NORTH SOUTH UNIVERSITY

Real Time Bangla Sign Language Detection Using Deep Learning

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF
ELECTRICAL AND COMPUTER ENGINEERING
OF NORTH SOUTH UNIVERSITY
IN THE PARTIAL FULFILMENT OT THE REQUIREMENTS
FOR THE DEGREE OF BACHELOR OF SCIENCE IN
COMPUTER SCIENCE AND ENGINEERING

Date: 07 January, 2023

Declaration

It is hereby acknowledged that:

- No illegitimate procedure has been practiced during the preparation of this document.
- This document does not contain any previously published material without proper citation.
- This document represents our own accomplishment while being Undergraduate Students in the **North South University**

We declare that this CSE498R report entitled *Real Time Bangla Sign Language Detection Using Deep Learning* has not been accepted for any degree and is not concurrently submitted in candidature of any other degree. We would like to request you to accept this report as a partial fulfillment of Bachelor of Science degree under Electrical and Computer Engineering Department of North South University. Sincerely,

Student 1: Mosarrat Shazia Kabir
1831228042

Student 2: Syeda Karishma Naaz
1831270642

Student 3: Simon Uddin
1831858642

Approval

This is to certify that the CSE498R report entitled Real Time Bangla Sign Language Language Detection Using Deep Learning, submitted by Mosarrat Shazia Kabir (Student ID: 1831228042), Syeda Karishma Naaz (Student ID: 1831270642), Simon Uddin (Student ID: 1831858642) are undergraduate students of the Department of Electrical Computer Engineering, North South University. This report partially fulfils the require-ments for the degree of Bachelor of Science in Computer Science and Engineering on December 25, 2022, and has been accepted as satisfactory.

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation.

Dr. Md Shariful Islam
Assistant Professor
Department of Mathematics and Physics
North South University, Dhaka
Bangladesh.

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation.

Dr. Rajesh Palit
Professor & Chair
Department of Electrical Computer Engineering
North South University, Dhaka
Bangladesh.

Abstract

A Real Time Sign Language detector is a huge step towards creating a bridge between the deaf, dumb and the general population. Computer recognition of sign language begins with the acquisition of sign gestures and progresses to the generation of text/speech. There are two types of sign gestures: static and dynamic. Although static gesture recognition is easier than dynamic gesture recognition, both systems are useful to the human community. It is significant to demonstrate and implement such a system that can detect sign language using a convolutional neural network. To implement Transfer learning, we utilized a Pre-Trained SSD Mobile net V2 architecture, and YOLO (You only Look Once) version-5 trained on our own dataset. We created a robust classification system that consistently classifies sign language in the majority of instances. Furthermore, this strategy will be extremely beneficial to sign language learners in terms of sign language practice. Throughout the project, various human-computer interface methodologies for posture recognition were investigated and evaluated. The most effective approach was determined to be a series of image processing techniques with human movement classification. The dataset consists of 50 different classes, including Bangla Alphabets, Numbers, Bangladesh, Dhaka, and Salam. Without a controlled background and low light, the system can recognize selected Sign Language signs with an accuracy of 88-93% using SSD Mobilenet V2 model and 90%–98% using YOLO V5 Algorithm.

Contents

List of Figures

List of Tables

Acknowledgements

We would like to express our special Thanks of gratitude to *Dr. Md Shariful Islam* Sir who gave us the opportunity to do the wonderful project on the DIRECTED RESEARCH (CSE498R), which helped us in Research and we came to know about so many new things. We are really thankful for him. It is his guidance and patience that led us to envision our project *"Real Time Bangla Sign Language Detection"* to be a full-fledged solution for a blind people.

Glossary

Deep Learning :
Deep learning is the subset of machine learning is composed of algorithms that permit software to train itself to perform tasks, like speech and image recognition, by exposing multi-layered neural networks to a vast amount of data.

Object Detection:
Object detection is the process of locating objects in images or videos using computer vision techniques. An object detection algorithm typically utilizes machine learning or deep learning to produce meaningful results.

Tensorflow:
TensorFlow is an open-source framework developed by Google researchers for performing machine learning, deep learning, and other statistical and predictive analytics tasks.

Feature Extraction:
Feature extraction refers to the process of converting raw data into numerical features that can be processed while conserving the information in the initial data set. It yields better results than applying machine learning directly to raw data.

CNN:
Convolutional Neural Network or CNN is a type of artificial neural network that is widely used for image/object recognition and classification. Deep Learning recognizes objects in an image using CNN.

SSD Mobilenet:
The mobilenet-ssd model is a Single-Shot multibox Detection (SSD) network that performs object detection. It is a model typically deployed on low compute devices such as mobile with high accuracy performance.

YOLO:
YOLO (You Only Look Once) is a regression-based algorithm that predicts classes and bounding boxes for the whole image in one run of the algorithm.

1 Introduction

1.1 Sign-Language

Sign languages are a mode of communication that is mostly used by deaf or mute people. In sign language, hand gestures and movements, body language, or facial expressions are incorporated to convey meaning. Languages in this mode are not universal. People from different cultures and countries use different sign languages [1]. Around the world, there are more than 300 sign languages [2]. In sign language, gestures or symbols are organized linguistically. Sign language is a bridge between the deaf, mute and common people. Though many people think sign languages are not real languages, in linguistic terms, they are as rich and complex as any spoken language. Professional linguists have discovered that sign language exhibits the same characteristics as all languages [3].

1.2 Sign Language Detection

Together with the rapid advancements in information and technology, the interaction between humans and computers has also changed in the modern world. There are more than 430 million people with speech and hearing disabilities worldwide, which is more than five percent of the world's population [4]. A sign language detection project provides assistance for these people. The sign language detection project uses a web camera to capture hand gestures, label those images, and then train those labeled images with the SSD Mobile Net algorithm. In order to get successful sign recognition three steps must be followed: 1. Capture the user's hand gestures or signs. 2. Classify each frame in real time. 3. Get the classification score and display the most likely sign [5].

1.3 Review of hand-gesture for sign-language recognition

Human life is rendered easier by hand gesture recognition, which serves as a key to overcoming many difficulties. An extensive range of applications can be developed using the ability of machines to understand human activities and their meanings. Recognition of sign languages is one of the specific fields of interest [6]. Sign language recognition uses methods like identifying hand motions for different signs and segmenting hands from the background. This is done to forecast and string them into sentences that are both semantically correct and meaningful. The accuracy of a model is influenced by its background and environment. Gestures can be recognized in many ways, including vision-based and sensor-based systems. The vision-based approach is a prebuilt solution that is ready to configure and deploy using image and video footage of hand gestures. A sensor-based system, on the other hand, consists of multiple sensors that are aggregated as a single component that detects various parameters such as location, velocity, and trajectory [5]. There are a few steps to follow to develop a sign language recognizer:

1.3.1 Camera access and capturing image

This sign language model is based on frames obtained by a web camera on a laptop. Different sign language symbols were captured from a variety of angles and in a variety of lighting conditions in order to improve their accuracy. The images were processed using the OpenCV python library.

