MAJOR PROJECT REPORT – PHASE I

On

Real Time Sign Language Recognition

Using Transfer Learning

submitted to

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

by

Pavan Karthik	19241A05L6
Pradyumna Sinha	19241A05M0
Jashwanth Naidu	19241A05J4
Anand Thota	19241A05M6

under the esteemed guidance of

Dr. Lipika Goel
Associate Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Gokaraju Rangaraju Institute of Engineering and Technology

(Autonomous)

Bachupally, Hyderabad, Telangana

GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY
(Autonomous)
Bachupally, Hyderabad, Telangana - 500090

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the mini project titled "**Real Time Sign Language Recognition using Transfer Learning**" is a bonafide work done by **Pavan Karthik** (19241A05L6), **Pradyumna Sinha** (19241A05M0), **Jashwanth Naidu** (19241A05J4), **Anand Thota** (9241A05M6) in partial fulfillment for the award of Bachelor of Technology in Computer Science and Engineering of the Jawaharlal Nehru Technological University Hyderabad, Hyderabad and that this work has not been submitted for the award of any other Degree/Diploma of any Institution/University.

Section Project Coordinator Project Guide

DECLARATION

We hereby declare that the project titled **"Real Time Sign Language Recognition using Transfer Learning"** is original and bonafide work of our own in the partial fulfillment of the requirements for the award of Degree of Bachelor of Technology in Computer Science and Engineering, submitted to the Department of Computer Science and Engineering, GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY(Autonomous) Hyderabad under the guidance of **Dr. Lipika Goel,** Associate Professor and has not been copied from any earlier reports/works.

Pavan Karthik	19241A05L6
Pradyumna Sinha	19241A05M0
Jashwanth Naidu	19241A05J4
Anand Thota	19241A05M6

Pavan Karthik 19241A05L6	Pradyumna Sinha 19241A05M0	Jashwanth Naidu 19241A05J4	Anand Thota 19241A05M6

ACKNOWLEDGEMENT

We are very grateful to our Project guide **Dr. Lipika Goel**, Professor, Department of Computer Science and Engineering, Gokaraju Rangaraju Institute of Engineering and Technology (Autonomous), for his extensive patience and guidance throughout our project work.

We are very grateful to our Project Co-ordinator **Ms.K.Surekha,** Assistant. Professor, Department of Computer Science and Engineering, Gokaraju Rangaraju Institute of Engineering and Technology (Autonomous), for extending his support and assisting us throughout our project work.

We would like to thank **Dr.K.Madhavi**, Head of the Department, Computer Science and Engineering, Gokaraju Rangaraju Institute of Engineering and Technology (Autonomous), for having provided the freedom to use all the facilities available in the department, especially the laboratories and library.

We also express a deep sense of gratitude to our **Director Sir** & **Principal Sir**, for having provided us with adequate facilities to pursue our project and for providing us the conducive environment for carrying through our academic schedules and projects with ease.

We sincerely thank all the staff of the Department of Computer Science and Engineering, for their timely suggestions, healthy criticism, and motivation during the course of our project work.

Finally, we express our immense gratitude with pleasure to all who have either directly or indirectly contributed to our need at right time for the development and success of our project work

Pavan Karthik	**19241A05L6**
Pradyumna Sinha	**19241A05M0**
Jashwanth Naidu	**19241A05J4**
Anand Thota	**19241A05M6**

ABSTRACT

Communication is very crucial to human beings, as it enables us to express ourselves. We communicate through speech, gestures, body language, reading, writing, or through visual aids, speech being one of the most commonly used among them. However, unfortunately, for the speaking and hearing-impaired minority, there is a communication gap.

Sign language is learned by the deaf and mute and usually it is not known to normal people. It strikes our mind to bridge the gap between hearing impaired and normal people & make communication easier.

To overcome this barrier, we propose a method where we collect sign language gestures using a webcam and by using **Transfer learning**, we train a **Tensor Flow model** to create a **Real-time Sign Language Recognition system**. This will help a lot of people in communicating with deaf and mute people.

Table of Contents

Chapter No:	TITLE	Page No
	Certificate **Declaration** **Acknowledgment** **abstract**	**2** **3** **4** **5**
1	**Introduction**	**7**
	1.1 Existing System	7
	1.1.1 Limitations in Existing System	7
	1.2 Proposed System	8
	1.2.1 Advantages over Existing Systems	9
2	**Literature Survey**	10
3	**Technology Used**	11
4	**System Specifications**	15
	4.1 Scope of the Project	15
	4.2 System Requirement Specification	15
5	Output	17
6	**References**	**20**

1. Introduction

In our project, we are introducing a Real Time Sign Language Detector Using a Transfer Learning methodology. In this, we run the code in Jupyter notebook and by using our webcam we will be capturing the hand gesture images and we will be detecting and displaying its correct sign language meaning in real-time (i.e. in live video). It plays a significant role in improving communication between the general people and deaf and mute people

Sign Language is used by more than 350 million people across the world. Hence, by introducing our model, Mute people can use gestures and we will be able to detect their language and understand them for communication.

1.1 Existing System

There are mainly two kinds of existing systems :

1. Data Glove approach: In this approach, we will be using various kinds of sensors and would attach them to the glove then by obtaining numeric input parameters such as accelerometer readings, and gyro meter readings we would be training the model.

2. Deep Learning approach: It is a commonly used vision-based technique in which we train the model by using a large number of images and apply neural networks for analysis and detection.

1.1.1 Limitations in Existing System

- ❖ Data Glove approach:

  1) The sensors used are generally very expensive so preparing the product tends to be costlier.

  2) It requires the person using the product must constantly wear gloves and findings are also less precise.

