

**Applied AI Spring 2025**

**Sign Language Recognition Software**

Mid Report



**Section:** 6A

**Group Members:**

Faizan Saleh 22L-7925

Syed Asfandyar Ali 22L-7945

National University Of Computer and Emerging Sciences

Department of Computer Science

Lahore, Pakistan

# Project Overview

Sign Language Recognition Software is an application designed to bridge communication gaps by communicating hand gestures captured via a camera and converting them into real-time text. The objective of this project is to develop a user-friendly system for recognizing both single characters and complete words in sign language. The project scope includes the development of a graphical user interface (GUI), creation of a custom dataset for complete words, implementation of a CNN for gesture classification. The timeline for the project is divided into initial setup and prototyping, model training and validation, and final integration and testing. Key stakeholders include the developers and potential end-users from the deaf and hard-of-hearing community.

# Progress Summary

## GUI Development:

A responsive GUI has been developed using Tkinter which is a python standard GUI toolkit. The interface includes functionalities for real-time gesture capture, dataset creation and display of text. Also used PIL (Python Image Library) to transfer real images from webcam to Tkinter. Images are rendered in mirror or flipped format to avoid true image format [5].

## OpenCV Integration and Threading:

OpenCV has been successfully integrated to facilitate real-time video capture and preprocessing of hand gestures. Computer Vision 2 library is used to capture images in resolution 480p in a non-HD webcam. Threading is utilized to capture images of each label separately to avoid performance and increase efficiency by reducing stutters.

## Dataset Creation:

The functionalities to create custom datasets has been developed and integrated which primarily focuses on complete words using Human Signalling Github Library [6]. It is used to label out a specific area in an image and convert that image into an XML formal which is later converted into an CSV file which can used to train the AI model.

## Model Implementation:

A CNN architecture has been selected and partially implemented for gestures classification. It has small artificial neurons which are filtered on our concerned area using object detection model library of Tensorflow 2.x and is used to train the AI model on these collective neuron’s information [7].

# Achievements and What Worked Well

## Custom Dataset:

The creation of custom datasets for complete words has worked well as we are able to easily capture large number of images for a single gesture from different directions that helps in the classification of gesture by CNN. Labels are captured through our GUI and then transferred to Human signaling software to generate an xml which later converted into an csv used to train our AI model.

## OpenCV Capturing:

The real-time video processing, which includes hand detection, operates efficiently within minimal latency and ensures smooth experience. Video Capture uses uuid to capture multiple images for 1 individual by choice to avoid collision while making use of threading to avoid stutters. Custom Webcams and mobiles can be used but will require change in original code.

# Challenges and What Did Not Work

## Learning Curve:

The project had a big learning curve as choosing the right model for this project was difficult due to our less exposure to the diverse range of models available for gesture classification which made the decision-making process complex. Project is beginner friendly but has enough guides to make things easy.

## Dataset Labeling issues:

Manual labelling of datasets using Human Signaling Repository proved time consuming. Many images are captured for a single label in a 5-second time frame. Due to Supervised Learning, all images are to labeled manually and filtered out which is a hectic and long task.

## TensorFlow Compatibility Issues:

We faced challenges with outdated TensorFlow libraries and API, and this step required extensive online search and troubleshooting of code from different online tutorials. Some issues still remain like not having any Nvidia CUDA GPU to train our AI on, So, we might have to rely on google collab or CPU performance to get a model on our custom dataset.

# Future Plans

## Model Training:

Complete the training of the CNN model from the Tensorflow object detection using the custom dataset and dataset from Kaggle. Kaggle doesn’t have sign language dataset for specific words but only letters for many languages. After we our done with this project we may upload it to help the community.

## Model and GUI integration:

Next integrate out trained model with the GUI such that we can observe real time gestures to text conversion. Trained Model is yet to be integrated with GUI as no model for our custom dataset available.

## Testing:

Conduct usability test to gather feedback and refine the system. Voice Feedback option is listed in our GUI to get reviews through voice recognition which will be implemented in the near future.

## Stretch Goals:

### Voice Output:

Implement text to speech functionality to provide auditory communication. Voice language recognition is only added as a dummy to Tkinter GUI and its functionality is yet to implemented on.

### Live Translation Display:

Show the translated text in a scrollable Text widget. Accumulate signs into meaningful sentences (with auto-spacing and punctuation if possible).

# Updated Expectations and Outcomes

Based on the current progress our expectations and outcomes remain unchanged due to our initial research which gave us a good understanding of the workings of tools for this project.

# References

1. TensorFlow API Documentation. Available: https://www.tensorflow.org/api\_docs
2. OpenCV Documentation. Available: https://docs.opencv.org/4.x/index.html
3. LabelImg Github Repository. Available: https://github.com/HumanSignal/labelImg
4. Nicholas Renotte’s YouTube Tutorials on Sign Language Recognition. Available: <https://www.youtube.com/@NicholasRenotte>
5. Python Image Library. Available: <https://pypi.org/project/pillow/>
6. Human Signal Repo. Available: <https://github.com/HumanSignal/labelImg/>
7. Object Detection Model. Available: <https://github.com/tensorflow/tensorflow>