediction of model to image capturing by integrating on each iterations.

7.4 SYSTEM TESTING

System testing is testing conducted on a complete integrated system to evaluate the system's compliance with its specified requirements.

System testing takes, as its input, all of the integrated components that have passed integration testing. The purpose of integration testing is to detect any inconsistencies between the units that are integrated together. System testing seeks to detect defects both within the "inter-assemblages" and also within the system as a whole. The actual result is the behaviour produced or observed when a component or system is tested.

System testing is performed on the entire system in the context of either functional requirement specifications (FRS) or system requirement specification (SRS), or both. System testing tests not only the design, but also the behaviour and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software or hardware requirements specifications.

7.5 TESTING RESULTS

TEST CASE ID	TEST CASE NAME	TEST CASE STEP	EXPECTED RESULT	STATUS	DEFECTS
1	Image Capturing	Sign gestures are captured and processed using the OpenCV python library.	Camera ON Real time and start capturing.	PASS	NIL
2	Pre-processing	The captured gesture is resized, converted to a grayscale image, and the noise is filtered to achieve prediction with high accuracy.	Capturing only ROI (Region of Interest) and conversion of image.	PASS	NIL
3	Classification	The classification and predication are done using a convolution neural network (CNN).	Model should classify different signs to corresponding letters by using given dataset.	PASS	NIL
4	Prediction	The converted real-time text appears in the text box. Generated text can be edited if necessary, and there is an option to save it on a clipboard, which can be pasted on any application we want.	Model should predict classified labels corresponds to live sign language with good confidence rate and these results should appear on text field.	PASS	NIL

CHAPTER 8
RESULTS AND DISCUSSION

8.1 RESULTS (SALIENT FEATURES)

The objective of the proposed system is to identify corresponding letters for given American Sign Language (ASL) symbols with the help of computer vision, machine learning and deep learning. We make this possible with the help of Python libraries like OpenCV, Tensor Flow, Keras, etc... Then we can use the model to classify images. At first, we have to capture the sign languages provided by end user by filtering out the noises.

- User interface is designed such that they are very user friendly and the user is not required to input data manually.
- Not much training required.
- Easy analysis of data and statistical view.
- There are no needs of experts.
- The new system is more user friendly.
- This system is much faster and efficient than the old system.
- Lower points failure than the old system.
- It is cheaper than the existing system.
- No need for expensive equipment.
- The system not only generates corresponding letters. It combines these letters to form a sentences on the text filed provided.
- This text field can be edited and further be copied on different applications.

