**REAL TIME INDIAN SIGN LANGUAGE RECOGNITION**

**A Project Report submitted in partial fulfillment of the requirements for the award of the degree of**

**BACHELOR OF TECHNOLOGY IN**

**COMPUTER SCIENCE AND ENGINEERING**

**Submitted by**

| **Shifa Mehreen** | **121910313005** |
| --- | --- |
| **Chakrapani Anisetti** | **121910313015** |
| **Ch. Rohith Reddy** | **121910313033** |
| **R. Sai Vinay** | **121910313041** |

**Under the esteemed guidance of**

**Mr Vuna Venkata Vidya Sagar**

**Asst. Professor**



## DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

## GITAM

**(Deemed to be University) VISAKHAPATNAM**

**MARCH 2023**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING GITAM SCHOOL OF TECHNOLOGY**

**GITAM**

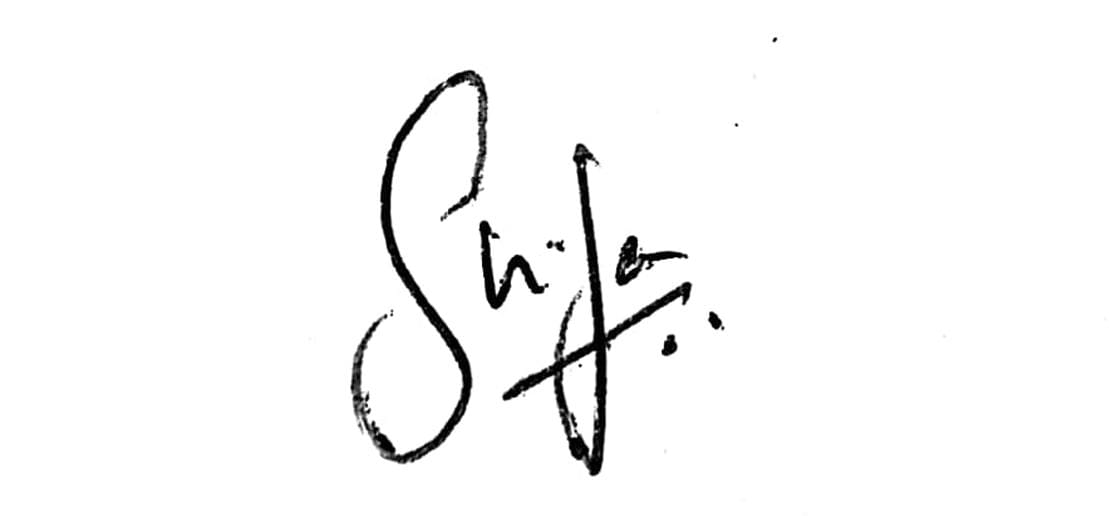
