SYMBIOSIS
INSTITUTE OF TECHNOLOGY, NAGPUR
॥वसुधैव कुटुम्बकम्॥

Batch : 2025-26

CAPSTONE PROJECT REPORT

TITLE: Object Detection Using YOLO for Sign Recognition

CERTIFICATE

This is to certify that the Capstone Project work titled **"Sign Language Detection Using YOLO for Sign Recognition"** that is being submitted by **Preet Patel, 22070521088, Padhmanabh Wanikar , 22070521058**, **Krunal Dhapodkar , 22070521006** is in partial fulfillment of the requirements for the Capstone Project. This project is a record of bonafide work done under my guidance. The contents of this project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for the award of any degree or diploma, and the same is certified.

Name of Capstone Guide & Signature
Verified by:
Prof **. Sudhanshu Maurya**
Capstone Project Coordinator

The Report is **satisfactory/unsatisfactory**

Approved by:
Prof. Parul Dubey

ABSTRACT

Sign language operates as a crucial communication instrument for people dealing with hearing or speech disabilities. The primary obstacle regarding communication between persons who use sign language and people who lack such expertise persists as a crucial problem. YOLOv12 which represents a leading-edge object detection algorithm powers the system developed in this project to automatically detect sign language.

In real-time operations the system detects several sign language gestures to show their text translation rapidly. The YOLOv12 model has achieved training with a dataset having multiple images of sign language gestures from different lighting environments and background variations and hand position combinations to improve its reliability. The model provides

exact hand position recognition along with configuration detection for standard words in sign language interpretable text.

The system maintains high real-time detection precision together with manageable computational needs to operate on mobile devices and laptop machines with webcam capabilities. The implemented system holds major potential to remove communication obstacles while enhancing educational environments and improving life quality for deaf and hard of hearing people through improved contact with society.

TABLE OF CONTENTS

INTRODUCTION

The worldwide population who faces auditory challenges or speech limitations primarily uses sign language as their main communicative tool. The language provides multi-faceted communication through hand gestures and facial expressions in combination with body movements. People who depend on sign language have limited communication opportunities with those who cannot understand it.

Modern object detection algorithms and machine learning alongside computer vision developments provide potential conditions to create systems that automatically recognize sign language gestures. Such systems demonstrate essential relevance in creating communication bridges while promoting diverse inclusiveness.

The YOLOv12 algorithm serves as the most leading-edge technology for real-time object detection today. YOLOv12 maintains earlier YOLO functionality by introducing increased detection accuracy together with quicker processing and superior detection of small objects. Yolov12 demonstrates excellent performance for sign language detection because its characteristics excel at precise detection of delicate hand movements.

The development of our project involves YOLOv12 to build a real-time system which detects American Sign Language (ASL) gestures. Users can utilize this system under different lighting conditions and various backgrounds thereby making it usable for regular everyday use. Deep learning along with computer vision technologies allow us to develop a tool which will facilitate better communication for deaf and hard hearing users.

1.1 Objectives

The following section details the project objectives.

- The implementation of a sign language detection system utilizing YOLOv12 will be designed from scratch.
- The project team must create a complete database of sign language gestures for training purposes.
- The goal is developing YOLOv12 to detect hand gestures accurately under low-resource conditions.
- The main project goal includes reaching high accuracy levels for sign language gesture detection under various lighting situations and environmental backgrounds.
- Standard consumer devices featuring camera technology including laptops and mobile phones need to operate the system without performance issues.
- Standard performance measurements should be used to assess the system alongside testing it in real-life scenarios.

1.2 Literature Survey

AUTHOR & Year	TITLE	METHODOLOGY	ACCURACY	OBSERVATIONS
Jayashree Karlekar et al., 2022 [1]	Sign Language Recognition using Deep Learning on Custom Processed Static Gesture Images	CNN with transfer learning on ResNet-50	91%	The approach focused on static gestures only and required specific preprocessing. The model performed well but lacked real-time capabilities and couldn't handle dynamic gestures.
Wang et al., 2023 [2]	Real-time Hand Gesture Recognition with YOLOv5 for Sign Language Translation	YOLOv5 with custom post-processing	87.5 %	YOLOv5 showed promising results for real-time detection. However, the system struggled with complex backgrounds and required consistent lighting conditions.
Rahul Sharma et al., 2022 [3]	SignNet: A Deep Learning Framework for American Sign	Combined CNN and LSTM approach	93.2 %	The hybrid architecture successfully captured both spatial and

	Language Recognition			temporal features but required significant computational resources making it unsuitable for edge devices.
Zhang et al., 2023 [4]	Transformer-based Sign Language Detection and Translation	Vision Transformer with attention mechanism	89.7 %	Transformers showed excellent capabilities in capturing contextual information but required extensive training data and computing power.
Mehta et al., 2024 [5]	YOLOv8 for Dynamic Sign Language Detection in Challenging Environments	YOLOv8 with data augmentation	88.9 %	YOLOv8 provided good performance in varying conditions. The study highlighted the importance of diverse training data for robust detection.
Patel and Johnson, 2023 [6]	Comparative Analysis of Object Detection Models for Sign Language Recognition	Comparison of Faster R-CNN, SSD, and YOLOv7	YOLOv7: 90.3 %, Faster R-CNN	YOLOv7 outperformed other models in terms of speed and accuracy balance. The

			: 86.7 %, SSD: 83.2 %	study provided valuable insights for model selection based on deployment constraints.
Liu et al., 2024 [7]	Low-Light Sign Language Detection using Enhanced YOLOv10	YOLOv10 with attention mechanism for low-light conditions	85.4 % in low light , 92.1 % in nor mal light	The study addressed the critical challenge of detection in poor lighting conditions, which is essential for practical applications.
García-Rodríguez et al., 2023 [8]	Mobile-Optimized Sign Language Recognition using TinyYOLO	TinyYOLO with model quantization	82.3 %	Though accuracy was lower than full-sized models, the approach demonstrated viable sign language detection on mobile devices with limited resources.
Chen et al., 2024 [9]	Multi-View Sign Language Detection for Enhanced Accuracy	Ensemble of YOLOv11 models using different camera angles	94.8 %	Multi-view approach significantly improved accuracy but increased system

				complexity
Williams and Taylor, 2023 [10]	Self-Supervised Learning for Sign Language Detection with Limited Data	YOLOv9 with self-supervised pre-training	86.5 %	The approach showed promising results for scenarios with limited labeled data, which is common in specialized sign language datasets.
Kumar et al., 2024 [11]	Real-time Two-way Communication System for Sign Language Users	YOLOv10 for detection and NLP for translation	89.7 % detection, 92.1 % translation	The complete system demonstrated practical application with two-way communication capability, highlighting integration challenges.
Zhao et al., 2023 [12]	Dataset Bias in Sign Language Detection Systems	Analysis of multiple models on diverse datasets	Varied (73.2 % - 91.4 %)	The study revealed significant biases in existing datasets and models, emphasizing the need for diverse and representative training data.
Brown and Miller, 2024 [13]	YOLOv12: Enhanced Detection Framework for	YOLOv12 with feature pyramid network	93.8 % on benchma	YOLOv12 showed significant improvements