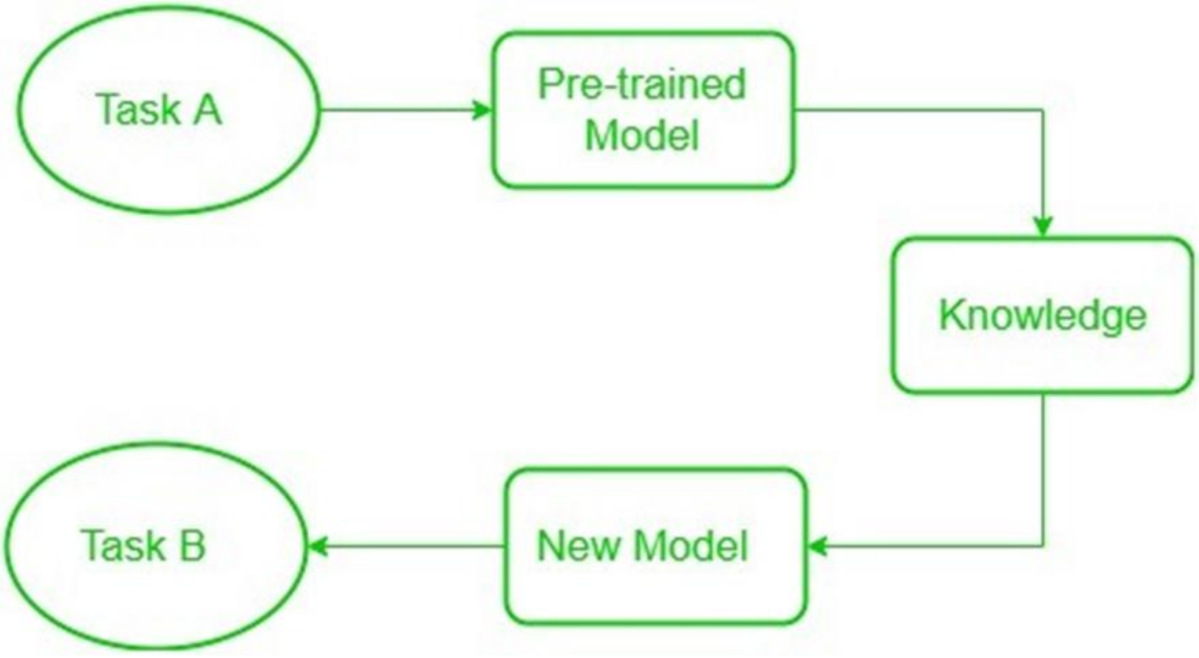
- Deep Learning approach:
    1) Takes a large number of image datasets for training
    2) Training requires a lot of time mostly taking days.
    3) Requires a lot of complex operations

.