**(Deemed to be University)**



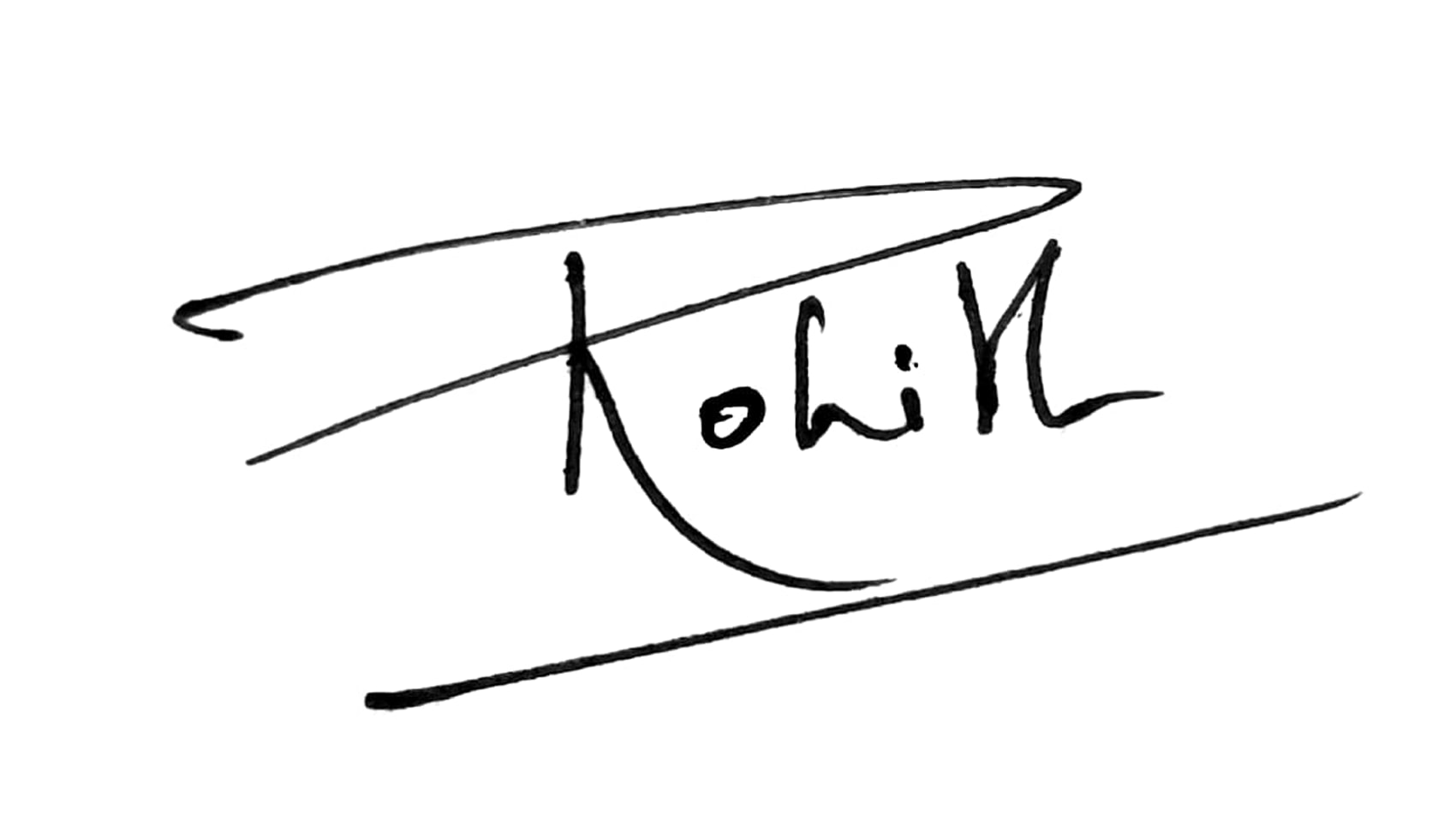
# **DECLARATION**

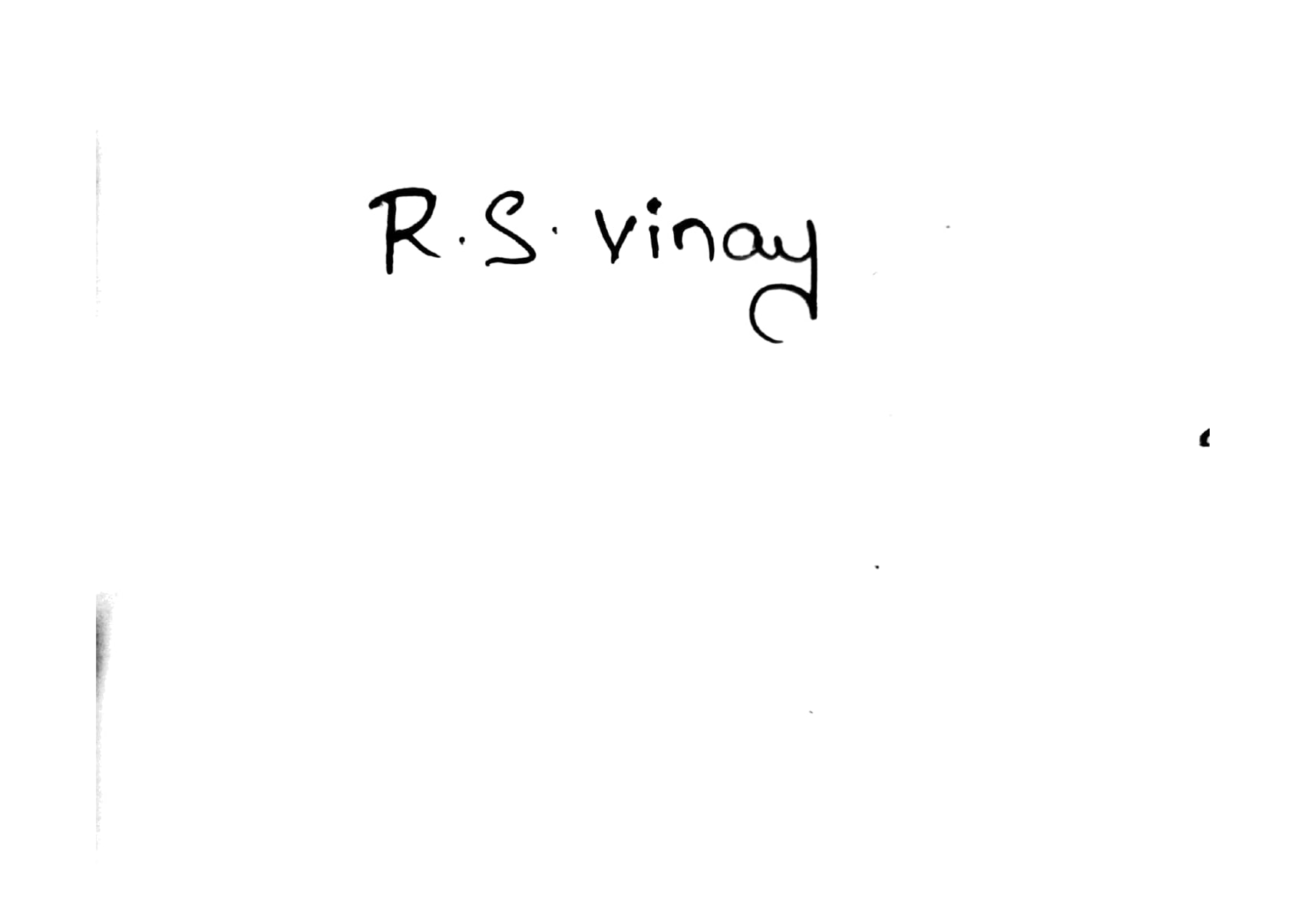
We, hereby declare that the project report entitled “**REAL TIME INDIAN SIGN LANGUAGE RECOGNITION**” is an original work done in the Department of Computer Science and Engineering, GITAM School of Technology, GITAM (Deemed to be University) submitted in partial fulfillment of the requirements for the award of the degree of B.Tech. in Computer Science and Engineering. The work has not been submitted to any other college or University for the award of any degree or diploma.