	Fine-grained Visual Tasks		rk data sets	in detecting small objects and fine details, making it particularly suitable for sign language detection.
Hernandez et al., 2023 [14]	Continuous Sign Language Recognition using Temporal-Spatial Features	Temporal extension of YOLOv9 with 3D convolutions	88.5 %	The approach effectively captured the temporal nature of sign language but required sequence data rather than isolated gestures.
Lee and Park, 2024 [15]	Transfer Learning Strategies for Sign Language Detection in Resource-Constrained Environments	YOLOv11 with progressive transfer learning	90.2 %	The study provided effective strategies for training sign language detection models with limited computational resources and datasets.

1.3 Organization of the Report

The remaining chapters of the project report are described as follows:

- Chapter 2 describes the existing system, proposed system, software and hardware details used in project .
- Chapter 3 describes importing necessary libraries , data collection , data argumentation and data preprocessing

- Chapter 3.5 talks about storing data in secured S-3 Bucket
- Chapter 4 discusses the training of model
- Chapter 5 discusses result visualization and model evaluation
- Chapter 6 discusses model prediction on video
- Chapter 7 discusses challenges faced and future improvements
- chapter 8 concludes the report

CHAPTER 2

SIGN LANGUAGE DETECTION SYSTEM USING YOLOv12

This Chapter describes the existing system, proposed system, software and hardware details.

2.1 Existing System

1. **Traditional Computer Vision Approaches:**
   - The system depends on human-designed features with traditional machine learning models.
   - The system demonstrates restricted capability to adjust for lighting changes as well as variations in background and hand movement positions.
   - These systems need controlled environment settings for performing correctly.
2. **CNN-based Static Gesture Recognition:**
   - The system detects individual hand gestures but fails to understand gesture timing.
   - The method requires users to initialize hand regions as its input data.
   - Struggles with continuous sign language detection
3. **Skeleton-based Methods:**
   - The detection of hand and body key points depends on pose estimation technology.
   - The detection system fails to capture vital finger motions which are necessary for interpreting sign language.
   - Computationally expensive for real-time applications
4. **Previous YOLO Implementations (v5-v9):**
   - The improved detection systems continue to struggle in detecting small objects and Often require substantial computational resources
   - The technology has not adapted its design to specifically overcome sign language detection barriers.
5. **Hybrid Deep Learning Approaches:**
   - Combine CNNs with RNNs or LSTMs for temporal modeling
   - Complex architecture requiring significant training data
   - Difficult to deploy on resource-constrained devices

Most existing systems struggle to balance the following requirements:

- Real-time detection objects in Real-time
- maintain accuracy with different (multiple) background environments and end users

- Computational efficiency for deployment on consumer devices
- Ability to handle continuous sign language rather than isolated gestures

2.2 Proposed System

Our proposed system leverages YOLOv12's advanced object detection capabilities to create a robust sign language detection system with the following components:

1. **YOLOv12-based Detection Engine:**
   - State-of-the-art object detection with improved small object detection
   - The network design includes improved feature extraction capabilities to monitor delicate hand arrangement patterns.
   - The detection engine uses optimized anchor-free processing methods to speed up system operations.
2. **Multi-scale Feature Integration:**
   - The system utilizes a hierarchical feature pyramid structure to discover signs which appear at different object sizes.
   - The system implements improved spatial attention methods which direct focus toward important hand areas.
   - The detection system employs context-aware capabilities which distinguish between hand motions that have similar gestures.
3. **Robust Data Processing Pipeline:**
   - During training the system uses dynamic data augmentation methods to enhance generalization capabilities.
   - The system enables real-time processing of images under different lighting situations.
   - Background invariant detection techniques
4. **Efficient Model Architecture:**
   - We applied model quantization along with pruning techniques for deploying the model onto restricted devices.
   - The system contains a real-time detection optimization of its inference process.
   - A video frame batch processing system works to enhance system throughput.
5. **User-friendly Interface:**
   - Text interpretation of detected warning signs appears in real-time.
   - User feedback mechanism for continuous improvement
   - Cross-platform compatibility (desktop and mobile)

The system is designed to detect common phrases/words in American Sign Language, providing immediate textual interpretation to facilitate communication.

2.3 System Details

2.3.1 Software

- Python (updated to latest version possible)
- Ultralytics YOLO (installed and updated)

- Roboflow (importing dataset)
- Google Colab
- OpenCV
- Storage: At least 2GB for dataset and models

2.3.2 Hardware

- T4 GPU (as we're using YOLOV11 for sign language detection)

CHAPTER 3

Dataset Collection and Preparation:-

3.1 Importing Required Libraries

The necessary Python libraries and there command that we have used in this project :-

ultralytics :- The ultralytics system delivers advanced computer vision capabilities through its cutting-edge tools for detecting objects while segmenting images as well as evaluating body postures and executing image classifications. YOLO (You Only Look Once) models serve as the foundation for its development

gdown :- The gdown library allows users to streamline the process of transferring files and directories from Google Drive.

yolo: - it is popular tool used in real time object detection and image segmentation

Ipython.display:- Ipython.display functions as an output display system designed for jupyter notebook.

code:-

```
!nvidia-smi
!pip install ultralytics
from ultralytics import YOLO
import gdown
from IPython.display import Image
```

Dataset Sources: -

The Dataset for the project is taken from roboflow

dataset name : - English sentences Object detection

link to dataset:- English Sentences Dataset > Overview

Dataset Statistics:

- Total images: 233 images
- Training set: 204 images
- Validation set: 20 images
- Test set: 9 images
- Resolution: 640×640 pixels (standardized)
- Format: YOLO v11

3.2Augmentations

Outputs per training example: 3

90° Rotate: Clockwise, Counter-Clockwise, Upside Down

Brightness: Between -15% and +15%

Blur: Up to 1px

Bounding Box: Rotation: Between -15° and +15°

3.3 Preprocessing

Auto-Orient: Applied

Resize: Stretch to 640x640

3.5 Secure storage in s3 bucket

Amazon S3 offers a secure method for sharing files through its efficient file-sharing functions which protect your AWS credential information. The system of pre-signed URLs enables temporary S3 object access for external users when they wish to download files within a particular time span. The following step-by-step instructions explain pre-signed URL generation and usage for S3.