8.2 SCREEN SHOTS

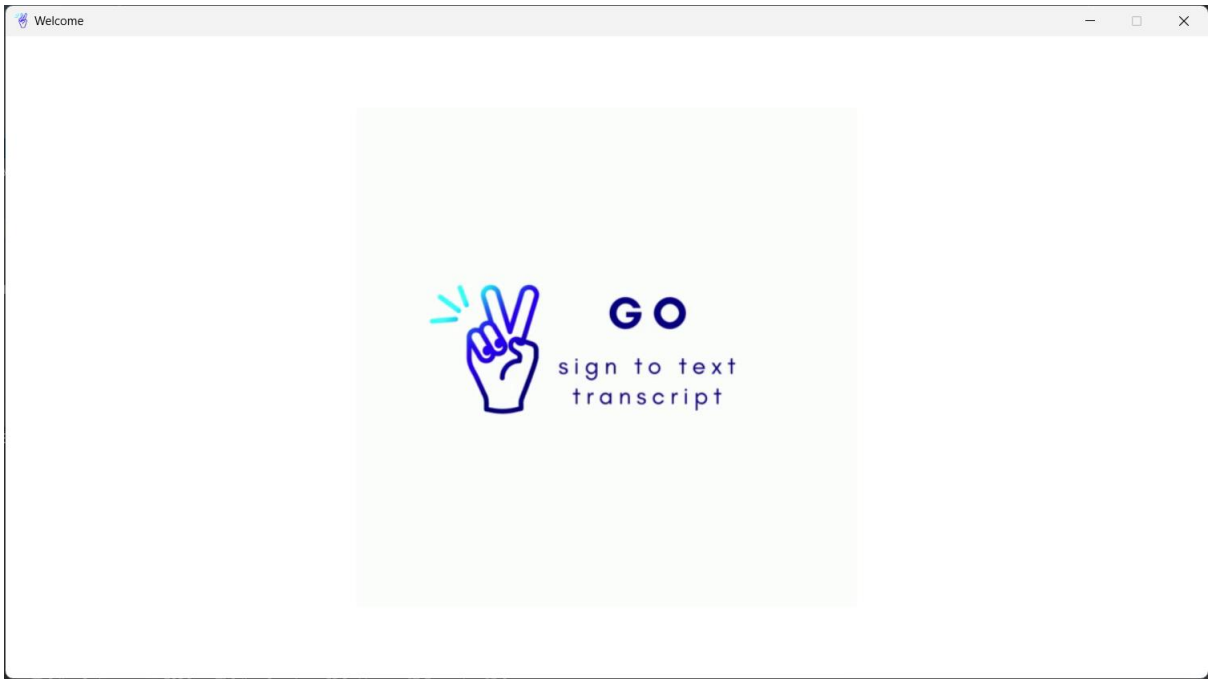


Figure 6: Loading page

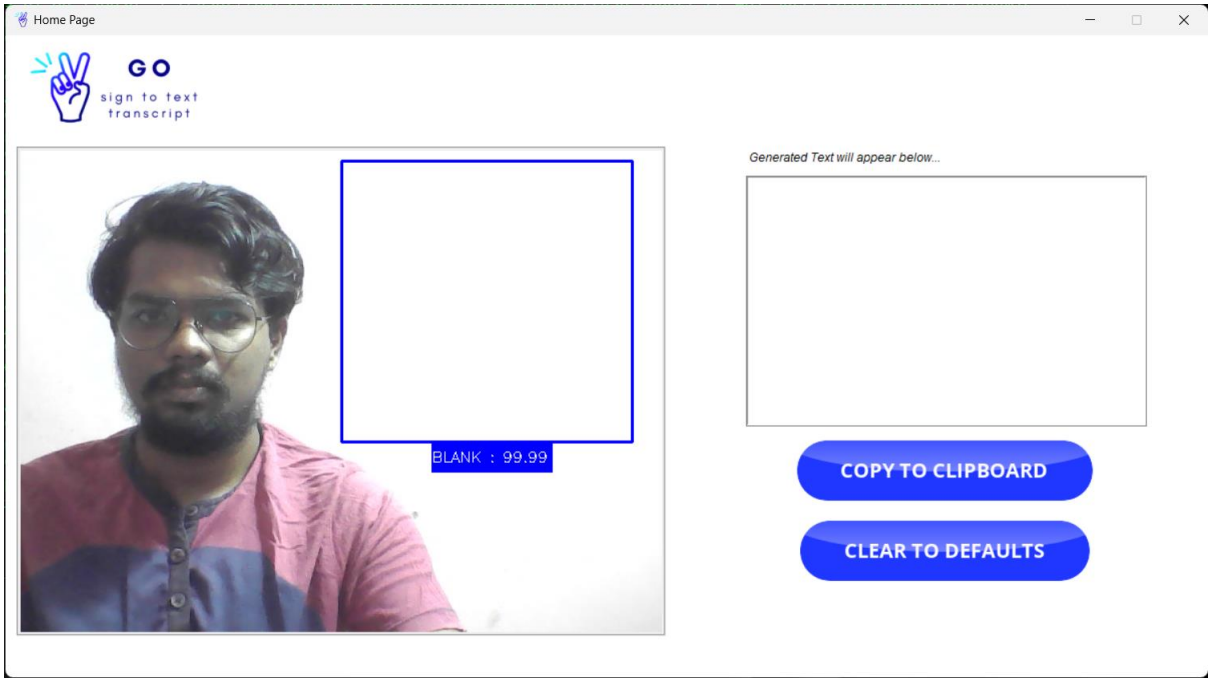


Figure 7: Home page

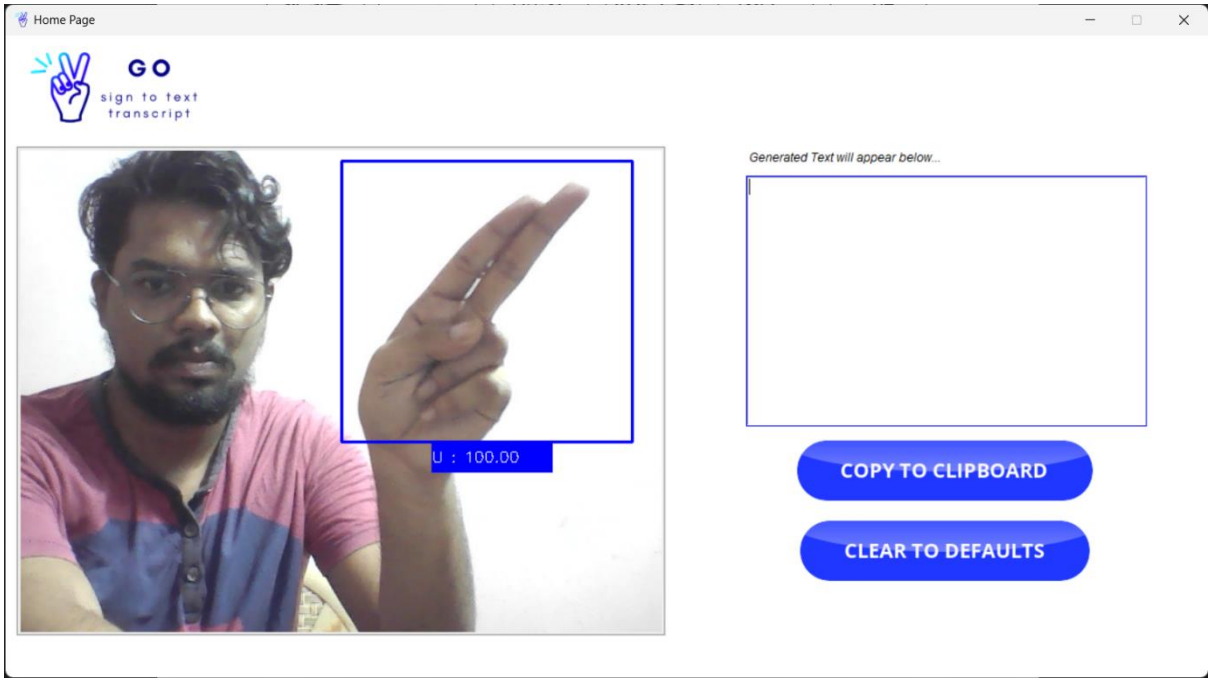


Figure 8: Capturing symbol for letter “u” more than 90% accuracy

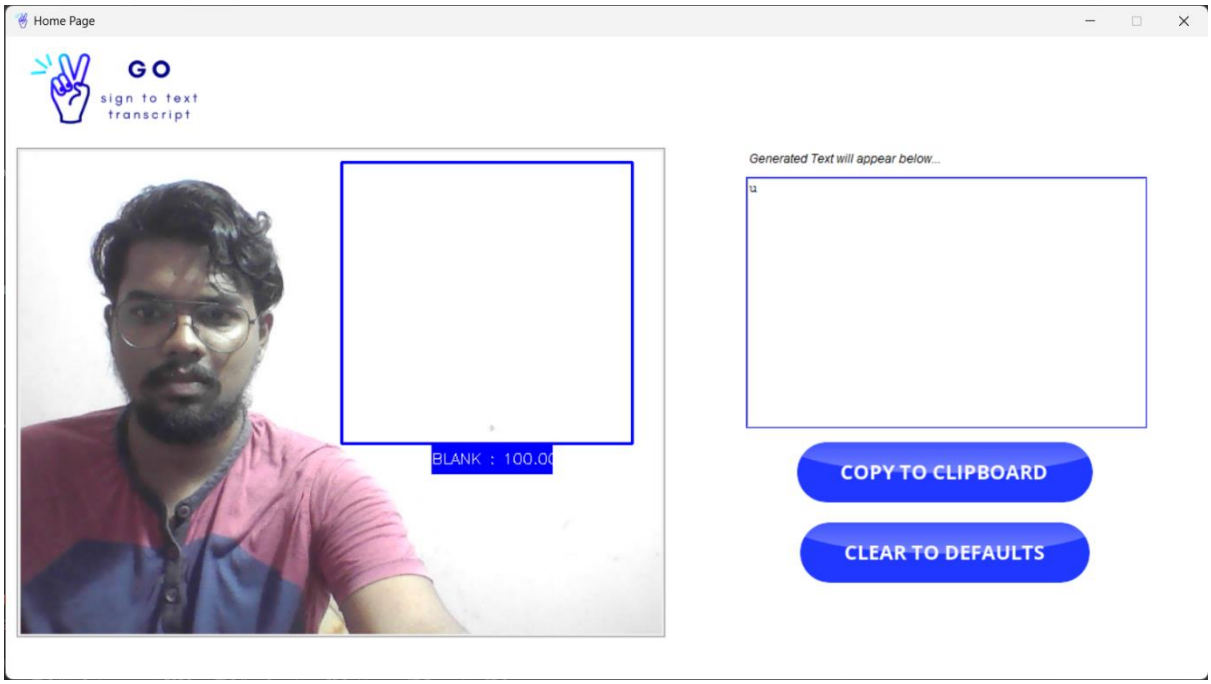


Figure 9: The letter “u” gets appended on text field after we release the symbol