Date: 21-03-2023

**Registration No(s). Name(s) Signature(s)**

121910313005 Shifa Mehreen

121910313015 Chakrapani Anisetti 

121910313033 Chitikela Rohith Reddy

121910313041 Rajapantula Sai Vinay

# **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING GITAM SCHOOL OF TECHNOLOGY**

**GITAM**

**(Deemed to be University)**



# **CERTIFICATE**

This is to certify that the project report entitled “**REAL TIME INDIAN SIGN LANGUAGE RECOGNITION**” is a bonafide record of work carried out by **Shifa Mehreen (121910313005), Chakrapani Anisetti (121910313015), Chitikela Rohith Reddy (121910313033), Rajapanthula Sai Vinay (121910313041)** students submitted in partial fulfillment of requirement for the award of degree of Bachelors of Technology in Computer Science and Engineering.

| **Project Guide** | **Head of the Department** |
| --- | --- |
| **Mr.Vuna Venkata Vidya Sagar** | **Dr. R.Sireesha** |
| **Asst. Professor** | **Professor** |

**TABLE OF CONTENTS**

| 1. | Abstract |  |
| --- | --- | --- |
| 2. | Introduction |  |
| 3. | Literature Review |  |
| 4. | Problem Identification & Objectives |  |
| 5. | System Methodology |  |
| 6. | Overview of Technologies |  |
| 7. | Code and Implementation |  |
| 8. | Results & Discussions |  |
| 9. | Conclusion & Future Scope |  |
| 10. | References |  |

**ABSTRACT**

Communication plays a significant role in surviving and building human relations. Since birth, sharing and receiving information through verbal or nonverbal actions has been instilled into our system. We use our voices to ask for needs and help or even scream and listen to cues when in danger. But as we grow, we learn to hear and mimic the words that the people around us speak and use that to express ourselves.

With communication being such an essential aspect of survival, every human must have access to express themselves. India being the most populated country, also has a vast deaf population of around 18 million people. There are many causes to why people can go deaf and dumb, including aging, genetics, and environmental reasons like pregnancy complications or being prone to highly noisy areas, etc.

There is no cure for this; hence, people who are deaf and hard of hearing have developed their own way of communicating using sign language between their community and others. But their imparity must not stand in the way of communicating with the other majority of the hearing population since only a few people know sign language.

With Technology being a substantial part of our lives today, we have tried to eradicate this problem using high-end technology frameworks like Machine learning. We have generated a model that can recognize sign language and help the deaf and dumb community reach their voices to an even larger community.

Using our model, people can interact through the camera and know what each sign means. There are different types of sign languages. A lot of work is being done in this field using ASL (American Sign Language), and we have tried to use ISL (Indian Sign Language) for our project, to help India bridge the communication gap between the differently abled people.

**INTRODUCTION**

People who are deaf-dumb frequently communicate with one another through sign language. A sign language is nothing more than a collection of diverse gestures made by varied hand gestures, including movements, orientations, and facial expressions. Those who are deaf-dumb typically utilize these gestures to convey their thoughts. In public settings including banks, hospitals, and post offices, dumb-deaf people encounter communication barriers while communicating with normal people. The deaf occasionally need the assistance of a sign language interpreter to communicate their thoughts to hearing persons, and vice versa. By implementing such a system, the social divide between the hearing and hearing-impaired would be reduced.

The sign language used there will vary depending on the local culture and language. Indian deaf people communicate using Indian sign language (ISL). Speaking in English and using ISL is a common and well-developed method of communication for hearing-impaired people in India. For Indian Sign Language, distinct symbols are used for various alphabets. Both word-level gestures and finger spelling are used. This article describes a technique for automatically identifying static motions in the alphabet of Indian sign language. 17 letters from the English alphabet are among the symbols that are being examined for recognition.