1.3.2 Segmentation

From these captured images, a specific region is selected that has the hand gesture of a sign language symbol. The goal is to predict that symbol. Images of the sign are surrounded by tight bounding boxes. With the LabelImg tool, all the different hand gestures were assigned specific names and labelled.

1.3.3 Classification

When it comes to computer predictions, machine learning is used. There are four ways to approach machine learning: supervised, semi-supervised, unsupervised, and reinforce-

ment learning. When it comes to supervised machine learning algorithms, training data is labeled and output data is specified. However, in unsupervised machine learning, the algorithm is trained on an unlabeled dataset and the outcome is predetermined [7]. The proposed model for sign language recognition utilizes supervised machine learning. This is because all of the images in the dataset have been labeled, and also because all results have been specified.

1.4 Related works

1.4.1 Real-time Sign Language detection

Research on sign language detection has been done in New Delhi, India, to improve the communication between the deaf and general folk. They showcased their creation of a sign language recognition model based on Convolutional Neural network (CNN) technology. They created a dataset that has over 2000 images. Their dataset has a total of 5 classes and 400 images for each class. The symbols were Hello, Yes, No, I love you, and Thank you. They trained their dataset using SSD (Single Short Detection) Mobile net V2 architecture. It was a robust model that can consistently classify Sign language in the majority of cases. Since they used the Mobile net SSD model which is a single-shot multibox detection network, their model could scan the pixels of every image that were inside the bounding box coordinates and class probabilities to conduct detection. They used Tensor-flow object detection API that includes the SSD Mobile Net model. Their system was able to recognize selected sign language signs with an accuracy of 70 to 80 % without a controlled background with a small light [5]. The accuracy table for the 5 classes:-

Gesture Name	**Accuracy (%)**
Hello	91.0
Yes	88.7
No	88.6
Thank You	84.1
I Love You	82.4

Table 1.1: The Accuracy Table

1.4.2 Sign Language Recognition : State of the Art

Sign Language is mainly a gestural language used by the blind and deaf. Three-dimension spaces and hand movements are used to exchange messages. Computer recognition of sign language deals with sign gesture acquisition and continues to text/speech generation. Computerized digital image processing and a wide variety of classification methods are used to recognize the alphabet flow and interpret sign language words and phrases. Four essential components in a gesture recognition system are – gesture modeling, gesture analysis, gesture recognition, and gesture-based application system. In this paper, the data acquisition, data processing, transformation, feature extraction, classification, and results are examined. The result includes parameters like dataset size, methods accuracy, etc.

Sign Language	**Dataset size**	**Method**	**Accuracy (%)**
American Sign Language	20	ANN	92.33
Vietnam Sign Language	23	Fuzzy rule-based system	100
Ukrainian Sign Language	85	Hidden Markov Model	94
American Sign Language	26	Combinational neutral network	100
Arabic Sign Language	3450	KNN	87
Lankan Tamil Sign Language	300	Artificial Neural Network	73.76

Table 1.2: The Accuracy Table

1.4.3 Sign Language Recognition with Tensor Flow

Communication is the way to express or exchange information, ideas, or feelings. To establish a commute between two or more people. Sometimes it becomes difficult to understand the meaning of sign language for normal people. The author has created an Indian sign language dataset by using a webcam and then using transfer learning, trained a TensorFlow model to create a real-time Sign language recognition system. They used the pre-trained COCO 2017 dataset which was used in SSD Mobile Net v2 320X32. It has approximately 650 images in total, 25 images for each alphabet. The method that has been used for creating SLR systems is – Hidden Markov Model (HMM). The various HMM that have been used are Multi-Stream HMM (MSHMM) which is based on two standard single sstreamHMM, light. Models are Naïve Bayes classifier, Multiplayer per-

ceptron, Unsupervised neural network self-organizing map, self-organizing feature map, and Simple Recurrent Network. Self-design methods have also been used such as the wavelet-based method and Eigen Value Euclidean distance. The result of the system is

Model	**Accuracy (%)**
The light-HMM	83.6
Multi stream HMM	86.7
SVM	97.5
Eigen Value	97
Wavelet Value	100

Table 1.3: The Accuracy Table

based on confidence rate and the average confidence rate of the system is 85.45 percent[9].

1.4.4 Sign Language Recognition using deep learning

The IISL2020 dataset of diverse hand motions is used in the article to demonstrate Indian Sign Language recognition using LSTM and GRU. The suggested model performs better than any other ISL model that is currently available for terms like hello, good morning, and work. Furthermore, adding more LSTM and GRU layers and applying LSTM before GRU further improves the model's ability to determine the ISL. Future research can create various datasets under ideal circumstances, and model accuracy can be increased by rotating the camera or utilizing a portable device. As it is, the constructed model only considers the viewpoint of a single character. This strategy can be used in interpretation, particularly in the case of the ISL of continuous sign language leading to syntactic generation [10].

2 Methodology

2.1 Data set

The dataset we used for the sign language detection project was built by ourselves. We used our laptop's webcam to click images and make annotations. Each class has 10 images. We have 500 images in the dataset, and there are 50 classes. Among the 50 classes, 36 are from the bangla alphabet, and 11 are from numbers, Bangladesh, Dhaka, and Salam. The labels for each class are given below:

Bangla Alphabets	Labels	Bangla Alphabets	Labels	Bangla Alphabets	labels
অ	A	ঢ	DH	ঁ	Chondrobindu
আ	AA	ন	N	ং	Aonishor
ই	I	ত	T-O	ঃ	Bidhorgo
উ	U	থ	TH-O	বাংলাদেশ	Bangladesh
এ	E	ম	M	ঢাকা	Dhaka
ও	O	ধ	DH-O	সালাম	Salam
ক	K	প	P		
খ	KH	ফ	PH		
গ	G	ব	B		
ঘ	GH	ভ	BH		
চ	C	য়	E-O		
ছ	CH	র	R		
জ	J	ল	L		
ঝ	JH	স	S		
ট	T	হ	H		
ঠ	TH				
ড	D				

English	অঙ্ক
Zero	০
One	১
Two	২
Three	৩
Four	৪
Five	৫
Six	৬
Seven	৭
Eight	৮
Nine	৯
Ten	১০

Figure 2.1: Bangla Alpahabets English Spelling

2.2 Data Collecting

We collected our data using OpenCV tools. OpenCV liberty, time, and uuid liberty are imported at the beginning, as well as the path to save images. Once the image frame was set, we activated our webcam and started taking photos.