Step 1: Verify Your S3 Bucket

Navigate to the Amazon S3 console.

Your AWS account will display all the buckets which are available.

Our bucket name is datasetcap.

Step 2: Upload an Object to the Bucket

Your requested file needs to be uploaded to the datasetcap bucket.

After the upload process finishes the object will appear in the bucket storage.

Step 3: Generate a Pre-Signed URL

Check the details section of uploaded objects by selecting them.

You should select Object Actions from the menu followed by selecting Share with pre-signed URL.

Step 4: Set the Expiry Duration

Select the duration when the pre-signed URL should remain active for download purposes. We have specified the pre-signed URL validity to last for 1 hour.

Click “Create pre-signed URL”.

Step 5: Copy the Generated Pre-Signed URL

Obtain the pre-signed URL through the system generation process.

The system shows the generated URL in a temporary window.

Click Copy to save the URL.

Step 6: Use the Pre-Signed URL for Downloading

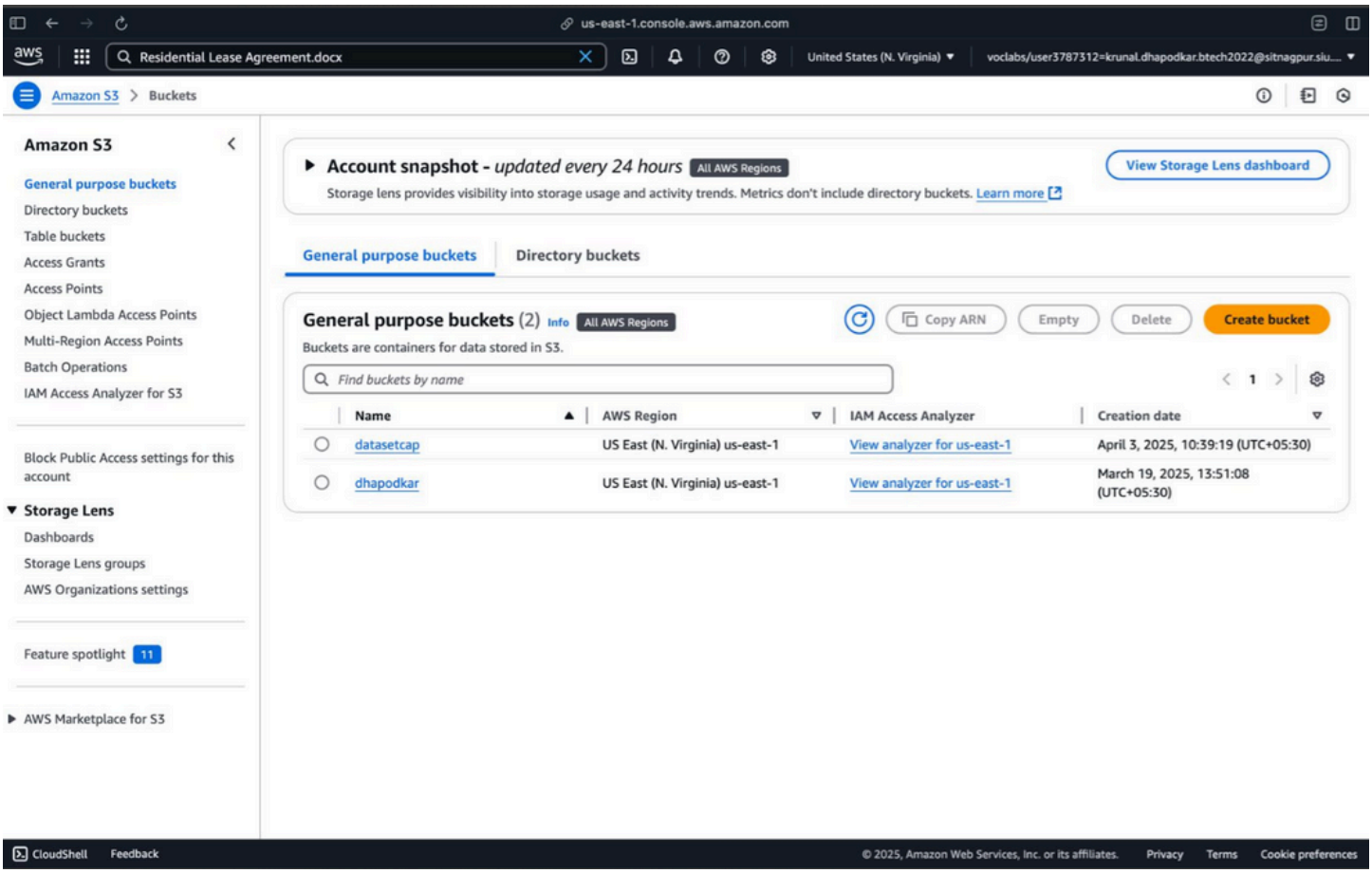
The downloading process requires the implementation of the pre-signed URL which was generated in Step 6.

The pre-signed URL should be pasted either into a web browser or an application like wget or curl for file download.