The classification and recognition of the Indian sign language provided in real time by the dumb-deaf user are the key goals of the suggested approach. So, the algorithm's speed and simplicity are crucial. The system approach entails segmenting the hand based on skin color data, converting that segmented image to binary, and then extracting features from the binary image using distance transformation algorithms.

**LITERATURE REVIEW**

This model deals with the “Real Time Indian Sign Language Recognition” for deaf and dumb people’s communication. Both manual and non-manual signals are used in sign language communication. Here we deal with manual signals. The goal of the sign language recognition system is to provide a quick and accurate way to translate text or voice, making it easier for hearing people to understand sign language.

Sign Language is basically hand gestures that require us to recognize what each sign means basing on already defined signs. These signs empower the deaf and dumb to visualize and understand what the other person wants to say. Since not everyone knows sign language, we tried to create a model that would help the ones that do not know sign language understand what a particular hand gesture means.

Machine Learning and Deep Learning is now being used everywhere. These powerful technologies have revolutionized the way businesses work and have made our daily lives simpler. By training our computers using the data available and allowing it to learn from that data has allowed us to predict outputs without having to explicitly program. Image and Pattern recognition are few such techniques that can read the images displayed and guess what the image is.

We have created a model where a person is required to interact with the webcam and using Computer Vision and advanced algorithms like YOLO, the machine could recognize the hand gesture based on the data it was trained with and label the gesture accordingly.

Artificial intelligence (AI)'s field of computer vision enables computers and systems to give out useful information from digital photos, videos, and other visual input. YOLO helps us to locate objects in particular images, localize and quickly recognize them based on the dataset given. It creates a bounding box around the object and labels it appropriately.

We have firstly created a dataset for training. We built a custom dataset, then labeled the images using the LabelImg module in Python. Then we did the preprocessing of images and feature extraction, where we extract the ROI, here the region of interest (ROI) is the hands, later image enhancement and segmentation were done. We then train the model using these images present in our custom dataset.

We used YOLO as our object detection model, considering its speed and efficiency. We have various classes for various sings, which we give to the YOLO algorithm for training. Now while testing, when we interact with the webcam, it tracks our hand, creates a bounding box and predicts what gesture it is based on the dataset given to it while training.

As an output, we get a label and the accuracy, of what the sign is on the bounding box. Different signs have different accuracies. Accuracy depends on how well, the model is able to classify the images based on the input data given.

We have used various websites that help us understand how Indian sign language is different from the other sign languages. ISL uses both the hands for the english alphabets, hence we had to track both the left and right hands and also the finger positions, while creating a data set and while testing. There are research works on-going basing on ASL mainly, hence we tried using ISL to have a better understanding to how we can improve ISL reach higher levels in research.

We used PyCharm to implement the code for creating a custom dataset for training purposes. We also used a Kaggle data as reference. We used python for the entire project. Python is a simple and easy to code language and has various predefined packages and has vast modules we can import from. The YOLO model was implemented in Google colab, where we cloned the YOLO’s official repository from git and used it for training our model with the sign language data. It basically used transfer learning techniques for recognizing signs in real time based on what our dataset is.

From the references, we were able to understand that there’s still a lot of work that can be done in this field. Classifying images and converting it into grammatically correct text in real time, using NLP to get the right sentences, AI to automate the process and other modern techniques requires a lot of time, effort, better resources like large dataset and GPU systems.

In this project we were only able to recognize what each sign means. We had different classes for different signs, through which we could predict what hand sign was being displayed using the video camera. Just like automated captions for a video like the youtube videos, we would need to analyze the person’s actions and gestures and give out a proper sentence as an output. We still have a long way to go.

**PROBLEM IDENTIFICATION AND OBJECTIVES**

**THE PROBLEM:**

When compared to other sign languages like American Sign Language (ASL), British Sign Language (BSL), Indian sign Language is not popular. One of the main reasons is the complexity of the Indian signs over other languages. Most of the indian signs are two handed, which tend to be more difficult than single sign languages. Using opencv and convolution neural network (CNN) foe training the model is very difficult as there are two different objects to detect within a bounding box and there is a lot of information within a matrix grid, a lot of information may get lost.

**THE SOLUTION:**

To solve this problem we have to use high end architectures such as YOLO(You Only Look Once) which are used in image segmentation and object detection. YOLO uses bounding boxes and neural networks in order to make predictions. LabelImg is an annotation tool which is used to label the objects within an image. This plays a key role in object detection. LabelImg returns a text file which contains class name and bounding box coordinates (top-left corner and bottom-right corner). As YOLO is trained using the COCO dataset, we use transfer learning to replicate the results.

**OBJECTIVES:**

* Image Collection
* Object Labeling
* Dataset Creation
* Training YOLO with a custom Dataset
* Using the custom weights for training which are produced after training with custom dataset

**SYSTEM DESIGN:**

****

**OVERVIEW OF TECHNOLOGIES:**

Python - Python is a high level language with huge libraries, capacity to handle enormous quantities of data, and beginner friendliness. It is mostly utilized in the fields of machine learning and data science. It also offers all the functionality that is included in other general purpose languages. It is object-oriented, enables functional programming, has dynamic typing, and Python's code is very legible.

opencv - OpenCV\_contrib is a specialized module present in the Python programming language, which is exclusively needed for the system to run SURF feature descriptions alongside the OpenCV module present in the open-source library.