1.2 Proposed System

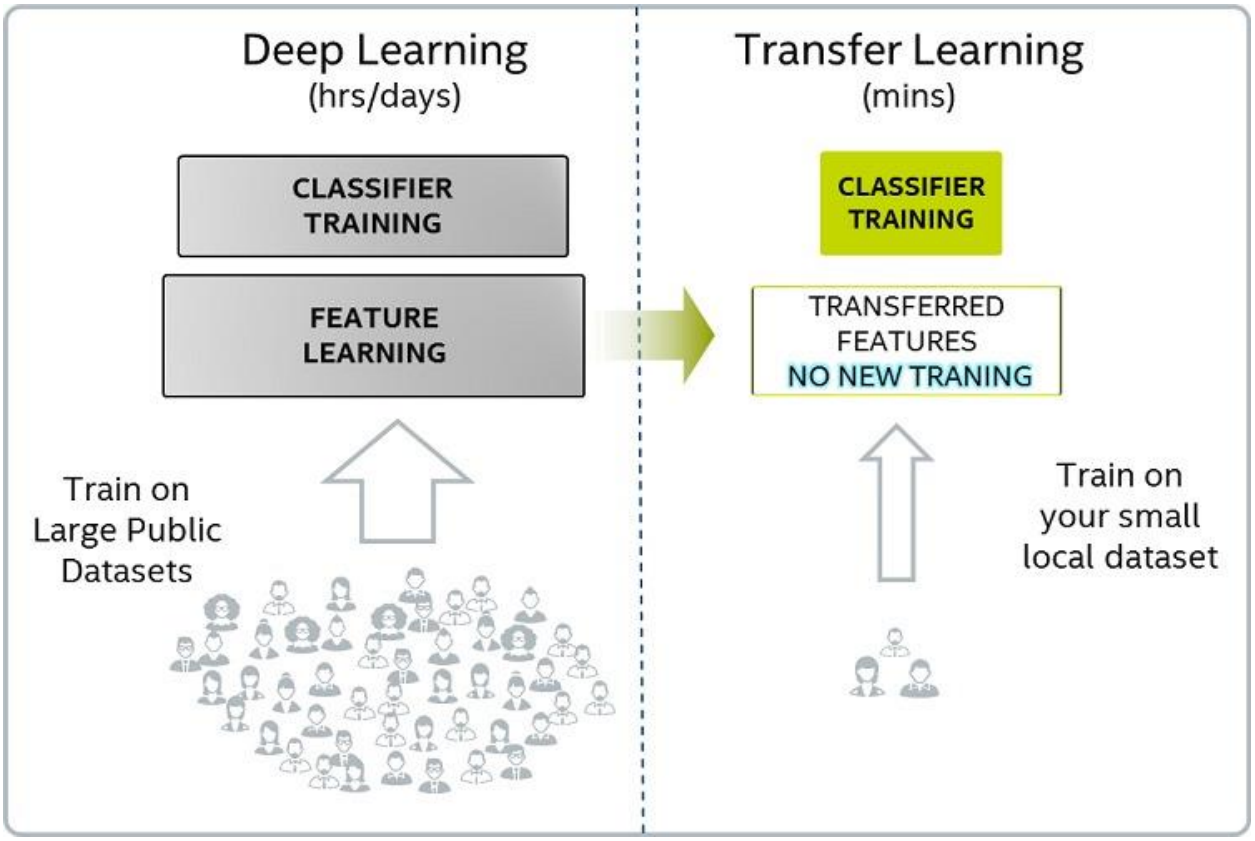
Transfer Learning approach

- By using a webcam and OpenCV we collect images.
- Label images using the LabelImg interface.
- Download a pre-trained model from Tensorflow Model Zoo.
- The model is trained with the help of transfer learning.
- Identify the sign language in real-time by running the code.



1.2.1 Advantages over Existing System

- Less number of images are required
- Model has trained very accurately when compared to the previous version
- Time taken for model training is relatively very less



2. LITERATURE SURVEY

With rapid development in technologies, the ways of interaction between humans and computers have drastically increased and are playing a significant role in all real-time applications.

From the past, there were various attempts made to identify the best way for detecting the sign language. Few of the methods were using sensor based systems in which they used sensors to identify the parameters of the gestures using its trajectory, velocity and location. The most popular technique was “Data Glove approach”. The flaw with this approach is that the user needs to always carry the equipment and is also stuffed with sensors which makes user feel uncomfortable.

But with the latest advancements made in the field of Information Technology, we are able to create vision based systems in which we provide video clips or images for detecting the hand gestures. There are various vision based systems based on “Deep Learning Approach” such as Artificial Neural Networks (ANN), Convolutional Neural Networks(CNN). But these all systems require tons of images for training . Hence, these methods take long time for model training.

Hence, to overcome the above we are introducing a new Machine Learning approach known as Transfer Learning which saves lots of time and efforts of correcting large number of images and also provide accurate results. It is a technique which uses the pre-trained model knowledge such as the weight of the neural network layers which were obtained after its heavy training and testing process. After that, we just need to fine-tune the softmax layer(last layers) of the pertained model according to our requirements i.e. real-time sign language detection.

3. TECHNOLOGY USED

- **Open Cv**
- **Computer Vision**
- **Numpy**
- **Transfer Learning**
- **Tensor Flow Object Detection API**

OpenCV - OpenCV is a cross-platform library using which we can develop real-time computer vision applications. It mainly focuses on image processing, video capture, and analysis including features like face detection and object detection. In this tutorial, we explain how you can use OpenCV in your applications.

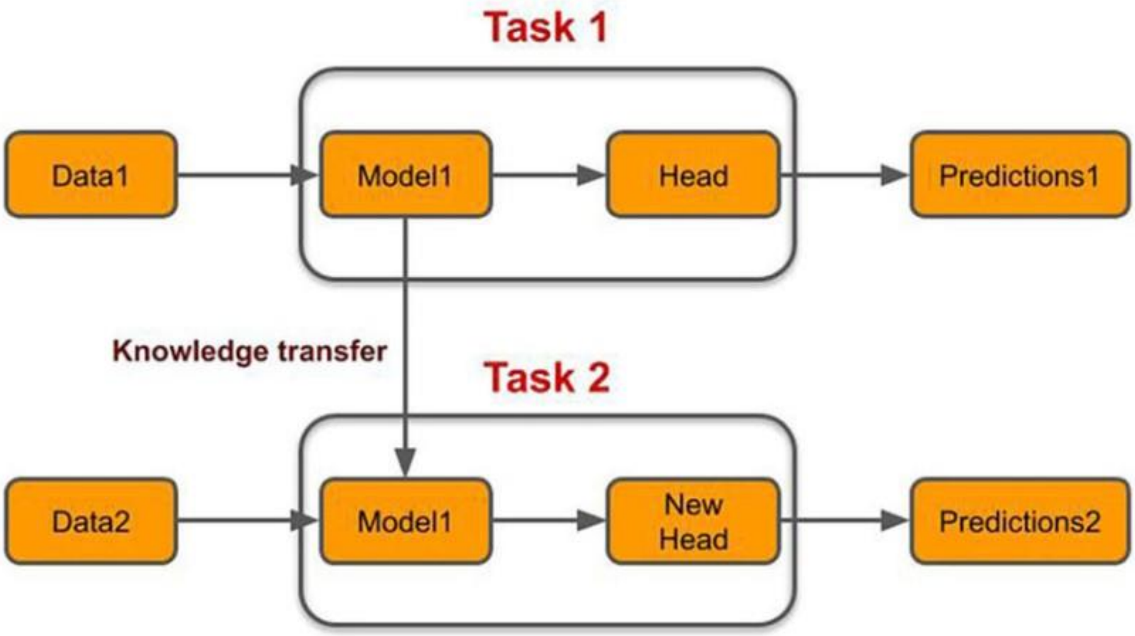
Computer Vision - Computer Vision can be defined as a discipline that explains how to reconstruct, interrupt, and understand a 3D scene from its 2D images, in terms of the properties of the structure present in the scene. It deals with modeling and replicating human vision using computer software and hardware.

NumPy - NumPy is a Python programming language library, that adds support for multiple, multi-layered and mathematical editing, as well as a large collection of high-level mathematical functions running on these arrays. NumPy's ancestor, Numeric, was created by Jim Hugunin with donations from several other developers. In 2005, Travis Oliphant made NumPy by adding the features of 17 competing Numarray to Numeric, with multiple modifications. NumPy is open-source software and has many sponsors.