In [2]:
```
import cv2 # open cv
import os
import time
import uuid
```

In [3]:
```
IMAGES_PATH = 'Tensorflow/workspace/images/collectedimages'
```

In [5]:
```
labels = ['bangladesh', 'Dhaka', 'salam']
number_imgs = 10
```

In [8]:
```
for label in labels:
    !mkdir {'Tensorflow\workspace\images\collectedimages\\'+label}
    cap = cv2.VideoCapture(0)
    print ('Collecting images for {}'.format(label))
    time.sleep(5)
    for imgnum in range(number_imgs):
        ret, frame = cap.read()
        imgname = os.path.join(IMAGES_PATH, label, label+''.''jpg'.format(str(uuid.uuid1())))
        cv2.imwrite(imgname, frame)
        cv2.imshow('frame', frame)
        time.sleep(2)

        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
    cap.release()
```

```
A subdirectory or file Tensorflow\workspace\images\collectedimages\bangladesh already exists.
Collecting images for bangladesh
```

Figure 2.2: Collecting Images

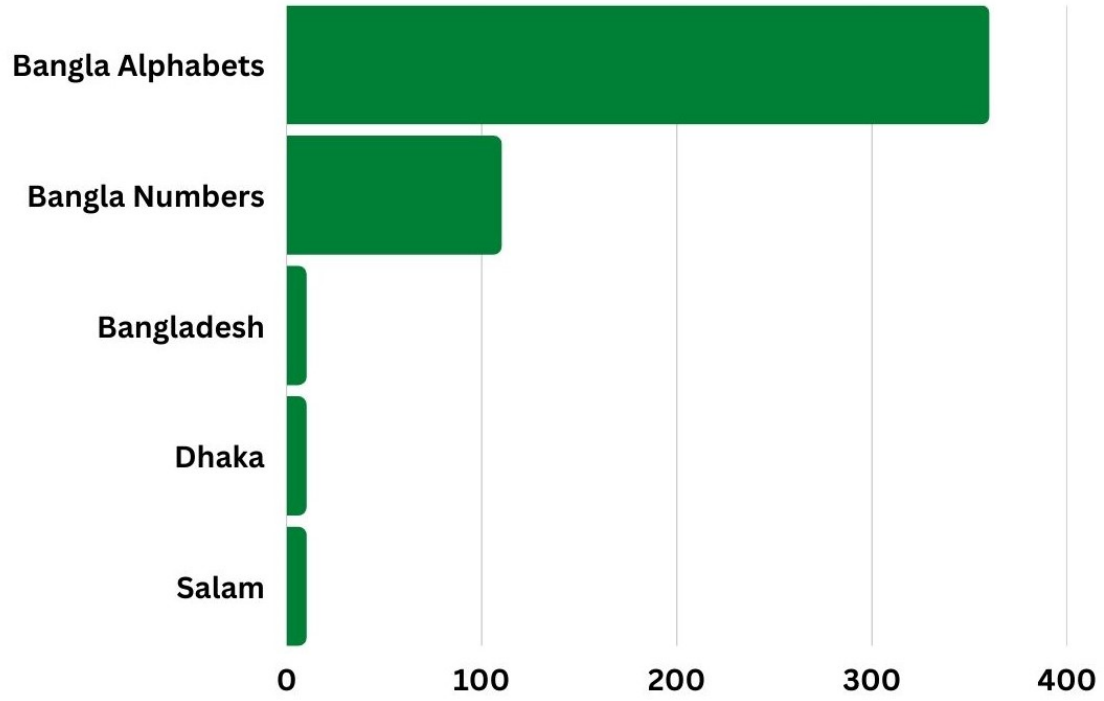


Figure 2.3: Image Distribution

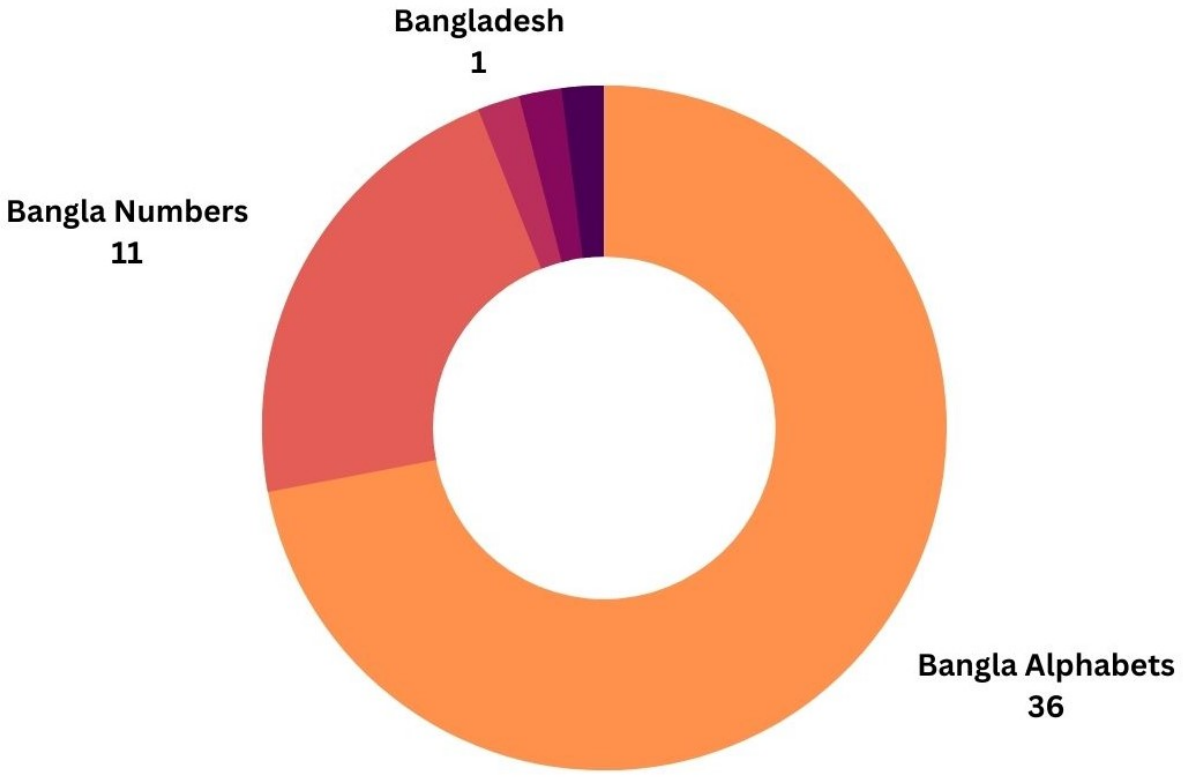


Figure 2.4: Image Classes

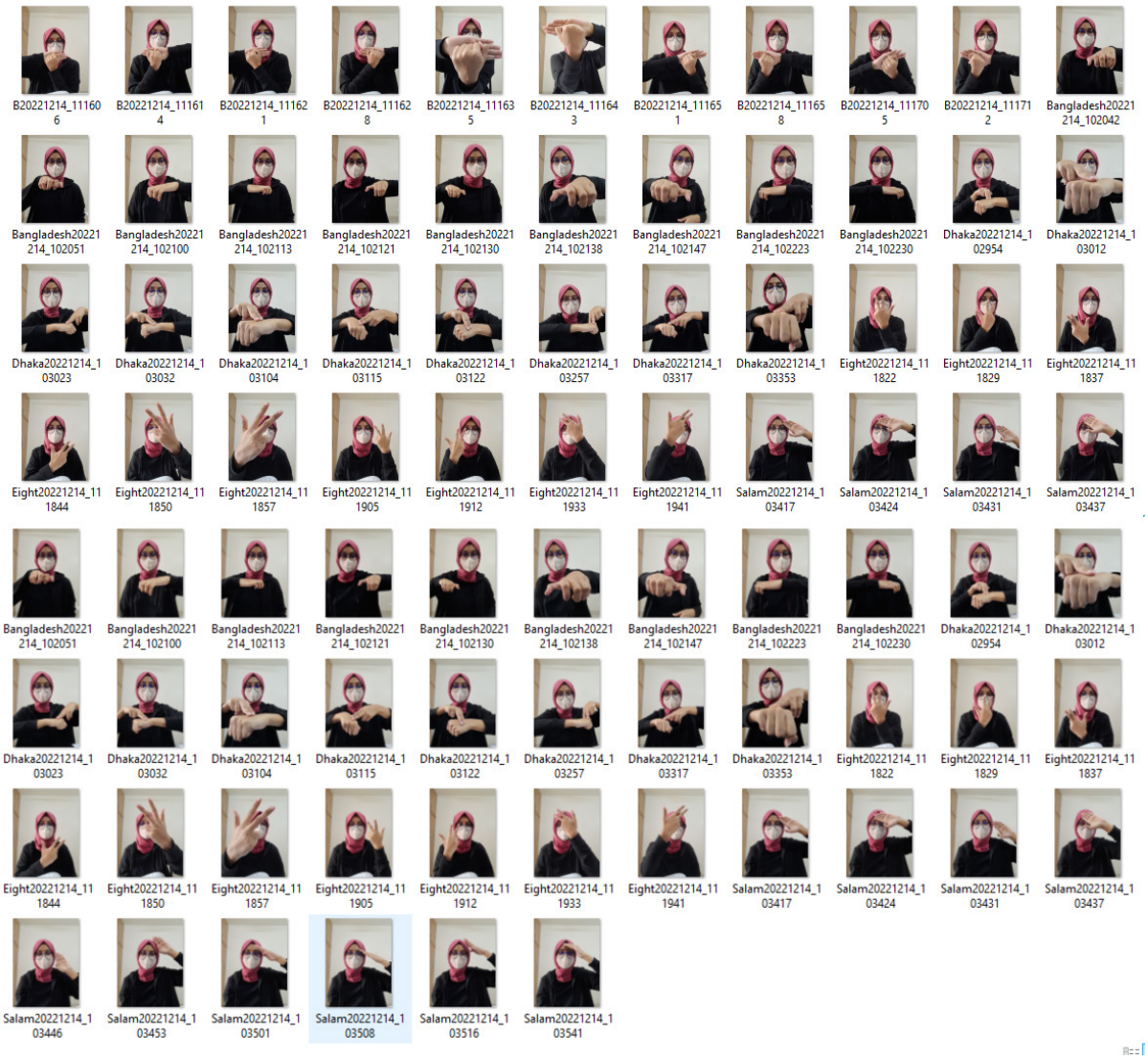


Figure 2.5: Collected Images

2.3 Data Labelling and Annatotations

As soon as we have collected all of our images, we will begin processing them. Using the LabelImg tool, the next step in the process was to label each of the images.

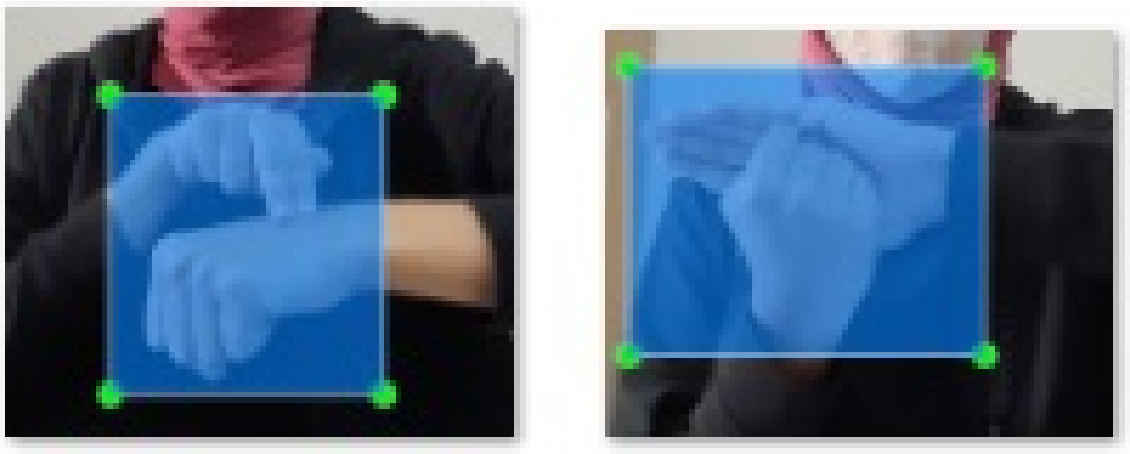


Figure 2.6: Labeling The Images

In this process we annotated all the images, we set bounding boxes around each sign and set the label.