YOLO -You only look once (YOLO) is a state-of-the-art, real-time object detection system.When testing, it considers the entire image, allowing the global context of the image to influence its predictions. In contrast to R-CNN systems, which require thousands of evaluations for a single image, it also makes predictions with a single network assessment. As a result, it is incredibly quick—more than 1000 times as quick as R-CNN and 100 times as quick as Fast R-CNN. For additional information on the complete system, see our paper. Darknet prints out the objects it found, along with its confidence level and the amount of time it took. Darknet can't display the detections directly because we didn't compile it with OpenCV. Instead, predictions.png is where it saves them.

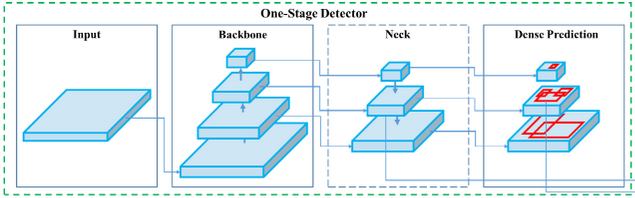
Torch - Written in the Lua programming language, Torch is an open source ML library for building deep neural networks. It's one of the most popular venues for deep learning study. The tool is created to shorten the time between research prototypes and deployment. The following crucial characteristics are offered by PyTorch: Automated differentiation, dynamic graph computation, Python support, tensor computation, TorchScript, and variable.

LabelImg -To mark object bounding boxes in photos, utilize the lightweight and user-friendly image annotation tool LabelImg. An overview of LabelImg, when to use it, and instructions on how to quickly annotate photos are provided in this page. For computer vision applications to be successful in the long run, selecting the appropriate image annotation software is essential.

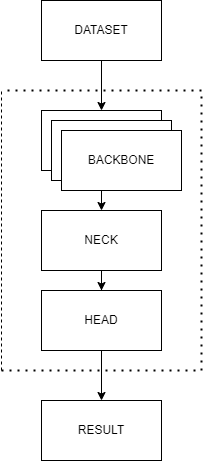
Colab - Colab is particularly well suited to machine learning, data analysis, and teaching. It enables anybody to create and execute arbitrary Python code through the browser. Technically speaking, Colab is a hosted Jupyter notebook service that offers free access to computer resources, including GPUs, and requires no setup to use. Colab's resources are neither endless or guaranteed, and the consumption caps occasionally change.

Github - Its software development and version control using Git are both supported by the website GitHub. It offers each project wikis, access control, bug tracking, software feature requests, continuous integration, and Git's distributed version control. The normal Git command-line interface may be used to access and manage projects on GitHub.com; all standard Git commands are compatible with it. Users can explore the public repositories on GitHub.com as well. Git plugins and a variety of desktop clients are also offered.

**METHODOLOGY:**

****

Object Detection is one of the computer vision tasks which uses image segmentation, image localization techniques. It is used to detect objects in an image or video stream. One of the earlier object detection modules is R-CNN, which uses a CNN with greater depth and Regions. As its successor, YOLO is far more advanced than R-CNN. YOLO is faster than R-CNN as it processes the whole image at once.



Input layer takes the input from the dataset. Backbone Layers extracts the features from the input. The Neck Layer is similar to the bottleneck layer in GAN. it compresses all the extracted features. YOLOv5 has 3 layers which are used to predict the class probabilities and bounding boxes.

YOLOv5 is pre trained using the COCO dataset. COCO Dataset is a huge dataset which contains Common things which we come across in our daily life. It contains up to 90 classes, making it one of the best dataset for object detection. So in order to do transfer learning, we need to have a custom labeled dataset. If the dataset is not labeled, we have to label tem using annotation tools such labelImg, VGG image annotator.