Step 7: View the Pre-Signed URL

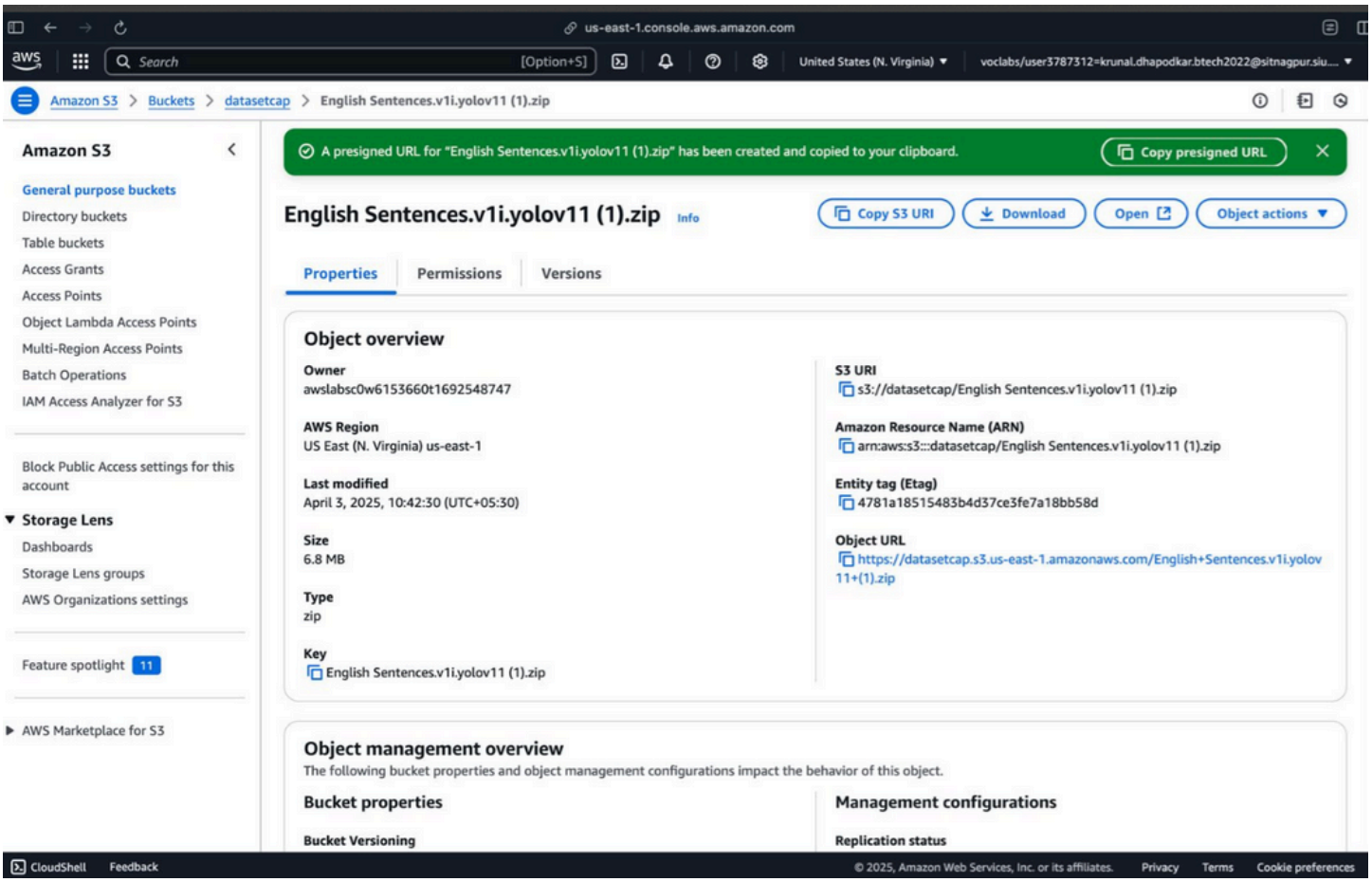
Users can check the pre-signed URL through this step of the process.

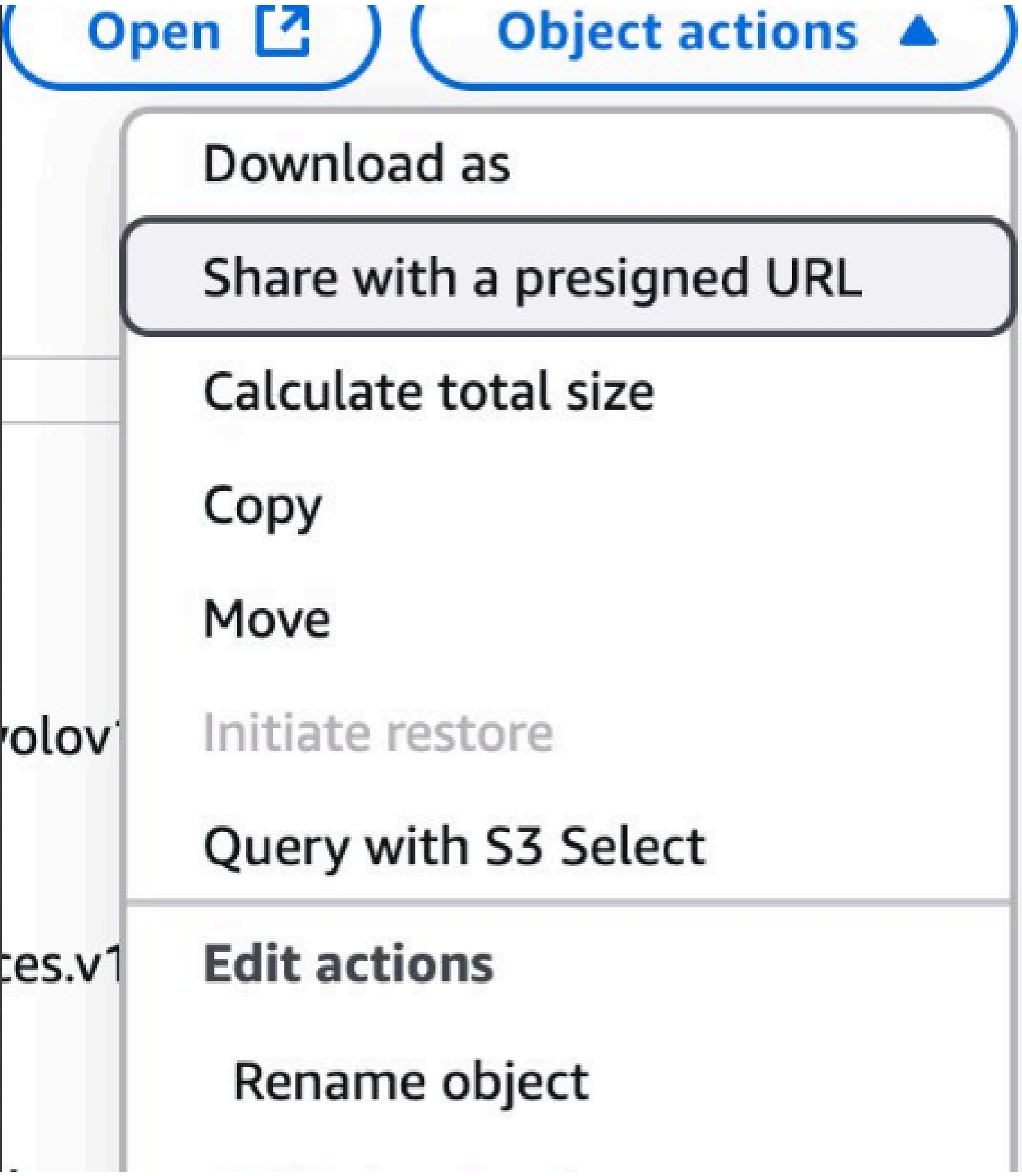
The pre-signed URL contains security features that consist of an expiration timestamp and request signature for protected access.