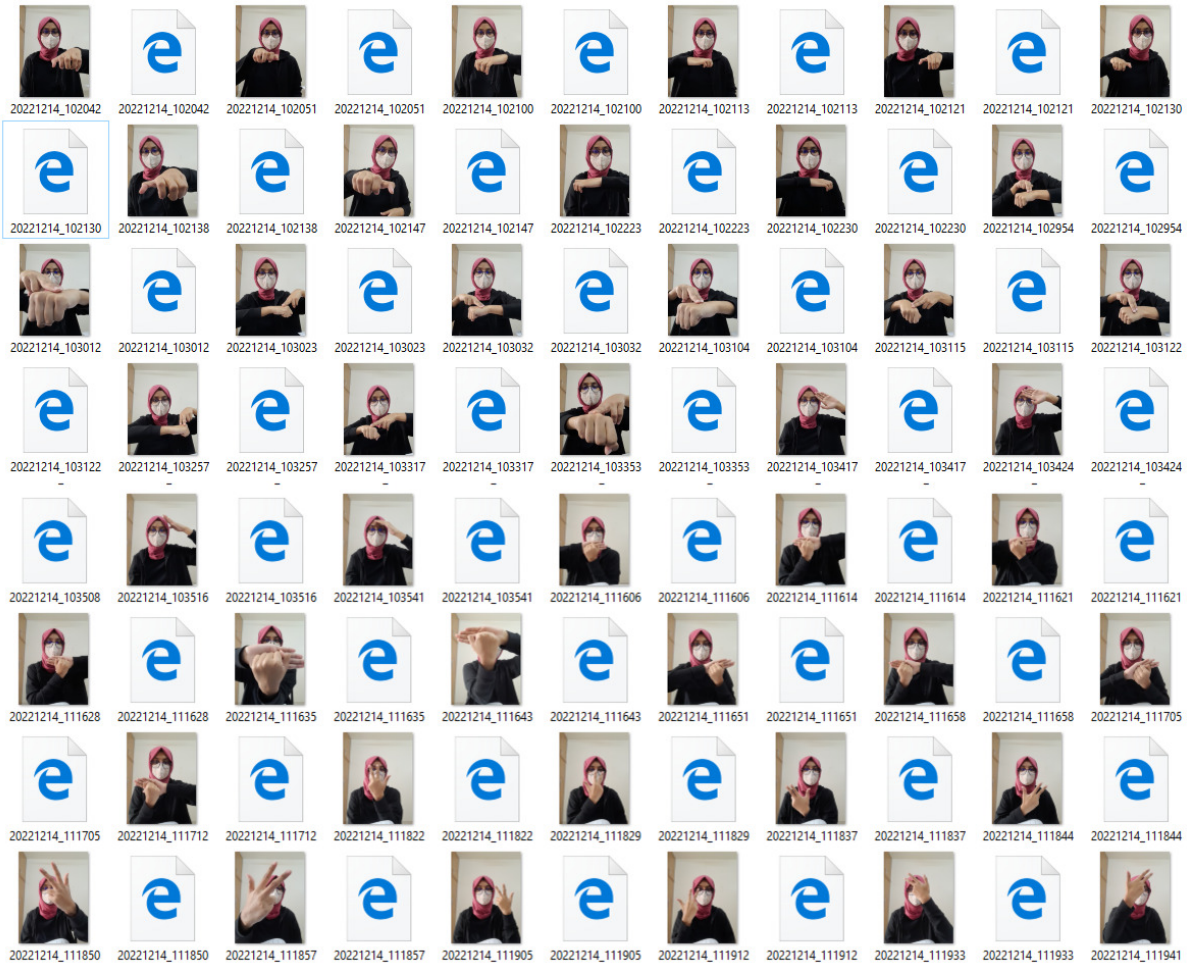


Figure 2.7: Annotated Images

The annotation work was done in two steps. First we annotated the images in PascalVOC format. Next we annotated images in YOLO format.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
- <annotation>
    <folder>collectedimages</folder>
    <filename>20221214_102042.jpg</filename>
    <path>C:\Users\User\RealTimeObjectDetection\Tensorflow\workspace\images\collectedimages\20221214_102042.jpg</path>
  - <source>
      <database>Unknown</database>
    </source>
  - <size>
      <width>1980</width>
      <height>2640</height>
      <depth>3</depth>
    </size>
    <segmented>0</segmented>
  - <object>
      <name>Bangladesh</name>
      <pose>Unspecified</pose>
      <truncated>0</truncated>
      <difficult>0</difficult>
    - <bndbox>
        <xmin>764</xmin>
        <ymin>1188</ymin>
        <xmax>1816</xmax>
        <ymax>1988</ymax>
      </bndbox>
    </object>
</annotation>
```

Figure 2.8: PascalVOC Formate Annotation

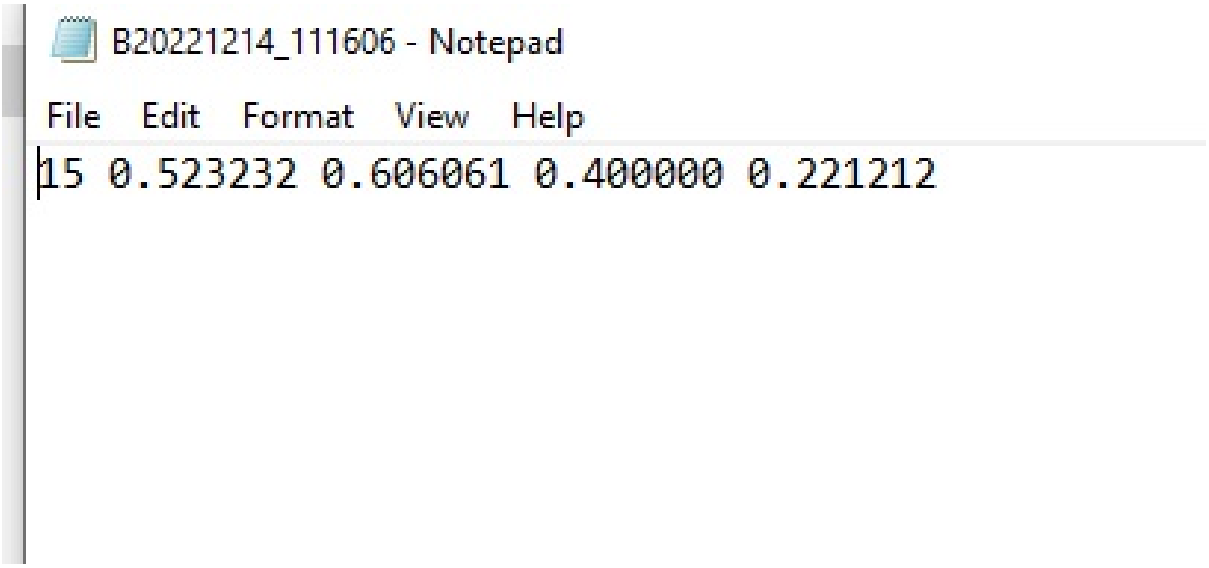


Figure 2.9: YOLO Formate Annotation

2.4 Application of Deep Learning

A discrete branch of artificial intelligence can be used to recognize sign language with its machine learning capabilities. This is a method of computation that utilizes data and algorithms to yield a high degree of accuracy. Deep learning is a subset of machine learning that uses vast volumes of data and complex algorithms to train a model. Deep learning CNN (Convolutional Neural Network) techniques are applied to detect sign language. To examine performance, two of the most well-known and successfully applied CNN techniques were applied to the analysis. The selected techniques are:

- SSD Moblilenet v2
- YOLO v5

All the techniques were tested using the default parameters to obtain the optimum results. To train our model we needed to install the necessary libraries like numpy, tensorflow and matplotlib etc. The developing environment was Google Collab and the runtime type was GPU.

We split our data into 3 segments- train, test and validation. This train, test, and validation split ratio was 60:20:20.

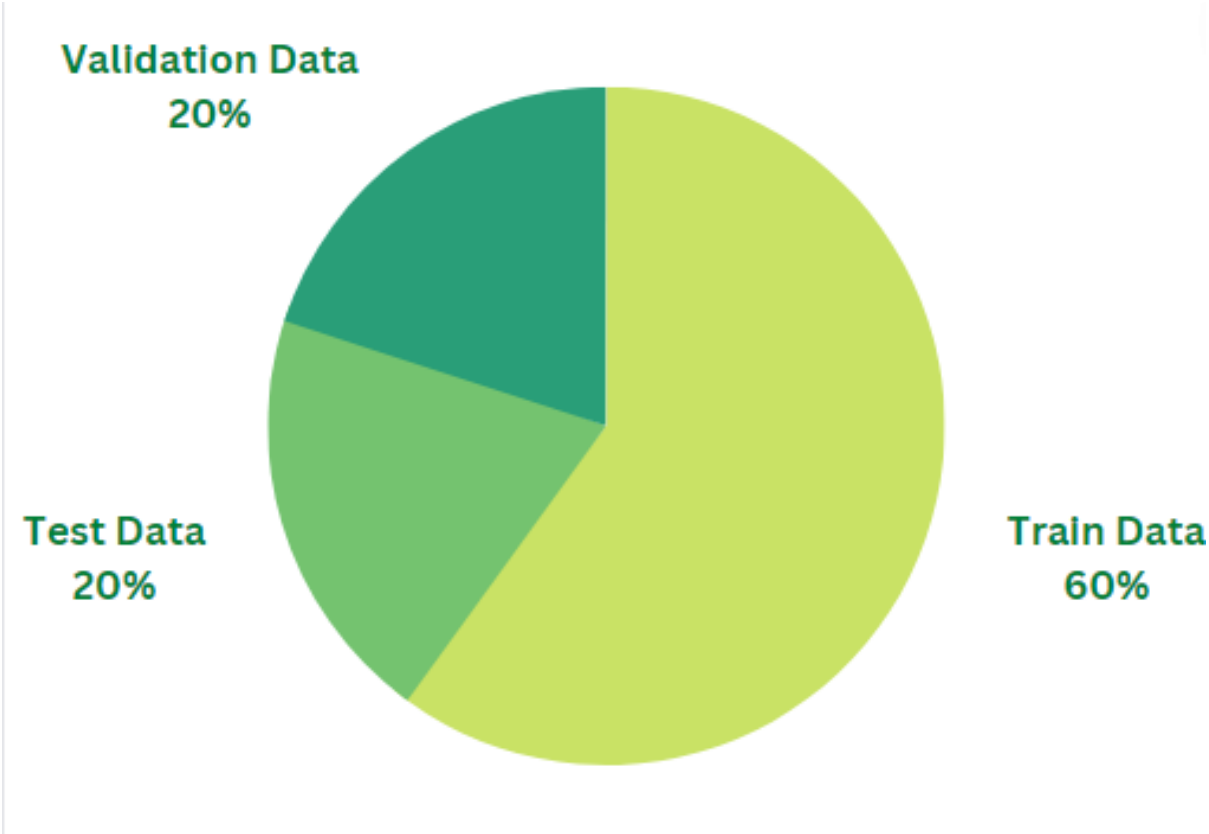


Figure 2.10: Spliting the Data

At first we trained our model with SSD Mobilenet v2; for this we had to install

Tensorflow and Tensorflow GPU. As a next step, we used YOLO v5 - we downloaded the git repository and trained it with our custom dataset. While training we again segmented our training set. First we trained with 50 images, then 100, 200 and lastly 500.

Finally, accuracy of the trained models were ploted into graph to determine the performance of the methods and techniques.

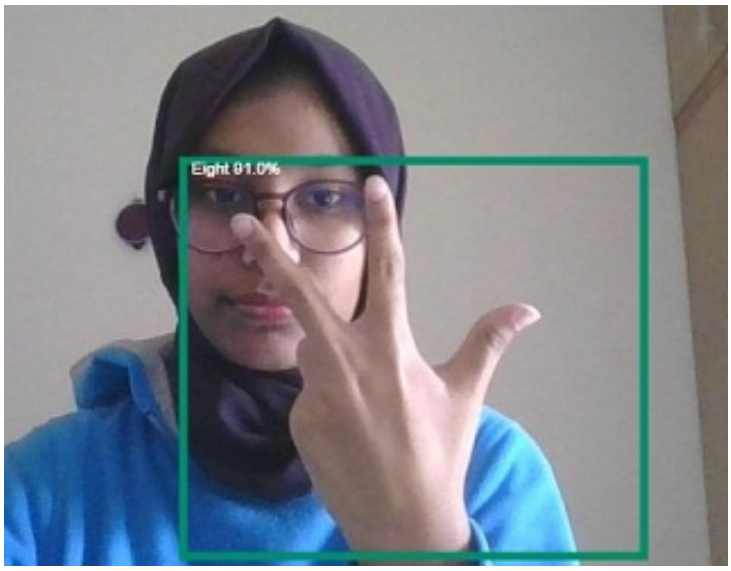
3 Results and Analysis

3.1 For SSD Mobilenet version-2:

The Mobile Net SSD model is a single-shot multibox detection (SSD) network that examines the pixels of an image that are within the bounding box coordinates and class probabilities to perform object detection. The model's architecture is based on the concept of an inverted residual structure, as opposed to traditional residual models. In this scenario, the input and output layers of the residual block are bottlenecked. Additionally, nonlinearities in intermediate layers are minimized, and a lightweight depthwise convolution is employed. This model is included into the TensorFlow object detection API. There was a range of accuracy of 88% to 93% for this real-time sign language detection model.