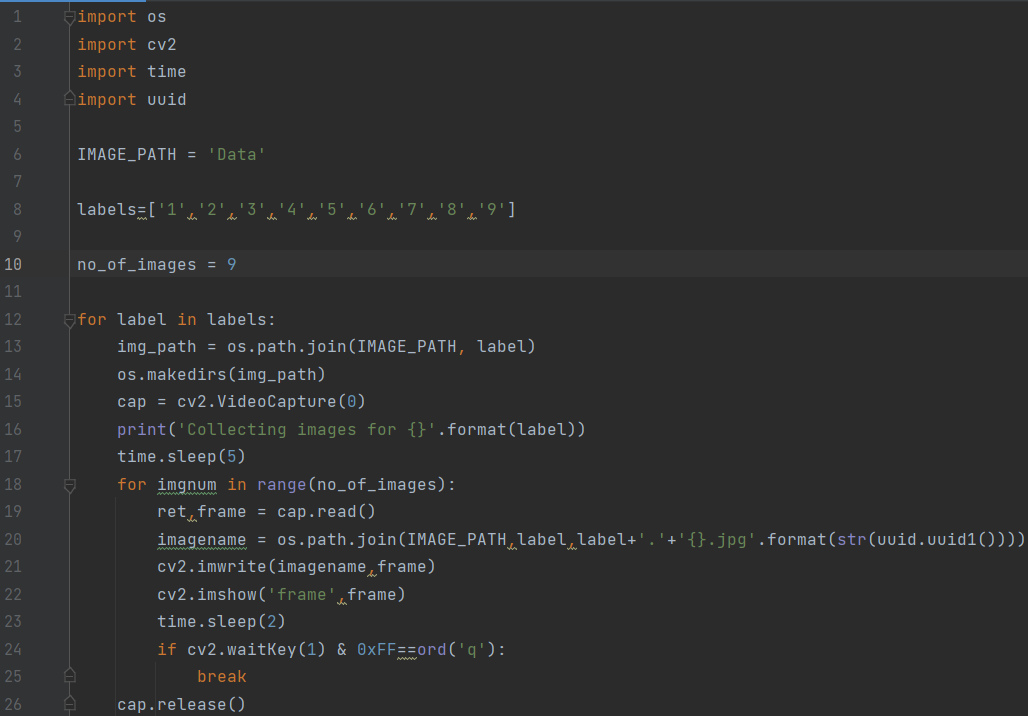
We used labelImg as it provides multiple annotation formats like XML, TXT. labeling a object in image gives its respective object number and coordinates of the bounding box. data should be separated into train and validation folder and in each folder the images and their corresponding labels are to be separated into their respective folders. as YOLO model looks for these folders.

The YAML file contains all the necessary data for our dataset to train. It contains the training data path and the validation data path. number of classes in the dataset and names of each object. This file should be uploaded to the data folder of the YOLO, because the yaml file of the previously trained dataset is present in the data folder.

Once the training is completed, the results are stored in the runs folder. It contains the weights which are obtained after training. These weights are then used for testing, and it also predicts a few classes to test the accuracy of the trained model, if we are not satisfied with the accuracy then the data can be trained once again by increasing the epochs.

With the updated weights assigned to the data, we can test the model by giving data from external sources which can be many formats, –0 is used to access webcam and .mp4 is used for video.

**CODE AND IMPLEMENTATION:**

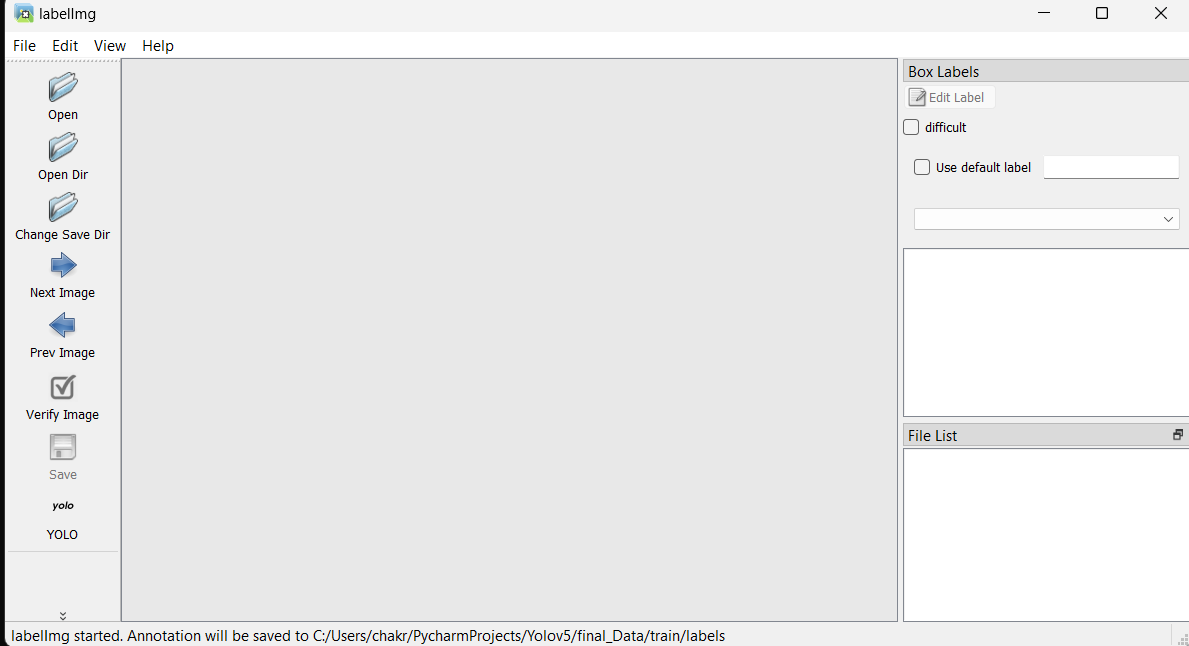
1. Data is collected using a python program which uses opencv module

* os module is used to assign paths to the folders which store the images.
* the opencv (cv2) module contains many functions, here we used VideoCapture() function, argument – 0 is used to access the webcam and release() is used to release all the resources.
* the time module is to add delay.
* the uuid module is to name the captured images.