Transfer Learning: In transfer learning, the knowledge of a previously trained machine learning model is applied to another model but related problem. For example, if you train a simple classifier to predict whether an image contains an airplane, you could use the knowledge that the model has learned during its training to recognize other objects like mobile.

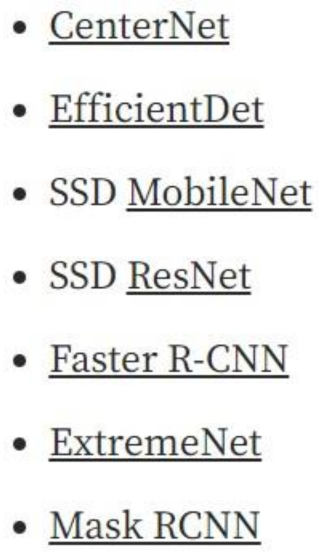
With transfer learning, we usually try to exploit what has been learned in one task to improve generalization in another. We transfer the weights that the network has learned at "task X" to a new "task Y”.

The general idea is to use the knowledge a model has learned from a task with the available labeled training data in a new task that does not have more data. Instead of starting the process from scratch, we start with patterns learned from solving a related task.



Tensor Flow Object Detection API: The TensorFlow Object Detection API is an open-source computer vision framework for building object detection and image segmentation models that can localize multiple objects in the same image. The framework works for both TensorFlow 1 and 2. Users are, however, encouraged to use the TF second version because it contains new architectures.

Some of the architectures and models that TensorFlow Object Detection API supports are:



- CenterNet
- EfficientDet
- SSD MobileNet
- SSD ResNet
- Faster R-CNN
- ExtremeNet
- Mask RCNN

4. System Specification

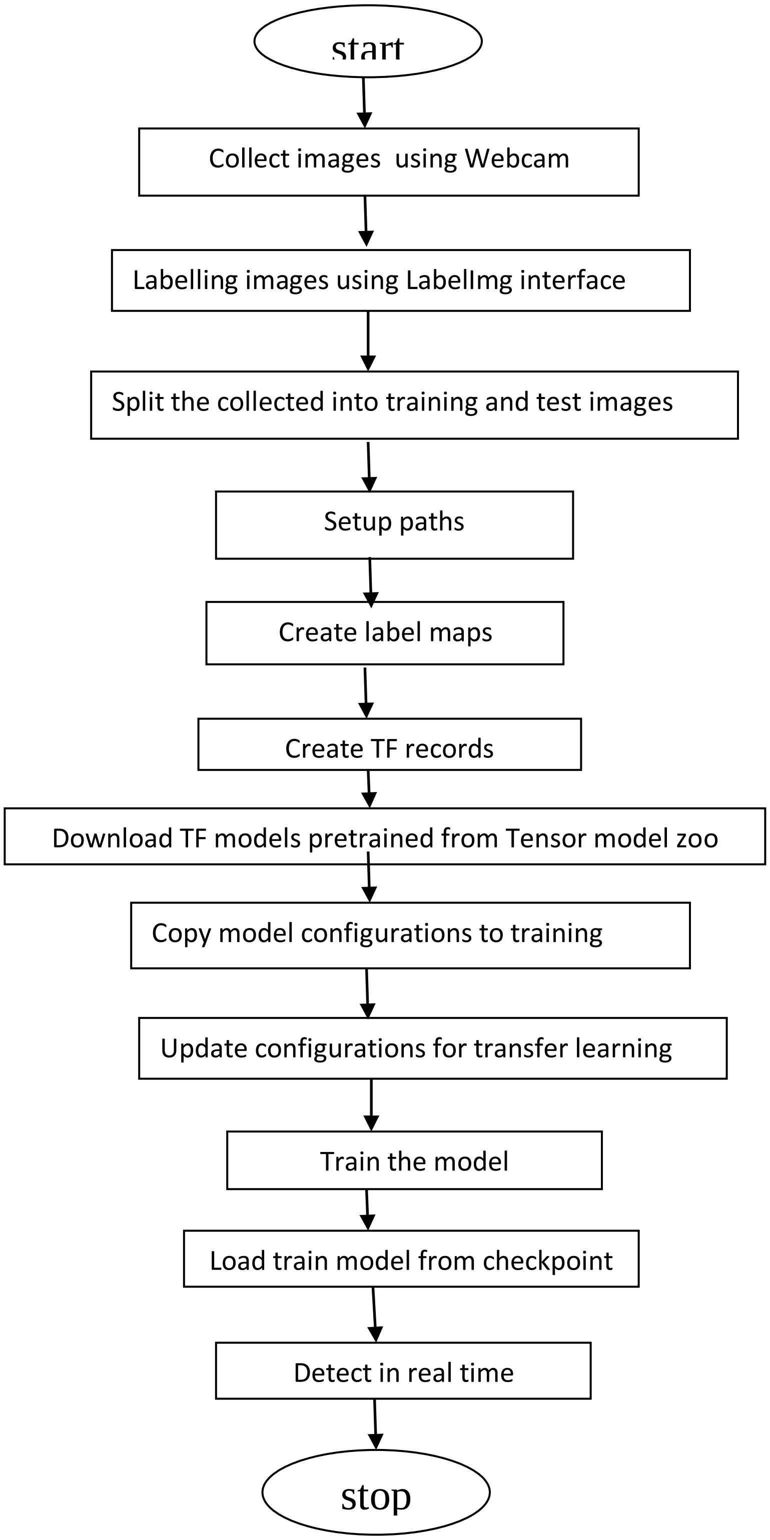
3.1 Scope of the Project

It is not only restricted to detecting the sign languages we trained but it can be used to train and detect the various kind of gestures. This approach we used can be applied to various kinds of computer vision use cases.