Number of Used Image For Training	**True Result**	**False Result**	**Accuracy%**
50	24	26	48
100	54	46	54
200	144	56	72
500	435	65	87

Table 3.1: The Accuracy Table for SSD Mobilenet v-2



(a) SSD Mobilenet v2

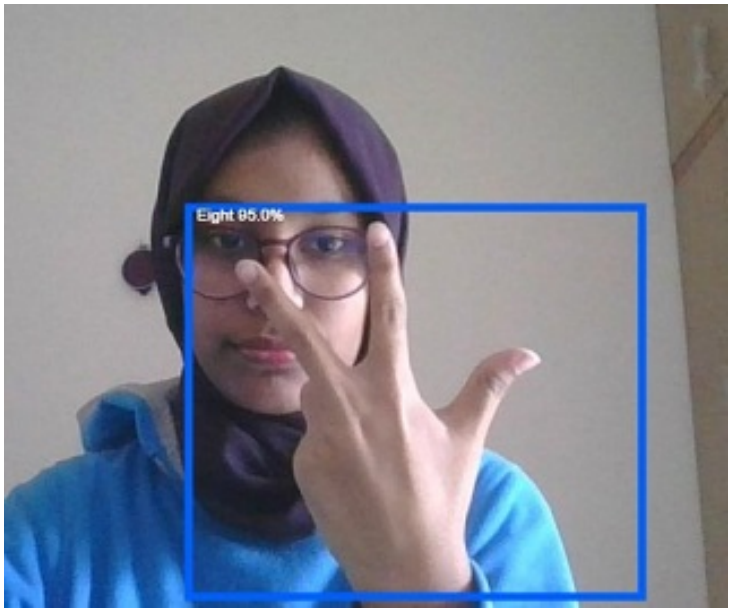
Figure 3.1: Detecting Sign Language using SSD Mobilenet v2 in Real Time

3.2 For YOLO version-5:

YOLO is an abbreviation for the term 'You Only Look Once'. This is an algorithm that identifies and recognizes different objects in a photograph (in real-time). YOLO performs object detection as a regression problem and returns the class probabilities of the recognized images. To detect objects in real-time, the YOLO algorithm employs convolutional neural networks (CNN). To detect objects, the algorithm requires only one forward propagation through a neural network, as the name implies. This means that the entire image is predicted in a single algorithm run. CNN is used to predict multiple class probabilities and bounding boxes at the same time. As a result of this real-time sign language detection model, there was a range of accuracy of 90% to 98% using the YOLOv5 algorithm.

Number of Used Image For Training	**True Result**	**False Result**	**Accuracy%**
50	24	26	48
100	56	43	56
200	152	48	76
500	446	54	89.2

Table 3.2: The Accuracy Table for YOLO Algorithm version-5



(a) YOLO v5

Figure 3.2: Detecting Sign Language using YOLO v5 in Real Time

4 Conclusion

4.1 Discussion and conclusion

