## Team Name – Signtists

**Topic** - Finger Spelling Detection - American Sign Language

**Team Members –**

23M1074, 23M2164, 23M2160, 23M2155, 23M2161, 23M2163

## 1) Problem Statement/Abstract:

In the realm of machine learning, image classification holds significant importance due to its wide-ranging applications, including the development of systems for American Sign Language (ASL) Finger Spelling detection. This project aims to address the challenge of ASL Fingerspelling detection from images, which is crucial for facilitating communication between the hearing-impaired and the hearing community. The project primarily focuses on creating a classification model capable of recognizing 26 classes corresponding to the English alphabet and an additional class representing an empty frame. The ultimate goal, if time permits, is to extend this model to real-time classification.

Developing a robust and efficient model for the detection and classification of American Sign Language Finger Spelling gestures from images is the core objective of this project. The problem entails the creation of a machine learning model that can accurately classify each image as one of the 26 English alphabet letters or as 'nothing' when the image does not contain any sign. We aim to evaluate the performance of our custom model in comparison to the state-of-the-art You Only Look Once (YOLO) and Single Shot MultiBox Detector (SSD) models to gain insights into the strengths and weaknesses of our approach.

The project's significance lies in its potential to enhance accessibility and communication for the hearing-impaired community by providing a tool for real-time recognition of ASL Fingerspelling gestures. Additionally, the comparative analysis with YOLO and SSD models will allow us to identify which model performs better in this context, and why, contributing to a deeper understanding of the applicability of different detection and classification approaches in sign language recognition.

## 2) Proposed Solution Approach:

To tackle the problem statement, we plan to follow different approaches to build the custom model which detects the ASL letter. Our first approach involves constructing a traditional feed-forward neural network, designed to ingest key points or landmarks corresponding to hands, extracted from images, as input and predict the corresponding English alphabet according to ASL. As per the suggestions received, we have also planned to implement single shot detection methods to solve the problem at hand. So, our next approach explores the implementation of YOLO (You Only Look Once) and SSD (Single Shot Detector) models. These models will be

adapted to identify ASL alphabets. Lastly, we will conduct a comprehensive performance comparison among all the models developed in these three approaches.

## 3) Code Survey:

### 1) Mediapipe Python Framework

We are using the Mediapipe framework to detect the hand landmarks in a given image. The coordinates of the hand landmarks are then used as features to train the feed forward neural network as our first approach to the solution.

### 2) Feedforward Neural Network using Keras

## 4) Datasets:

To train the classification model, we are using the ASL Alphabet dataset from Kaggle which consists of data in 29 classes i.e., 26 classes corresponding to English Alphabets, one corresponding to nothing in the frame, one corresponding to sign representing space, and one corresponding to sign representing delete. Out of these 29 classes, we plan to use 27 classes (classes corresponding to all the alphabets along with nothing). As suggested in the feedback, we plan to create our own custom dataset to test the model.

## 5) Implementation details and progress:

### Approach 1 (Custom approach):

The first approach is to apply hand gesture detection as the first stage of the task. We use MediaPipe to detect the hand gesture. Mediapipe returns 21 hand landmarks for each hand, where each landmark is a set of (x, y, z) coordinates. The thing to be noted is that these are normalized coordinates. So, we flatten all the hand landmarks into an array. This becomes our neural network input. We pass this input through a feed-forward neural network classifier which has 27 classes (A-Z and nothing.) We trained this model on a subset of the actual training data. Our training set had roughly 9000 data points and the validation set had roughly 3000 data points. We fit the model for 100 epochs and the training and validation accuracy went over 99%.

Followed by that we tested it on a small test set with one example from each class and we are getting a test accuracy of 77%. We should take this with a pinch of salt. As of now, the model is incorrectly detecting 6 classes. To test it further, we looked at the real-time prediction using a webcam. The model does predict most of the classes well. But the stability is not good enough i.e., small rotations in hand or fluctuations are changing the prediction.

We are trying to make it robust to such changes by augmenting the training data by rotating it, flipping it and so on. Hopefully, that should increase the accuracy.


**Results obtained from the implementation of the preliminary model (FFNN Model). The corresponding detected letters are displayed on the top left corner.**

### Approach 2: YOLO approach

YOLO (You Only Look Once) is a popular real-time object detection algorithm used in computer vision and deep learning. The main difference of YOLO lies in its ability to divide an image into a grid and predict bounding boxes and class probabilities for objects within each grid cell in a single forward pass of the neural network. This approach significantly reduces computational complexity, making it one of the fastest object detection methods available.

Though YOLO offers real-time object detection it has some limitations. Its speed sacrifices localization accuracy for small and overlapping objects, leading to false positives and challenges with fine-grained details and aspect ratios. Training can be computationally intensive, and it may struggle with multi-scale objects. We aim to implement YOLO on pre-processed images and do a comparative study between YOLO and the proposed custom approach. The results of the comparative study will be presented in the final report.