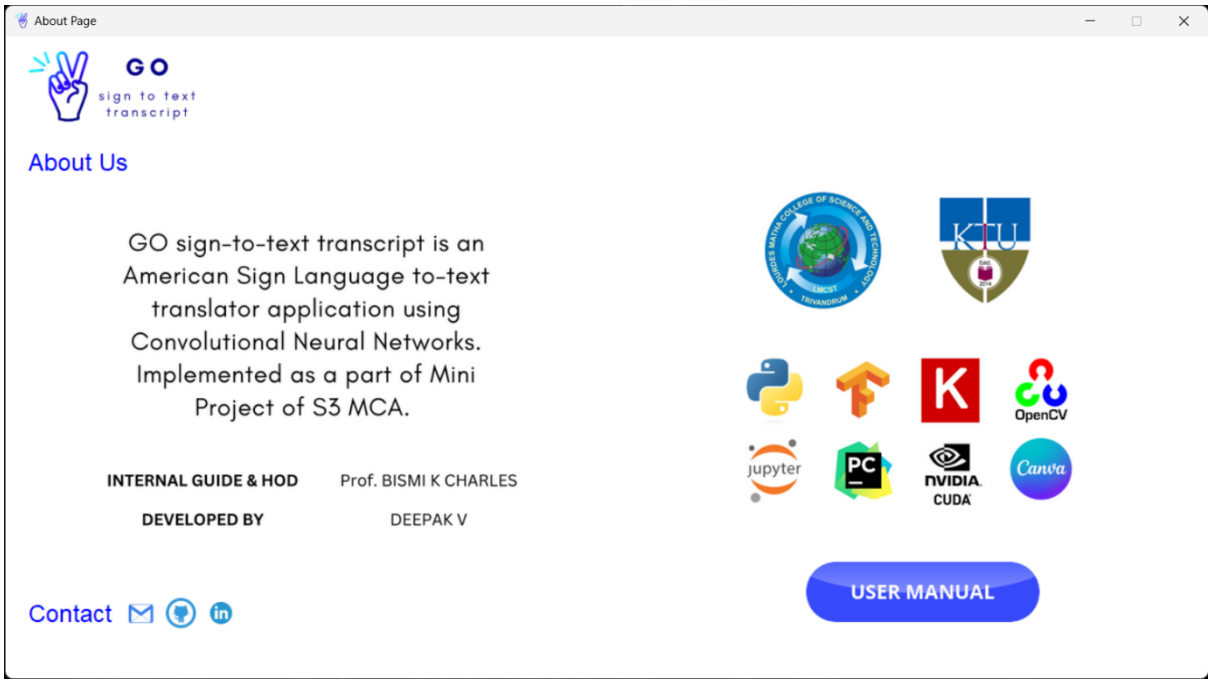


Figure 10: About page

CHAPTER 9
CONCLUSION

9.1 SYSTEM IMPLEMENTATION

After the system has been tested, the implementation type or the changeover technique from the existing system to the new system is a step-by-step process. In the system at first only a module of the system is implemented and checked for suitability and efficiency. When the end user related to the particular module is satisfied with the performance, the next step of implementation is preceded.

Backups are necessary since any time unexpected events may happen. And so during the program execution, the records are stored in the workspace. This helps to recover the original status of the records from any accidental updating or intentional deletion of records.