The primary objective of a sign language detecting system is to facilitate communication between able-bodied and deaf individuals through the use of hand gestures. The suggested system is accessible through webcam or any other built-in camera that detects and processes the indicators for recognition. We may conclude, based on the model's outcome, that the suggested system can provide precise results under controlled light and intensity. Additionally, it is simple to add custom gestures, and photographs captured from a variety of angles and frames will increase the model's precision. Increasing the dataset makes it simple to scale up the model to a huge size. The model is limited by environmental constraints such as low light intensity and an unmanaged background, which reduce the detection's accuracy. Consequently, we will next endeavor to eliminate these defects and expand the dataset for more precise results.

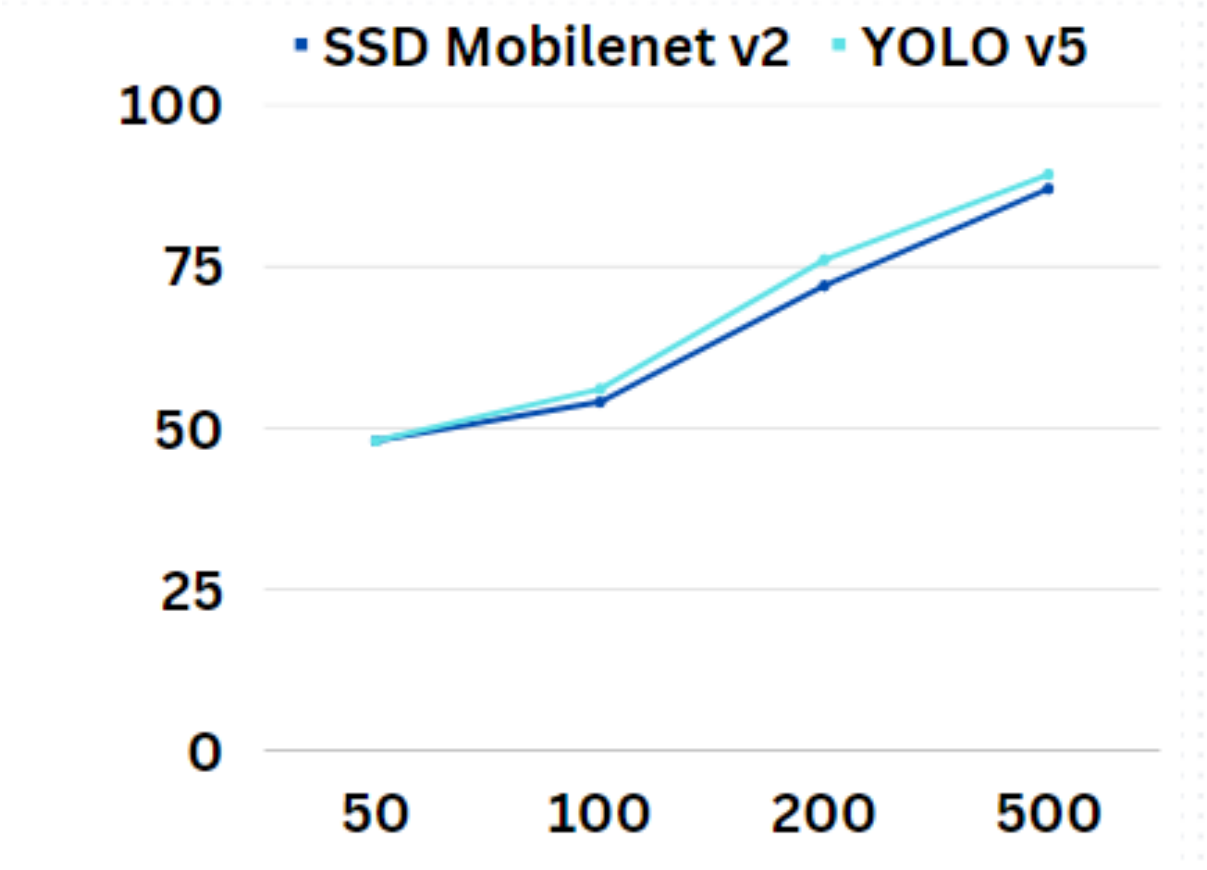
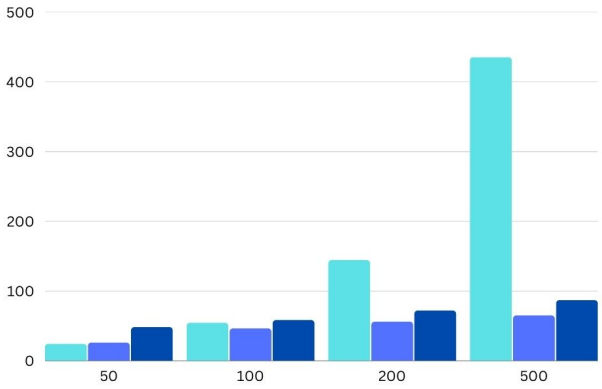
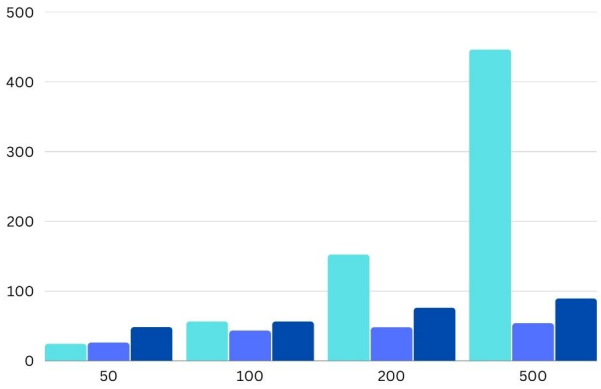


Figure 4.1: Accuracy comparison between SSD Mobilenet v2 and YOLO v5

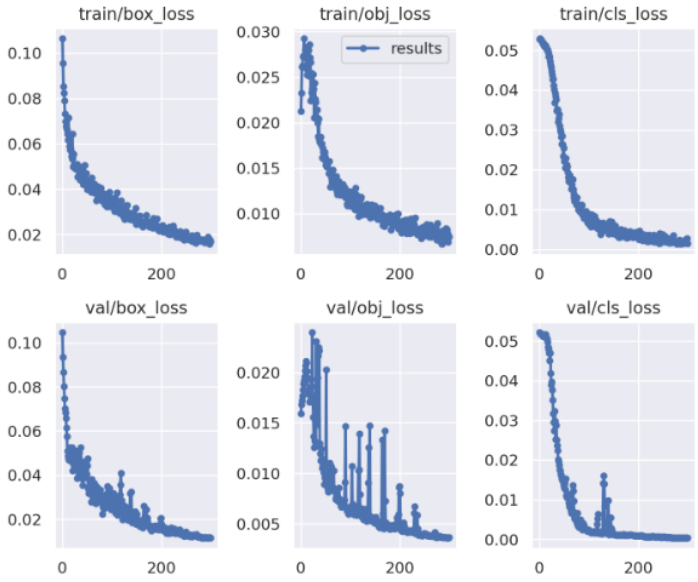


(a) Evaluation of SSD Mobilenet v2

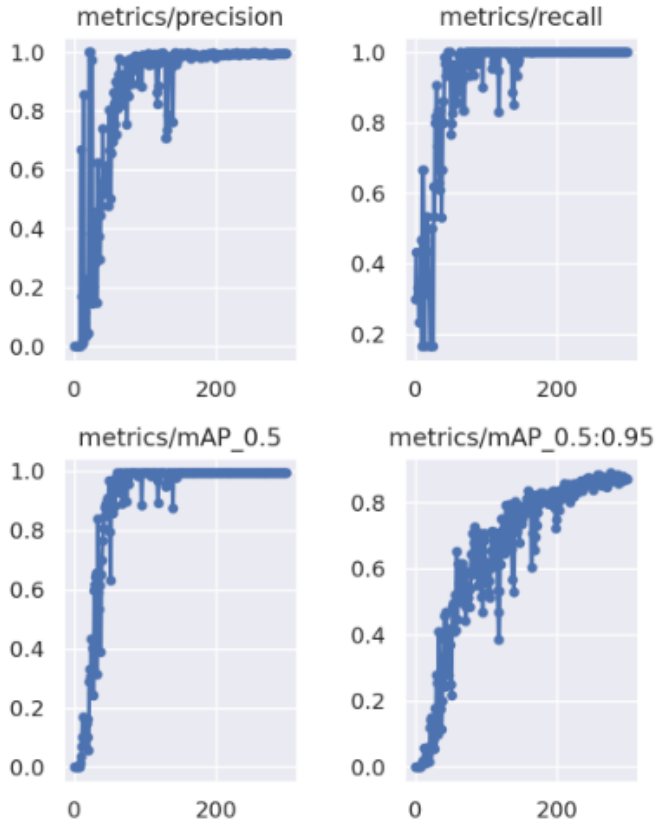


(b) Evaluation of YOLO v5

Figure 4.2: Comparison between SSD Mobilenet v2 and YOLO v5



(a) Evaluation of losses



(b) Evaluation of Accuracy

Figure 4.3: Data loss and Accuracy

4.2 Future Work

Currently, in this study, we are only analyzing Bangla Alphabets, Numbers (0-10) and three other words. We are planning on implementing this project in the future with Bangla words and phrases, in order to make our system capable of making sentences. There is also the possibility of using the same Bangla Sign Language Detection model alone in the field of Expression Recognition.

References