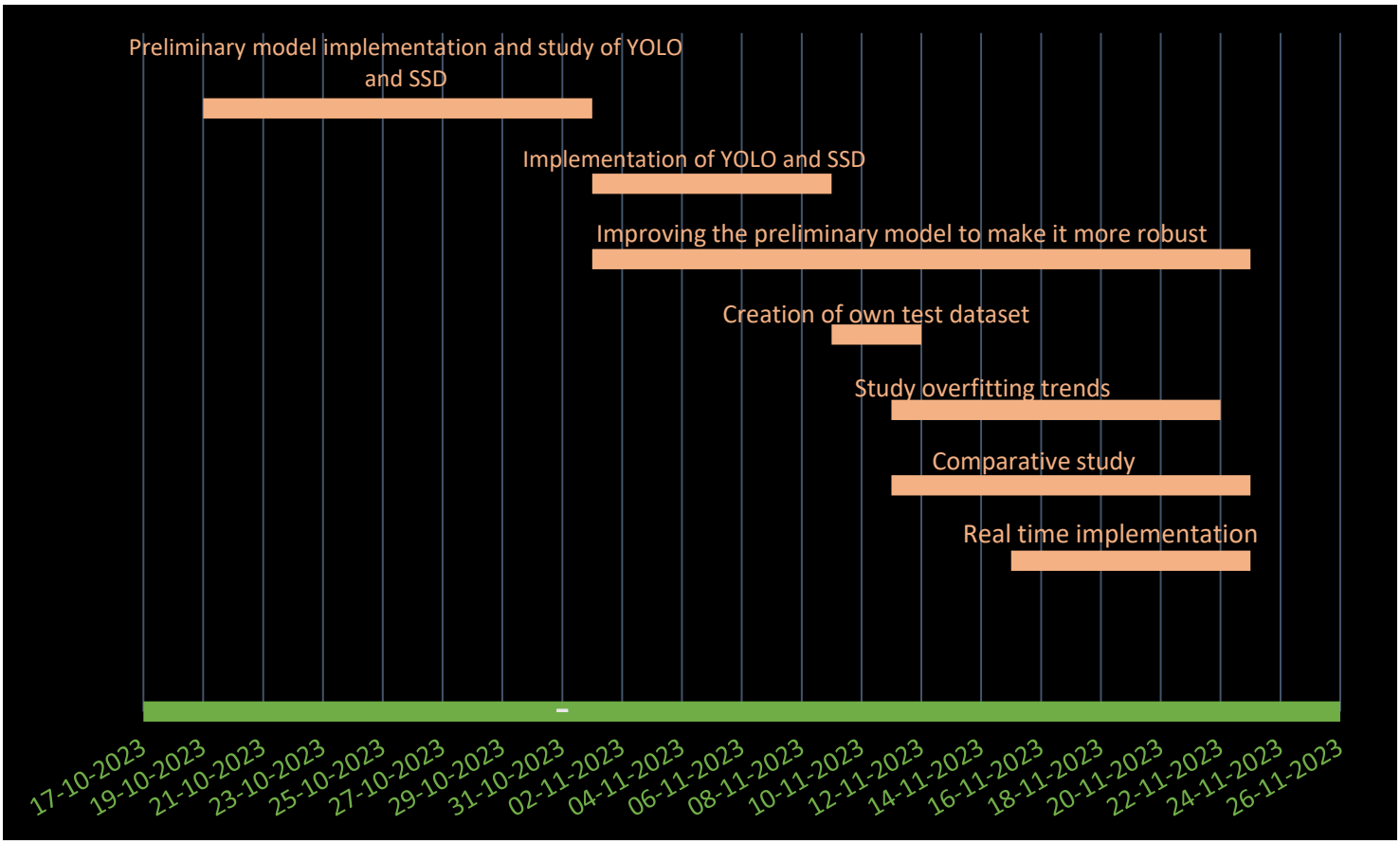
### Approach 3: SSD (Single Shot Detection) approach

SSD is easy to train and integrate into scenarios that require object detection. Also, it is a relatively simple method that completely eliminates proposal generation, and subsequent pixel or feature resampling stages and encapsulates all computation in a single network. SSD300 is one of the real-time detection methods that can achieve above 70% mAP with 59 frames per second. But SSD trade-off mAP with speed and confuses when the object is small or of different scale in the same image. We aim to implement SSD on pre-processed images and do a comparative study between SSD and the proposed custom approach. The results of the comparative study will be presented in the final report.

## 6) Roadmap:

The main goal of the project is to detect and classify the ASL fingerspelling gestures from images. The following are the planned tasks to be done in the remainder of the semester. Same is depicted in the figure below.

1) To make the custom approach robust to detect all alphabets, handle changes in orientation and positioning of the hand.
2) Creating our own test dataset.
3) Implement YOLO and SSD models with suitable preprocessing.
4) Study the overfitting trend in these models.
5) Compare the performance of the final model with that of YOLO and SSD.
6) Real time implementation.




**Project Timeline**