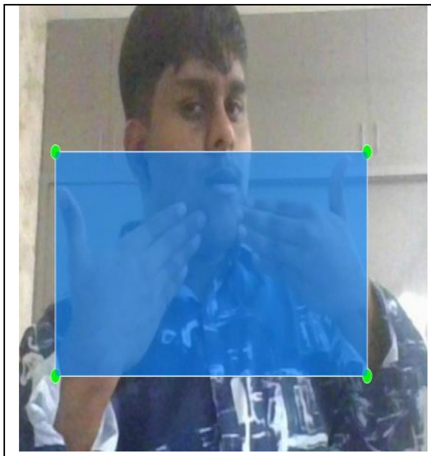
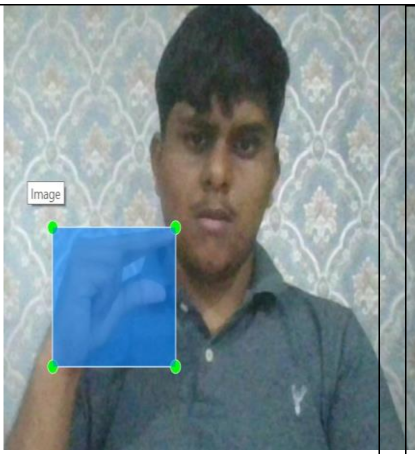
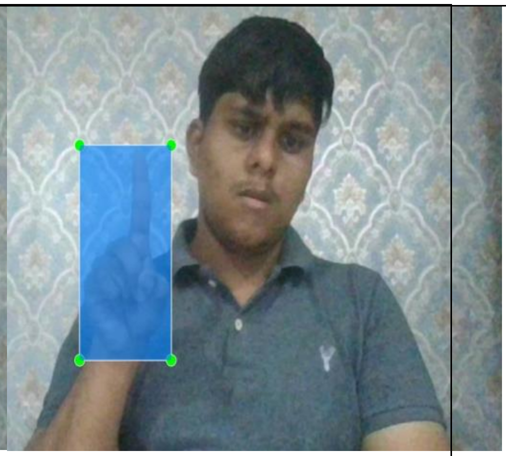
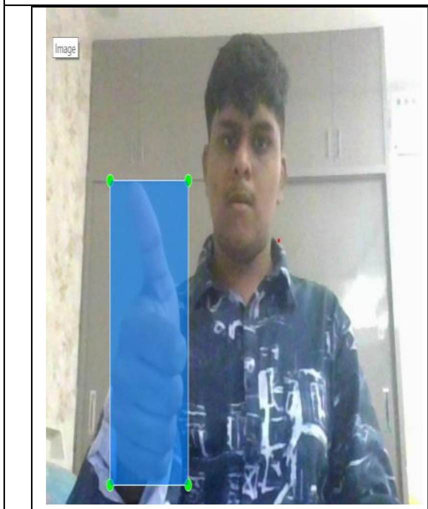
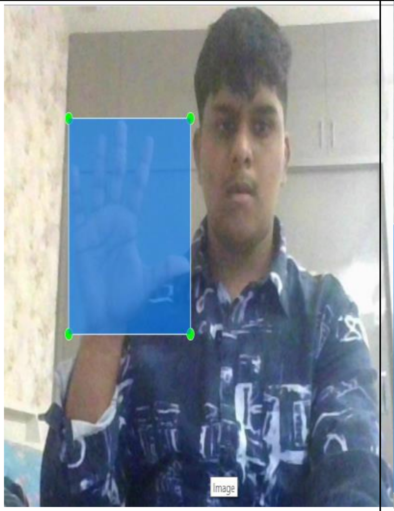
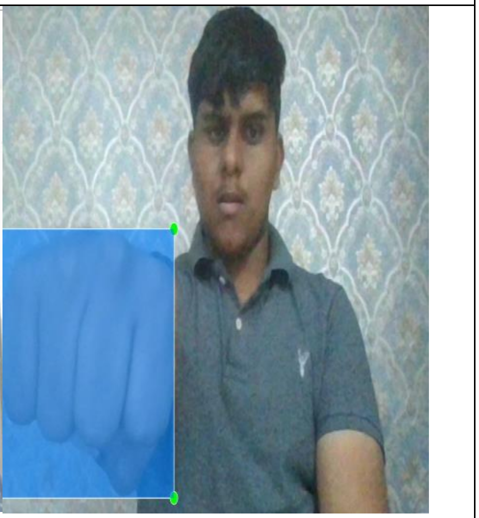
3.2 System Requirement Specification

- Python
- Jupyter Notebook
- Git
- Labelmg.py interface
- Tensorflow
- Tensorflow Object Detection API (Official Tutorial: https://molo-object-detection-api
- Pretrained Model-SDD MobileNet V2 320X320 COCO
- Visual studio Built tools (available procedure in Tensorflow Object Detection API tutorial)