An Implementation plan is a management tool for a specific policy measure, or package of measures, designed to assist agencies to manage and monitor implementation effectively. Implementation plans are intended to be scalable and flexible; reflecting the degree of urgency, innovation, complexity and or sensitivity associated with the particular policy measure. Agencies are expected to exercise judgment in this area; however, the level of detail should be sufficient to enable the agency to effectively manage the implementation of a policy measure. At a minimum, plans should reflect the standards outlined in the Guide to Preparing Implementation Plans.

The implementation stage involves following tasks:

- Careful planning
- Investigation of system and constraints
- Design of method to achieve the changeover phase

9.2 CONCLUSION

Deep learning helps computers to derive meaningful links from a plethora of data and make sense of unstructured data. Here, the mathematical algorithms are combined with a lot of data and strong hardware to get qualified information. With this method, information from digital data can be automatically extracted, classified and analyzed.

Although deep learning has been around for several years, the trend has only really picked up in the last three to four years. The reason for this was among other things better hardware resources, more sophisticated algorithms and optimized neural networks. Deep learning is not a new approach but a development of the older approach of artificial neural networks.

The proposed method presented in this project is mainly for people who depend on sign language and act as a helping hand for them. This system can be utilized on various platforms like kiosk system, various mobile and desktop applications, etc... With deep learning, particularly CNN, the model is able to train and learn to identify different sign languages using training and validation datasets.

9.3 FUTURE SCOPE

Since our proposed system is simple, flexible and easy to customized we can bring lots of enhancements like,

Implementation of media pipe rather than simple opencv image capturing. This can help capturing inputs from various lighting conditions.

Conversion sign language to audio as an option. This can help to send audio message to other people.

Introduce new custom signs and can be trained by end users in abstract manner. This can help to add our own language or even functionalities if we modify little bit.

Sharing of custom sign model to other users and can be used through this application. Our custom language can be shared to other people and can be used through this application.

Creating an online community to excavate and elevate the enhancement of this application. This can be implemented through making a website for our application for developing new ideas and implement those so that this system can be reliable and exist forever.

These all can be brought as upcoming version updates of this system.

BIBLIOGRAPHY

BOOKS

[1] Convolutional neural networks for visual computing (Chapter 4), Ragav Venkatesan and Baoxin Li CRC press

[2] Online book Dive Deep into Machine Learning at https://d2l.ai/

[3] E. Alpayidin, Introduction to Machine Learning, Prentice Hall of India (2005)

[4] Jeeva Jose, "Taming Python by Programming", Khanna Publishers, New Delhi, 2018

WEBSITES

[5] https://www.tensorflow.org/

[6] https://www.tutorialspoint.com/python/python_gui_programming.htm

[7] https://developers.google.com/machine-learning

[8] https://www.geeksforgeeks.org/opencv-overview/

[9] https://opencv.org/

JOURNEL AND PUBLICATIONS

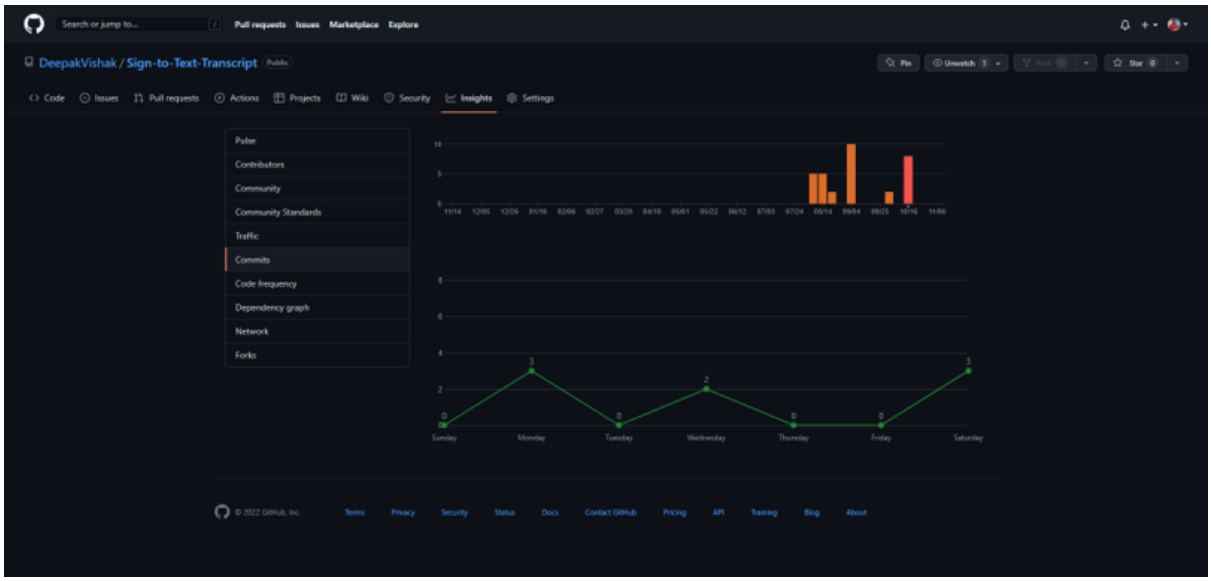
[10] Sign Language Translation**;** 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS); IEEE; 06-07 March 2020 https://ieeexplore.ieee.org/document/9074370