1. DATA folder contains all the images that are collected



1. Once all the data was collected, we started labeling, we used LabelImg which is an annotation tool.

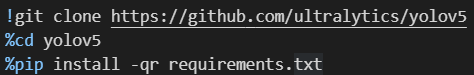


* open dir is used to open the directory which contains the images.
* change save dir is used to open a path which is used to save the annotated files.
* annotated file should be changed to yolo as we are working with yolo

1. In Colab notebook, we start with mounting our notebook with google drive.

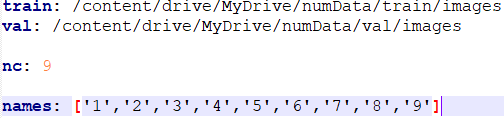


1. We cloned the official YOLOv5 repository into our notebook using the following command.

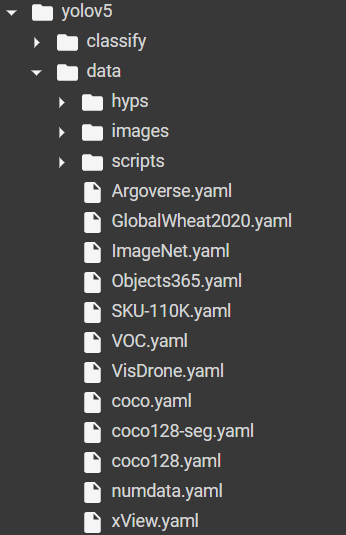


* Now the yolov5 folder is created in our notebook environment which contains all the source code.

1. A YAML file is created which contains the training and validation paths, number of classes and list of classes.



* This file is then loaded into the data folder which is present in the yolov5 folder.



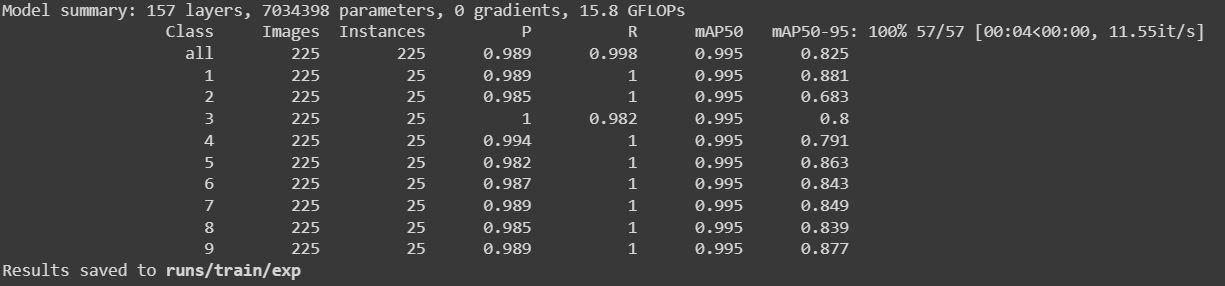
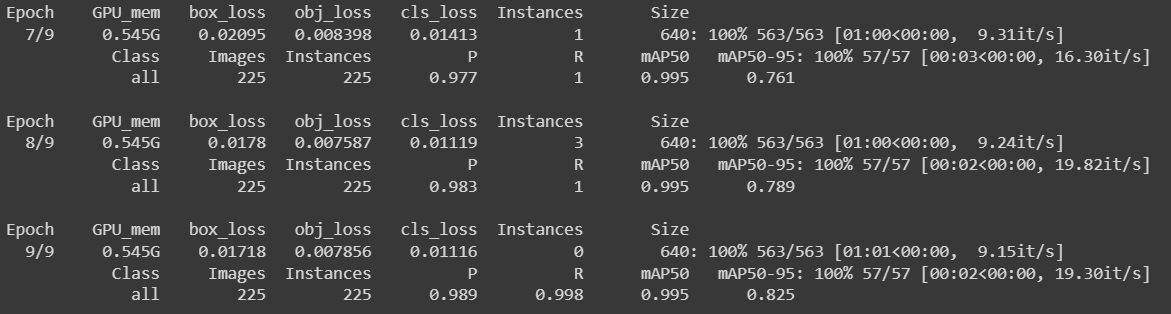
1. Now we will train the model with our custom dataset using the following command



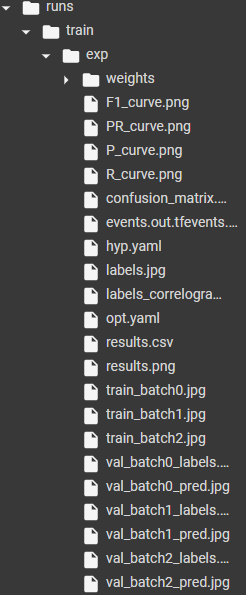
* train.py is one of the YOLOv5 source code.
* arguments : img – image size, batch – batch size, epochs – number of iterations.
* data – YAML file, weights – weights of pre trained model.