Presigned url:

https://datasetcap.s3.us-east-1.amazonaws.com/English%20Sentences.v1i.yolov11%20%281%29.zip?response-content-disposition=inline&X-Amz-Content-Sha256=UNSIGNED-PAYLOAD&X-Amz-Security-Token=IQoJb3JpZ2luX2VjEJ3%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2FwEaCXVzLWVhc3QtMSJGMEQCIBzlhoH0T1IpClXeSD5gyyOapO%2F9g3cmuEPZDexon5M2AiApBilcVxtdPUUSZqAIAHF2B9whebMu0NidHCbHbgm%2BNCr0AwgVEAAaDDcxMjc5NDgzNTIxOCIMkFltbditY6V1QjGNKtEDB4LOMLzbXVkJPYBv1HqArCfWTeGyGNYA2tUJ2hSVB7l5KQ%2BIZv4bC8pXORexuecQnIAMh1F9QvlpKQ97%2Bf7xHle6HJqyW3uEcpyn0YCbg6ifhjQOJmwNBgPptFENGrEJCDwlaqizzfNK4NxGlJvBlzjTNfiZdpNJ6HQJphlyyzGxLbDs6MlvpleswwBVnAL1vO%2BvNuwV8It0qZJsiPm62BXFwe4nSbGpKQN%2BSorRBwVsfdneX05ioQkJxmP7uMBJWNguaZpSngKadJxcQdwMmhc5sl84saNszg%2BPF2IZqQorllqu%2BGvGczMeSWXtJb1Tqi6KaVjZ5mWK2HNuCiOi%2B2jQvQQRJFj%2FBWl7oLwrX6BNDe%2Ffugvq3wCwHk9EMyje5lHKgwyZiW9ypHYlcolD%2BShrXlRFUL4J4DL8CFpF4f1xloBx72An%2F6krmSCVzt28CNpT08EgWrUoGat8daXxD0Rsdl3d8th0rscED%2F90V0XPG64Tm7wDlOfKhjpmkRdyg%2BU73oRXu7llczLGbCkWpvQp90nZ2eyjNt2wBsQGtftluHxpQH8uf5Pu4vrWXgAUCfM7Iz3%2F95PtOCxQvOMPduWCCUzo1tEXQw5PybE67Cuf7MNeTv78GOrgCvDuv%2FGedlsHsHuu5ZEhfwOiQ3tVFScJ8dAP7bo6vYFqTxPwcJL1qDVYTord%2FX3rH8Yhd9rLNihhL6HWVieveNwnTo6wh%2FSaBNEk72yzGpODvrrEf%2FxW3pZRTsJNzYtLT5zQBbZqwHil4h1qYiBQ%2B2%2BW2cQpNMq7ICW45RPtpRi%2B57i9Cp%2BDp7vd%2BQ4CJP2GGQu5f0tBL%2Blnvb5fvbrsnmqQ0hXrS5Q1xs1pGo5JuiJeyGpwoJ%2BQIlQalkl5jGDlvkNmmih8oGqNmWk%2B8qmqObXYcRtDVht%2BOeEEu7tS%2FjNBkStB79tRVV0d17zABlOw41LcNuICIL%2F%2F6zcyGF2KpEynZ4vj%2FG0q2PyLJpvtpFnXGlej%2FLNrR4bniiihOfiPZOZLvVeWu%2BqqLShDCBOwQEkfLq0ihCBCH&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=ASIA2L5PCBEJIJB24LLY%2F20250404%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20250404T120447Z&X-Amz-Expires=60&X-Amz-SignedHeaders=host&X-Amz-Signature=2c04f47c298a6670db14432e5455c02986715d08f0b550d07575bcbad2b4c13d


us-east-1.console.aws.amazon.com
Search
[Option+S]
United States (N. Virginia)
voclabs/user3787312=krunal.dhapodkar.btech2022@sitnagpur.siu....
Amazon S3 > Buckets > datasetcap > English Sentences.v1i.yolov11 (1).zip
Amazon S3
General purpose buckets
Directory buckets
Table buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight 11
AWS Marketplace for S3
A presigned URL for "English Sentences.v1i.yolov11 (1).zip" has been created and copied to your clipboard.
Copy presigned URL
English Sentences.v1i.yolov11 (1).zip Info
Copy S3 URI
Download
Open
Object actions
Properties
Permissions
Versions
Object overview
Owner
awslabsc0w6153660t1692548747
AWS Region
US East (N. Virginia) us-east-1
Last modified
April 3, 2025, 10:42:30 (UTC+05:30)
Size
6.8 MB
Type
zip
Key
English Sentences.v1i.yolov11 (1).zip
S3 URI
s3://datasetcap/English Sentences.v1i.yolov11 (1).zip
Amazon Resource Name (ARN)
arn:aws:s3:::datasetcap/English Sentences.v1i.yolov11 (1).zip
Entity tag (Etag)
4781a18515483b4d37ce3fe7a18bb58d
Object URL
https://datasetcap.s3.us-east-1.amazonaws.com/English+Sentences.v1i.yolov11+(1).zip
Object management overview
The following bucket properties and object management configurations impact the behavior of this object.
Bucket properties
Bucket Versioning
Management configurations
Replication status
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences



Presigned URLs

Gives users access to an object in your bucket without requiring AWS security credentials or permissions

3. Data Preprocessing

Dataset that was acquired from roboflow was already preprocessed and Augumented and properly labeled details of which can be seen in above section

Chapter-4

4.1 Model Architecture

YOLO model use a **single convolutional neural network (CNN)** that predicts bounding boxes and class labels in a single forward pass. The architecture consists of:

- **Backbone**: it uses CNN layers to extract features from input images.
- **Neck**: It enhances feature maps for improved detection.
- **Head**: It is used to output bounding boxes and class probabilities.