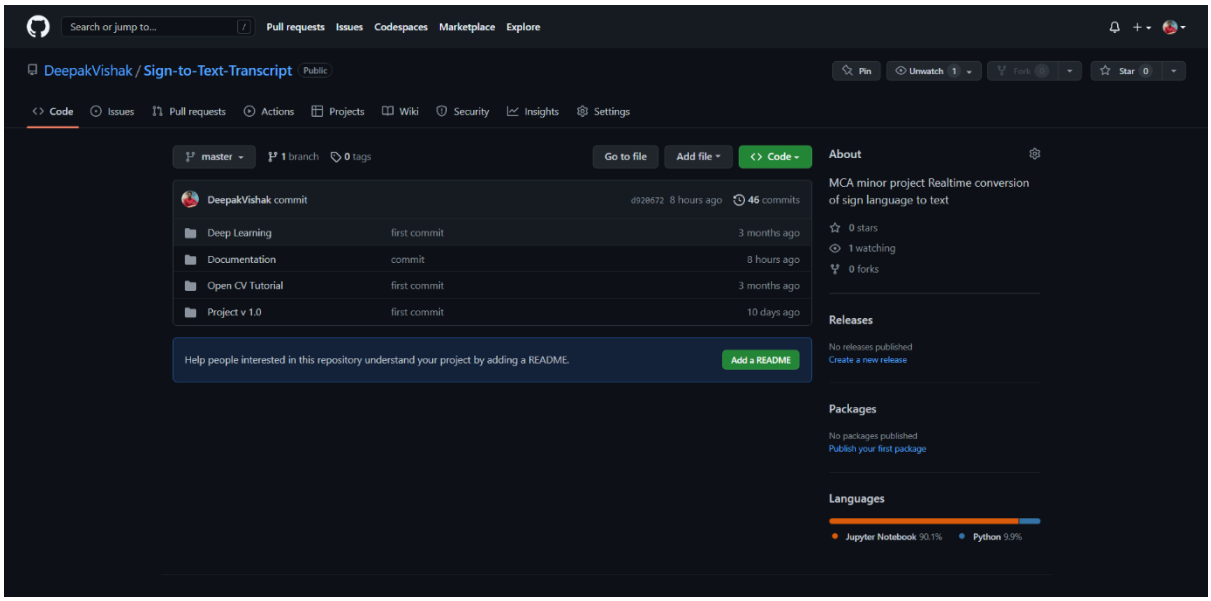
APPENDICES

- **LIST OF TABLES**

- **LIST OF FIGURES**

GIT HISTORY


DeepakVishak / Sign-to-Text-Transcript
Pulse
Contributors
Community
Community Standards
Traffic
Commits
Code frequency
Dependency graph
Network
Forks
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday


DeepakVishak / Sign-to-Text-Transcript
Public
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
master
1 branch
0 tags
Go to file
Add file
Code
DeepakVishak commit
46 commits
Deep Learning
first commit
3 months ago
Documentation
commit
8 hours ago
Open CV Tutorial
first commit
3 months ago
Project v 1.0
first commit
10 days ago
Help people interested in this repository understand your project by adding a README.
Add a README
About
MCA minor project Realtime conversion of sign language to text
0 stars
1 watching
0 forks
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package
Languages
Jupyter Notebook 90.1%
Python 9.9%