* nc is overridden by our numdata.yaml.



1. We can see the trained model is saved in the runs folder.



1. We can see the model has predicted some classes.

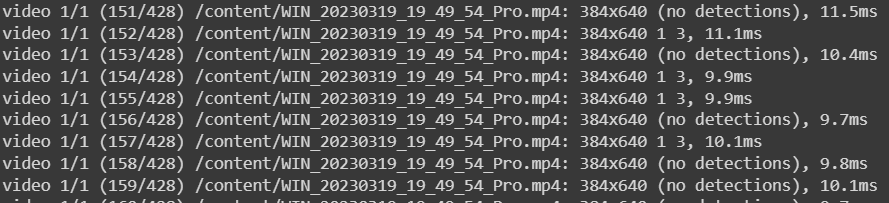
* This is done using the ML technique called Transfer Learning.
* Transfer Learning is the process by which we replicate the results of a pre-trained model with a custom model.
* It saves a lot of time.



1. Now we test our input using the following command:

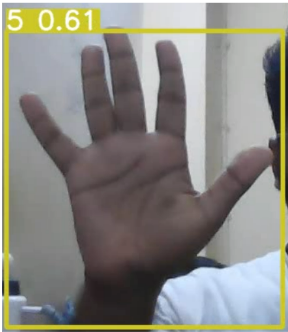
!python detect.py --weights /content/yolov5/runs/train/exp/weights/best.pt --img 640 --conf 0.35 --source /content/WIN\_20230319\_19\_49\_54\_Pro.mp4

* arguments : weights – weights which are obtained by training our custom dataset, img – image size, conf – confidence score, source – input



1. We can see that the video input is getting converted into frames and it detects labeled objects within those frames.
2. We can update YAML files to obtain pre-trained models for words and alphabets.

**RESULTS:**

****

* we can see the hands are surrounded by bounding boxes, with its corresponding label and accuracy.

**CHALLENGES AND LIMITATIONS:**

1. Limited Dataset.
2. Object Labeling - every object should be labeled manually, which takes a lot of time.
3. Not Every device contains a high processing GPU, without it training can take a lot of time.

**CONCLUSION & FUTURE SCOPE:**

In this way we created a model which can be used to interact with deaf and dumb people more effectively. With a huge labeled dataset we can train the model with classes present in it, which reduces training time and with a higher GPU system we can even label the videos by converting them into frames and each frame is labeled with a class. It solves the problem of complex signs.

We could later automate the process using AI, where the system can learn new signs and predict outputs on its own. Using a better camera, we could track the movements of hands and body, including the elbows and facial expressions, to understand different signs.

We were now only able to recognize a few alphabets and numbers that were displayed using hand gestures. We aim to recognize continuous hand, body and facial movements, to form sentences and fully understand what the person is trying to express and in what tone. Further, one could interact with the video camera using sign language and the other could read the text developed from that sign language, i.e, sign to text translation. This could allow hearing and hearing impaired people to interact in the video calls easily. People could learn sign language in an easier manner. This will help the differently abled people express their thoughts, ideas and opinions reach a larger audience and feel empowered overall.

**REFERENCES:**

SOURCE:

[1] Indian Sign language - [ISL](https://indiansignlanguage.org/)

[2] Talking Hands - [Talking hands](http://www.talkinghands.co.in/)

[3] [Indian Sign language and Training Centre](https://islrtc.nic.in/)

TECHNOLOGIES:

[3] YOLOv5 - [YOLOv5](https://github.com/ultralytics/yolov5.git)

[4] LabelImg - [LabelImg](https://github.com/heartexlabs/labelImg.git)

REFERENCE DATASET:

[5] [Indian Sign Language](https://www.kaggle.com/datasets/prathumarikeri/indian-sign-language-isl)

JOURNALS:

[6] [Automatic Indian Sign Language Recognition System](https://ieeexplore.ieee.org/abstract/document/6514343?casa_token=T7JZVRZyiScAAAAA:KgZDx4Cn6bDGCdGAIFZtDurW6K3L514C8Bs8bK-Ehf0IusIAFnFOMe_WiWQFrwk14a9EfuZMYg)

[7] [Real time Indian Sign Language Recognition System to aid deaf-dumb people](https://ieeexplore.ieee.org/abstract/document/6157974?casa_token=ymQrGHDMVuEAAAAA:bI8v1Cdvb9BoUobTSI12QcrBWF-gvV017_M15BjgsdEnBmOYHDekflBWqX1dBXXYlycVCUPn_g)

[8] [Sign language recognition using image based hand gesture recognition techniques](https://ieeexplore.ieee.org/abstract/document/7916786)