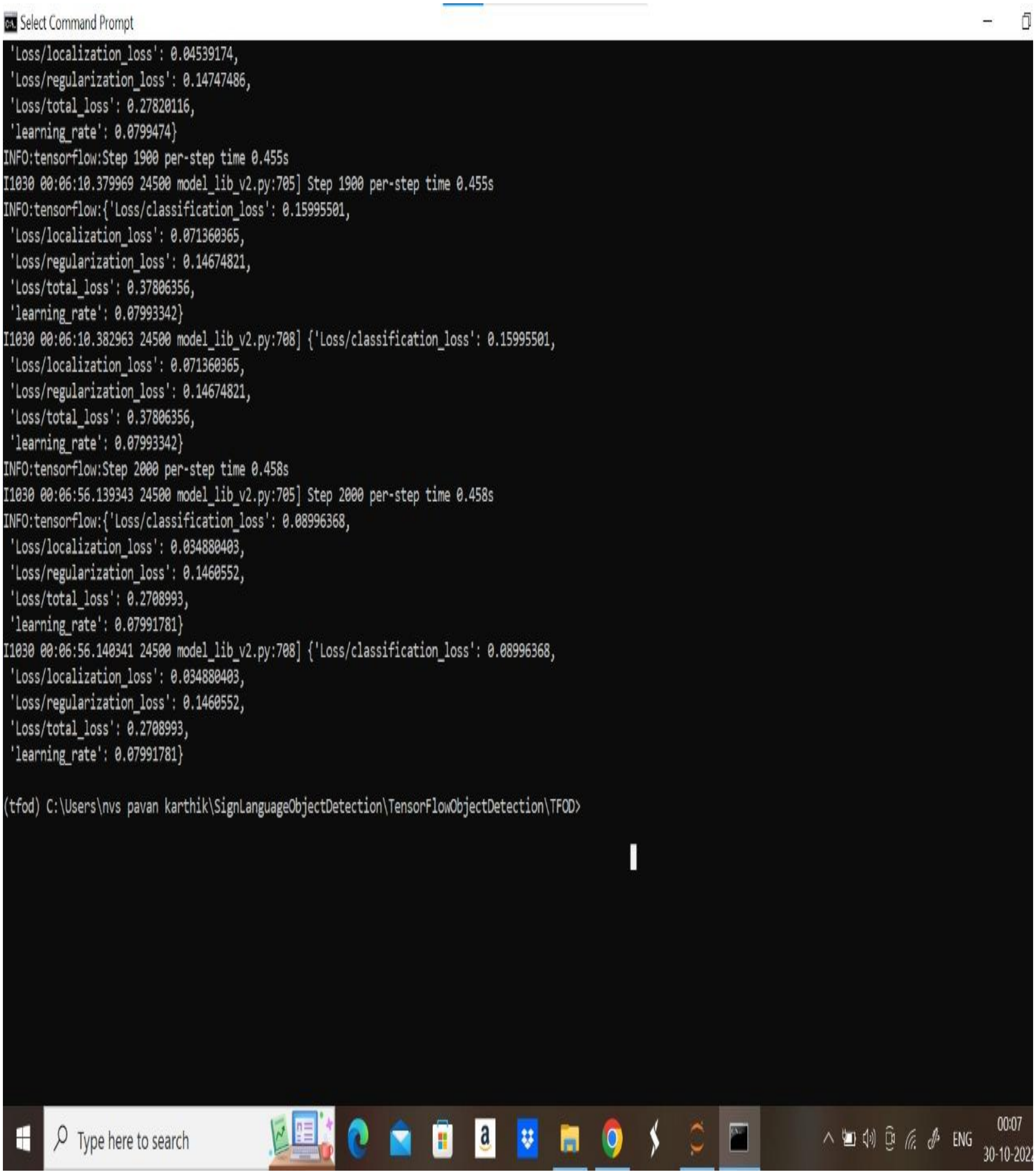
5. System Implementation


start
Collect images using Webcam
Labelling images using LabelImg interface
Split the collected into training and test images
Setup paths
Create label maps
Create TF records
Download TF models pretrained from Tensor model zoo
Copy model configurations to training
Update configurations for transfer learning
Train the model
Load train model from checkpoint
Detect in real time
stop