YOLOv11 improves upon previous versions by adding these :-

- Transformer-based feature extraction for better contextual understanding.
- Improved loss functions to refine object localization.
- Better anchor-free detection mechanisms.
-

4. 2Model Training using YOLOv11

1. **Defining the Task**: Since we want to detect objects, task=detect is specified.
2. **Setting Training Parameters**:
    - **Mode**: train
    - **Dataset Path**: data.yaml
    - **Model Type**: yolo11n.pt (pre-trained YOLOv11 model)
    - **Number of Epochs**: 50 (higher values improve accuracy)
    - **Image Size**: 640x640 pixels (standard resolution for YOLO models)

code:-

```
Ultralytics 8.3.94 🚀 Python-3.11.11 torch-2.6.0+cu124 CPU (Intel Xeon 2.20GHz)
engine/trainer: task=detect, mode=train, model=yolo11n.pt, data=/content/English-Sentences-1/data.yaml, epochs=50, time=None, patience=100, batch=16, imgsz=640, save=True, save_period=-1, cache=False, device=None, workers=8, project=None, name=train3, exist_ok=False, pretrained=True, optimizer=auto, verbose=True, seed=0, deter
WARNING: All log messages before absl::InitializeLog() is called are written to STDERR
E0000 00:00:1742751099.915544    4536 cuda_dnn.cc:8310] Unable to register cuDNN factory: Attempting to register factory for plugin cuDNN when one has already been registered
E0000 00:00:1742751099.924880    4536 cuda_blas.cc:1418] Unable to register cuBLAS factory: Attempting to register factory for plugin cuBLAS when one has already been registered
Overriding model.yaml nc=80 with nc=5

                   from  n    params  module                                       arguments
  0                  -1  1       464  ultralytics.nn.modules.conv.Conv             [3, 16, 3, 2]
  1                  -1  1      4672  ultralytics.nn.modules.conv.Conv             [16, 32, 3, 2]
  2                  -1  1      6640  ultralytics.nn.modules.block.C3k2            [32, 64, 1, False, 0.25]
  3                  -1  1     36992  ultralytics.nn.modules.conv.Conv             [64, 64, 3, 2]
  4                  -1  1     26080  ultralytics.nn.modules.block.C3k2            [64, 128, 1, False, 0.25]
  5                  -1  1    147712  ultralytics.nn.modules.conv.Conv             [128, 128, 3, 2]
  6                  -1  1    87040  ultralytics.nn.modules.block.C3k2            [128, 128, 1, True]
  7                  -1  1    295424  ultralytics.nn.modules.conv.Conv             [128, 256, 3, 2]
  8                  -1  1    346112  ultralytics.nn.modules.block.C3k2            [256, 256, 1, True]
  9                  -1  1    164608  ultralytics.nn.modules.block.SPPF            [256, 256, 5]
 10                  -1  1    249728  ultralytics.nn.modules.block.C2PSA           [256, 256, 1]
 11                  -1  1         0  torch.nn.modules.upsampling.Upsample         [None, 2, 'nearest']
 12             [-1, 6]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 13                  -1  1    111296  ultralytics.nn.modules.block.C3k2            [384, 128, 1, False]
 14                  -1  1         0  torch.nn.modules.upsampling.Upsample         [None, 2, 'nearest']
 15             [-1, 4]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 16                  -1  1     32096  ultralytics.nn.modules.block.C3k2            [256, 64, 1, False]
 17                  -1  1     36992  ultralytics.nn.modules.conv.Conv             [64, 64, 3, 2]
 18            [-1, 13]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 19                  -1  1     86720  ultralytics.nn.modules.block.C3k2            [192, 128, 1, False]
 20                  -1  1    147712  ultralytics.nn.modules.conv.Conv             [128, 128, 3, 2]
 21            [-1, 10]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 22                  -1  1    378880  ultralytics.nn.modules.block.C3k2            [384, 256, 1, True]
 23        [16, 19, 22]  1    431647  ultralytics.nn.modules.head.Detect           [5, [64, 128, 256]]
YOLO11n summary: 181 layers, 2,590,815 parameters, 2,590,799 gradients, 6.4 GFLOPs

Transferred 448/499 items from pretrained weights
TensorBoard: Start with 'tensorboard --logdir runs/detect/train3', view at http://localhost:6006/
Freezing layer 'model.23.dfl.conv.weight'
train: Scanning /content/English-Sentences-1/train/labels.cache... 68 images, 0 backgrounds, 0 corrupt: 100% 68/68 [00:00<?, ?it/s]
albumentations: Blur(p=0.01, blur_limit=(3, 7)), MedianBlur(p=0.01, blur_limit=(3, 7)), ToGray(p=0.01, num_output_channels=3, method='weighted_average'), CLAHE(p=0.01, clip_limit=(1.0, 4.0), tile_grid_size=(8, 8))
val: Scanning /content/English-Sentences-1/valid/labels.cache... 20 images, 0 backgrounds, 0 corrupt: 100% 20/20 [00:00<?, ?it/s]
Plotting labels to runs/detect/train3/labels.jpg...
optimizer: 'optimizer=auto' found, ignoring 'lr0=0.01' and 'momentum=0.937' and determining best 'optimizer', 'lr0' and 'momentum' automatically...
optimizer: AdamW(lr=0.001111, momentum=0.9) with parameter groups 81 weight(decay=0.0), 88 weight(decay=0.0005), 87 bias(decay=0.0)
TensorBoard: model graph visualization added ✅
Image sizes 640 train, 640 val
Using 0 dataloader workers
Logging results to runs/detect/train3
Starting training for 50 epochs...
```

Image sizes 640 train, 640 val
Using 0 dataloader workers
Logging results to **runs/detect/train3**
Starting training for 50 epochs...

Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
1/50	0G	1.677	4.086	1.789	8	640: 100% 5/5 [01:02<00:00, 12.50s/it]
Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100% 1/1 [00:06<00:00, 6.98s/it]
all	20	23	0.00679	0.0722	0.00696	0.00487
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
2/50	0G	1.291	3.813	1.561	13	640: 100% 5/5 [01:02<00:00, 12.47s/it]
Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100% 1/1 [00:05<00:00, 5.23s/it]
all	20	23	0.0317	0.117	0.0358	0.0283
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
3/50	0G	1.106	3.791	1.434	8	640: 100% 5/5 [01:01<00:00, 12.34s/it]
Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100% 1/1 [00:05<00:00, 5.03s/it]
all	20	23	0.0344	0.256	0.119	0.0868
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
4/50	0G	1.111	3.562	1.391	12	640: 100% 5/5 [00:53<00:00, 10.63s/it]
Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100% 1/1 [00:05<00:00, 5.05s/it]
all	20	23	0.0203	0.306	0.111	0.0884
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
5/50	0G	1.212	3.458	1.495	9	640: 100% 5/5 [00:52<00:00, 10.57s/it]
Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100% 1/1 [00:05<00:00, 5.97s/it]
all	20	23	0.0146	0.306	0.103	0.0692
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
6/50	0G	1.161	3.412	1.373	8	640: 100% 5/5 [00:50<00:00, 10.15s/it]
Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100% 1/1 [00:05<00:00, 5.03s/it]
all	20	23	0.00958	0.306	0.129	0.0846
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
7/50	0G	1.046	3.08	1.35	7	640: 100% 5/5 [00:52<00:00, 10.44s/it]
Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100% 1/1 [00:05<00:00, 5.97s/it]
all	20	23	0.0112	0.35	0.172	0.109
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
8/50	0G	1.022	3.242	1.297	7	640: 100% 5/5 [00:53<00:00, 10.73s/it]
Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100% 1/1 [00:10<00:00, 10.17s/it]

```
      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       9/50         0G     0.9835       2.92      1.317          7        640: 100% 5/5 [00:55<00:00, 11.01s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.08s/it]
                   all         20         23     0.0101      0.483      0.184     0.0993

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      10/50         0G      1.083      2.837      1.398         10        640: 100% 5/5 [00:50<00:00, 10.20s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.80s/it]
                   all         20         23    0.00896       0.55        0.2      0.121

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      11/50         0G      1.148       2.79      1.432          8        640: 100% 5/5 [00:53<00:00, 10.68s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.70s/it]
                   all         20         23    0.00665      0.528      0.234      0.128

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      12/50         0G      1.059      2.659      1.355         11        640: 100% 5/5 [00:55<00:00, 11.18s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.06s/it]
                   all         20         23      0.956      0.111      0.275      0.162

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      13/50         0G          1      2.633      1.307         10        640: 100% 5/5 [00:52<00:00, 10.55s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.43s/it]
                   all         20         23      0.939      0.102      0.293      0.149

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      14/50         0G     0.9228      2.427      1.221          9        640: 100% 5/5 [00:54<00:00, 10.88s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.86s/it]
                   all         20         23      0.902     0.0698      0.289      0.162

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      15/50         0G      1.019      2.577      1.323          9        640: 100% 5/5 [00:51<00:00, 10.33s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.05s/it]
                   all         20         23      0.945     0.0889      0.288      0.188

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      16/50         0G      1.045      2.552      1.324          7        640: 100% 5/5 [00:51<00:00, 10.32s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.95s/it]
                   all         20         23      0.989      0.133      0.332      0.231

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      17/50         0G     0.9874      2.466      1.316          8        640: 100% 5/5 [00:51<00:00, 10.27s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.03s/it]
                   all         20         23      0.953     0.0444       0.29      0.195

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      18/50         0G     0.9376       2.38      1.242         10        640: 100% 5/5 [00:49<00:00,  9.98s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.94s/it]
                   all         20         23          1     0.0859      0.303      0.194
```

```
               all         20         23          1      0.6839      0.383      0.194

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      19/50         0G     0.9401      2.253      1.297         10        640: 100% 5/5 [00:50<00:00, 10.20s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.07s/it]
                   all         20         23      0.968      0.111      0.291      0.166

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      20/50         0G      0.974      2.258      1.268         10        640: 100% 5/5 [00:51<00:00, 10.32s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.82s/it]
                   all         20         23      0.809      0.156      0.352      0.186

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      21/50         0G      1.038      2.238      1.345         10        640: 100% 5/5 [00:52<00:00, 10.56s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.06s/it]
                   all         20         23      0.425      0.272      0.449      0.247

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      22/50         0G       1.08      2.294      1.333          9        640: 100% 5/5 [00:53<00:00, 10.73s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.03s/it]
                   all         20         23       0.42      0.334       0.49      0.307

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      23/50         0G      1.043        2.2      1.311         10        640: 100% 5/5 [00:53<00:00, 10.62s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:06<00:00,  6.13s/it]
                   all         20         23      0.425      0.426      0.516      0.298

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      24/50         0G     0.9075      2.006      1.288          8        640: 100% 5/5 [00:52<00:00, 10.43s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.02s/it]
                   all         20         23      0.757      0.358      0.466      0.314

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      25/50         0G     0.9806      2.124       1.32          8        640: 100% 5/5 [00:52<00:00, 10.46s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.71s/it]
                   all         20         23      0.931       0.35      0.516      0.325

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      26/50         0G     0.9299      1.965      1.197          9        640: 100% 5/5 [00:51<00:00, 10.24s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.07s/it]
                   all         20         23      0.956      0.337      0.514      0.325

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      27/50         0G     0.8406      1.918      1.195          7        640: 100% 5/5 [00:52<00:00, 10.40s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.42s/it]
                   all         20         23      0.882      0.414      0.501      0.315
```

```
      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      28/50         0G      0.838      1.908      1.203         13        640: 100% 5/5 [00:51<00:00, 10.35s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.38s/it]
                   all         20         23      0.698      0.406      0.487      0.284

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      29/50         0G     0.8111      1.766      1.171         13        640: 100% 5/5 [00:52<00:00, 10.54s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.09s/it]
                   all         20         23      0.778      0.369      0.467      0.272

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      30/50         0G     0.9026      1.875      1.237         11        640: 100% 5/5 [00:51<00:00, 10.21s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.77s/it]
                   all         20         23      0.639      0.422       0.49      0.282

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      31/50         0G     0.7816      1.661      1.171          9        640: 100% 5/5 [00:51<00:00, 10.29s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.05s/it]
                   all         20         23      0.661      0.398      0.416      0.256

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      32/50         0G     0.8095      1.695      1.204          7        640: 100% 5/5 [00:51<00:00, 10.23s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.93s/it]
                   all         20         23      0.986      0.356      0.499       0.32

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      33/50         0G     0.8428      1.718       1.21         11        640: 100% 5/5 [00:52<00:00, 10.54s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.04s/it]
                   all         20         23      0.774      0.451      0.484      0.296

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      34/50         0G     0.8615      1.831      1.207          9        640: 100% 5/5 [00:50<00:00, 10.17s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.85s/it]
                   all         20         23      0.661      0.417       0.45       0.29

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      35/50         0G     0.7944      1.611      1.182         10        640: 100% 5/5 [00:51<00:00, 10.36s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.05s/it]
                   all         20         23      0.717      0.394      0.443      0.286

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      36/50         0G     0.7611      1.555      1.153         12        640: 100% 5/5 [00:51<00:00, 10.26s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.81s/it]
                   all         20         23      0.659      0.372       0.48      0.326
```

```
      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      37/50         0G     0.7812      1.607      1.158         14        640: 100% 5/5 [00:51<00:00, 10.37s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.04s/it]
                   all         20         23        0.8      0.404      0.497      0.326

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      38/50         0G     0.8618      1.648      1.154          8        640: 100% 5/5 [00:50<00:00, 10.10s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.85s/it]
                   all         20         23      0.822      0.405      0.514      0.337

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      39/50         0G     0.7943      1.558      1.188          7        640: 100% 5/5 [00:52<00:00, 10.49s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.06s/it]
                   all         20         23      0.793      0.419      0.556      0.351

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      40/50         0G     0.8186      1.677      1.225          9        640: 100% 5/5 [00:53<00:00, 10.75s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.05s/it]
                   all         20         23      0.465      0.483      0.574      0.354
Closing dataloader mosaic
albumentations: Blur(p=0.01, blur_limit=(3, 7)), MedianBlur(p=0.01, blur_limit=(3, 7)), ToGray(p=0.01, num_output_channels=3, method='weighted_average'), CLAHE(p=0.01, clip_limit=(1.0, 4.0), tile_grid_size=(8, 8))

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      41/50         0G     0.6443      1.956      1.087          4        640: 100% 5/5 [00:52<00:00, 10.53s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:06<00:00,  6.06s/it]
                   all         20         23      0.479      0.511      0.568      0.347

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      42/50         0G     0.6431      1.863      1.076          4        640: 100% 5/5 [00:49<00:00,  9.95s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.03s/it]
                   all         20         23      0.492      0.517      0.576      0.378

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      43/50         0G     0.6011      1.881      1.022          4        640: 100% 5/5 [00:51<00:00, 10.27s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.98s/it]
                   all         20         23       0.56      0.483      0.574      0.381

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      44/50         0G     0.6398      1.895      1.108          4        640: 100% 5/5 [00:50<00:00, 10.17s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.04s/it]
                   all         20         23      0.583      0.468      0.554      0.372

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      45/50         0G     0.5908      1.838      1.071          5        640: 100% 5/5 [00:52<00:00, 10.46s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.86s/it]
                   all         20         23      0.582      0.474      0.553      0.373
```