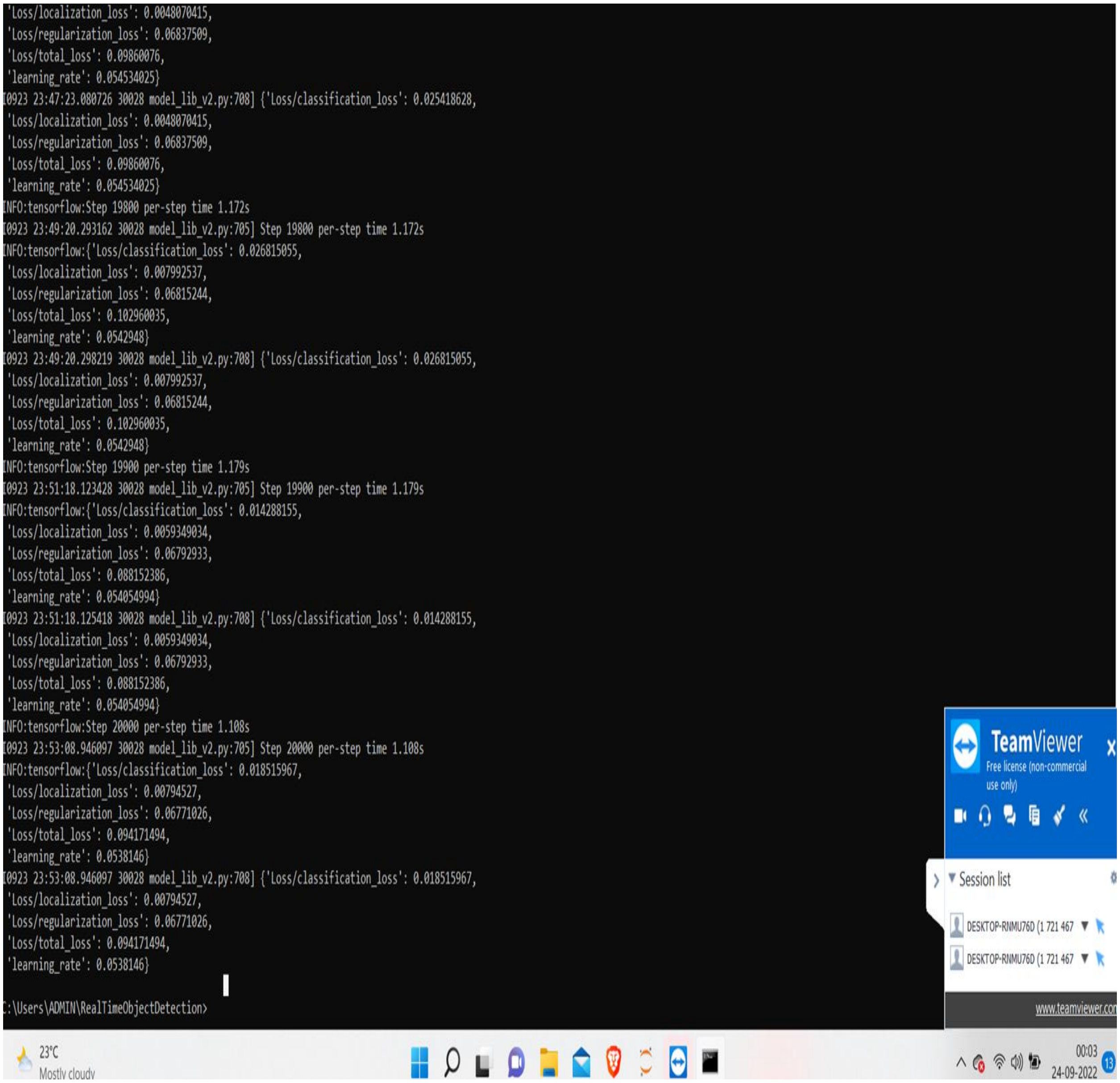
<u>Labeling our 6 Sign Languages</u>

		
Thank you	Good	I am
 Yes	 Hi	 No

6. Output

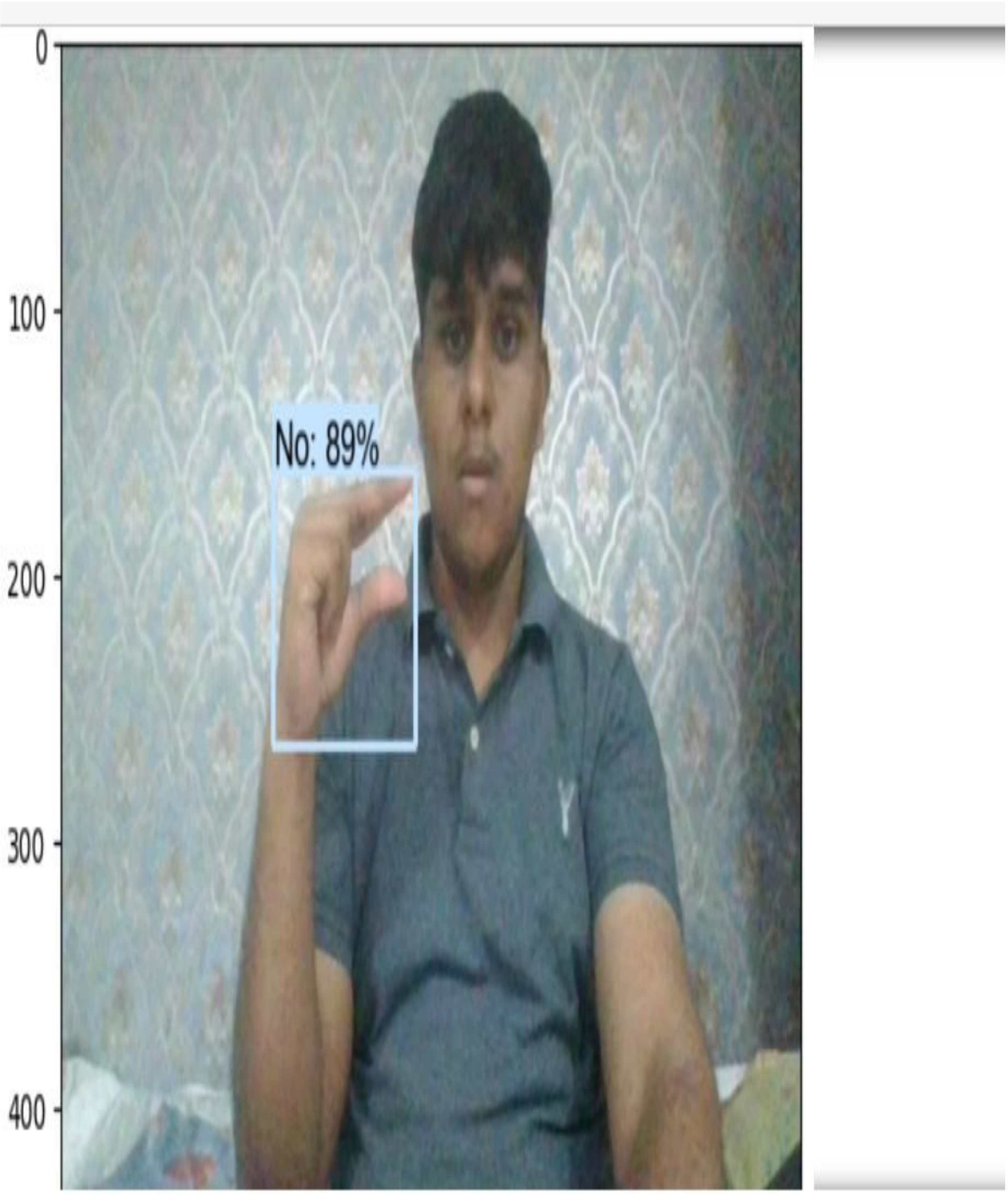


```
 'Loss/localization_loss': 0.04539174,
 'Loss/regularization_loss': 0.14747486,
 'Loss/total_loss': 0.27820116,
 'learning_rate': 0.0799474}
INFO:tensorflow:Step 1900 per-step time 0.455s
I1030 00:06:10.379969 24500 model_lib_v2.py:705] Step 1900 per-step time 0.455s
INFO:tensorflow:{'Loss/classification_loss': 0.15995501,
 'Loss/localization_loss': 0.071360365,
 'Loss/regularization_loss': 0.14674821,
 'Loss/total_loss': 0.37806356,
 'learning_rate': 0.07993342}
I1030 00:06:10.382963 24500 model_lib_v2.py:708] {'Loss/classification_loss': 0.15995501,
 'Loss/localization_loss': 0.071360365,
 'Loss/regularization_loss': 0.14674821,
 'Loss/total_loss': 0.37806356,
 'learning_rate': 0.07993342}
INFO:tensorflow:Step 2000 per-step time 0.458s
I1030 00:06:56.139343 24500 model_lib_v2.py:705] Step 2000 per-step time 0.458s
INFO:tensorflow:{'Loss/classification_loss': 0.08996368,
 'Loss/localization_loss': 0.034880403,
 'Loss/regularization_loss': 0.1460552,
 'Loss/total_loss': 0.2708993,
 'learning_rate': 0.07991781}
I1030 00:06:56.140341 24500 model_lib_v2.py:708] {'Loss/classification_loss': 0.08996368,
 'Loss/localization_loss': 0.034880403,
 'Loss/regularization_loss': 0.1460552,
 'Loss/total_loss': 0.2708993,
 'learning_rate': 0.07991781}

(tfod) C:\Users\nvs pavan karthik\SignLanguageObjectDetection\TensorFlowObjectDetection\TFOD>
```



```
 'Loss/localization_loss': 0.0048070415,
 'Loss/regularization_loss': 0.06837509,
 'Loss/total_loss': 0.09860076,
 'learning_rate': 0.054534025}
I0923 23:47:23.080726 30028 model_lib_v2.py:708] {'Loss/classification_loss': 0.025418628,
 'Loss/localization_loss': 0.0048070415,
 'Loss/regularization_loss': 0.06837509,
 'Loss/total_loss': 0.09860076,
 'learning_rate': 0.054534025}
INFO:tensorflow:Step 19800 per-step time 1.172s
I0923 23:49:20.293162 30028 model_lib_v2.py:705] Step 19800 per-step time 1.172s
INFO:tensorflow:{'Loss/classification_loss': 0.026815055,
 'Loss/localization_loss': 0.007992537,
 'Loss/regularization_loss': 0.06815244,
 'Loss/total_loss': 0.102960035,
 'learning_rate': 0.0542948}
I0923 23:49:20.298219 30028 model_lib_v2.py:708] {'Loss/classification_loss': 0.026815055,
 'Loss/localization_loss': 0.007992537,
 'Loss/regularization_loss': 0.06815244,
 'Loss/total_loss': 0.102960035,
 'learning_rate': 0.0542948}
INFO:tensorflow:Step 19900 per-step time 1.179s
I0923 23:51:18.123428 30028 model_lib_v2.py:705] Step 19900 per-step time 1.179s
INFO:tensorflow:{'Loss/classification_loss': 0.014288155,
 'Loss/localization_loss': 0.0059349034,
 'Loss/regularization_loss': 0.06792933,
 'Loss/total_loss': 0.088152386,
 'learning_rate': 0.054054994}
I0923 23:51:18.125418 30028 model_lib_v2.py:708] {'Loss/classification_loss': 0.014288155,
 'Loss/localization_loss': 0.0059349034,
 'Loss/regularization_loss': 0.06792933,
 'Loss/total_loss': 0.088152386,
 'learning_rate': 0.054054994}
INFO:tensorflow:Step 20000 per-step time 1.108s
I0923 23:53:08.946097 30028 model_lib_v2.py:705] Step 20000 per-step time 1.108s
INFO:tensorflow:{'Loss/classification_loss': 0.018515967,
 'Loss/localization_loss': 0.00794527,
 'Loss/regularization_loss': 0.06771026,
 'Loss/total_loss': 0.094171494,
 'learning_rate': 0.0538146}
I0923 23:53:08.946097 30028 model_lib_v2.py:708] {'Loss/classification_loss': 0.018515967,
 'Loss/localization_loss': 0.00794527,
 'Loss/regularization_loss': 0.06771026,
 'Loss/total_loss': 0.094171494,
 'learning_rate': 0.0538146}

C:\Users\ADMIN\RealTimeObjectDetection>
```

TeamViewer
Free license (non-commercial use only)
Session list
DESKTOP-RNMU76D (1 721 467
DESKTOP-RNMU76D (1 721 467
www.teamviewer.com
23°C
Mostly cloudy
00:03
24-09-2022
13


0
100
200
300
400
No: 89%

7. REFERENCES

1. Creating a custom object detector using transfer learning by Joyce Varghese. Journal: **The startup**
2. Transfer Learning in Sign Language by Ali Farhadi, David Forsyth, Rhyan White. Journal: **IEEE**
3. Sign Language Recognition using Transfer Learning Technique by Shivani Y, Jarul R, Vandana N. Journal: **Artificial & Computational Intelligence.**
4. Transfer Learning for British Sign Language Modelling by Boris Mocialov, Graham Turner, Helen Hastie. Journal: **aclanthology.org**
5. A Review Paper on Sign Language Recognition by Reddygari Sandhya Rani R Rumana, R. Prema. Journal: **IJERT**