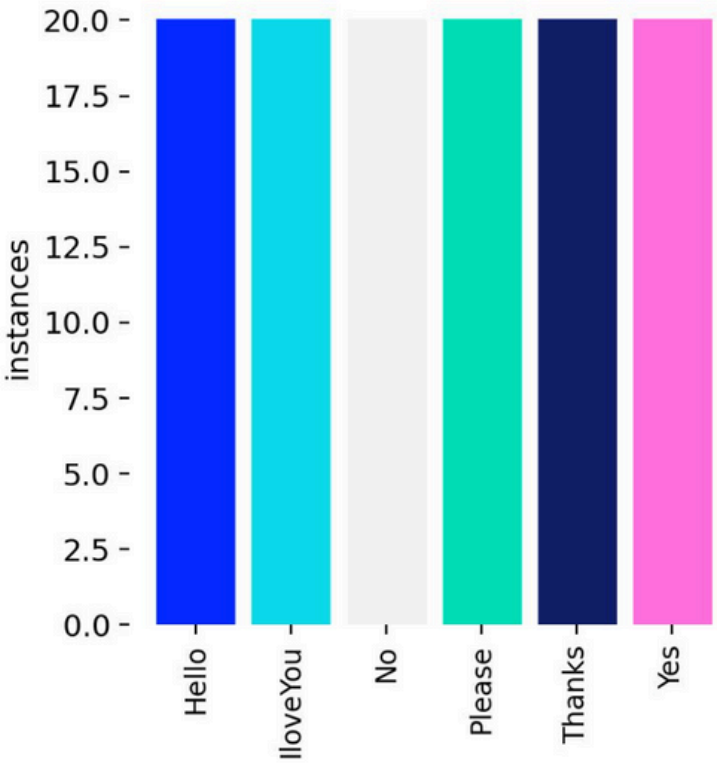
```
      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      46/50         0G     0.6039      1.927      1.017          4        640: 100% 5/5 [00:51<00:00, 10.23s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:04<00:00,  4.99s/it]
                   all         20         23      0.555      0.461      0.541      0.369

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      47/50         0G     0.5354      1.988     0.9897          4        640: 100% 5/5 [00:51<00:00, 10.32s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.89s/it]
                   all         20         23      0.539      0.443      0.496       0.34

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      48/50         0G     0.6194      1.921      1.025          5        640: 100% 5/5 [00:50<00:00, 10.07s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.04s/it]
                   all         20         23      0.667      0.408      0.517      0.353

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      49/50         0G     0.6661      2.013      1.043          4        640: 100% 5/5 [00:51<00:00, 10.26s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.94s/it]
                   all         20         23      0.662      0.427      0.517      0.353

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      50/50         0G      0.603      1.753      1.041          4        640: 100% 5/5 [00:50<00:00, 10.17s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.02s/it]
                   all         20         23      0.656      0.433      0.519      0.356

50 epochs completed in 0.815 hours.
Optimizer stripped from runs/detect/train3/weights/last.pt, 5.5MB
Optimizer stripped from runs/detect/train3/weights/best.pt, 5.5MB

Validating runs/detect/train3/weights/best.pt...
Ultralytics 8.3.94 🚀 Python-3.11.11 torch-2.6.0+cu124 CPU (Intel Xeon 2.20GHz)
YOLO11n summary (fused): 100 layers, 2,583,127 parameters, 0 gradients, 6.3 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:05<00:00,  5.28s/it]
                   all         20         23      0.559      0.484      0.573       0.38
                 Hello          4          4      0.586       0.25      0.446      0.301
             Love_you          9          9      0.737      0.889      0.956      0.709
                    No          1          1          0          0          0          0
             Thank_you          3          3       0.69      0.667      0.665      0.433
                   Yes          3          6      0.784      0.613        0.8      0.459
Speed: 2.4ms preprocess, 242.7ms inference, 0.0ms loss, 6.1ms postprocess per image
Results saved to runs/detect/train3
💡 Learn more at https://docs.ultralytics.com/modes/train
```

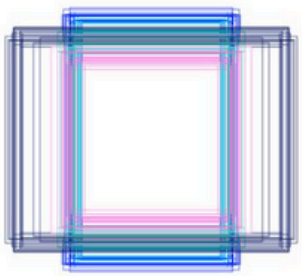
Chapter-5

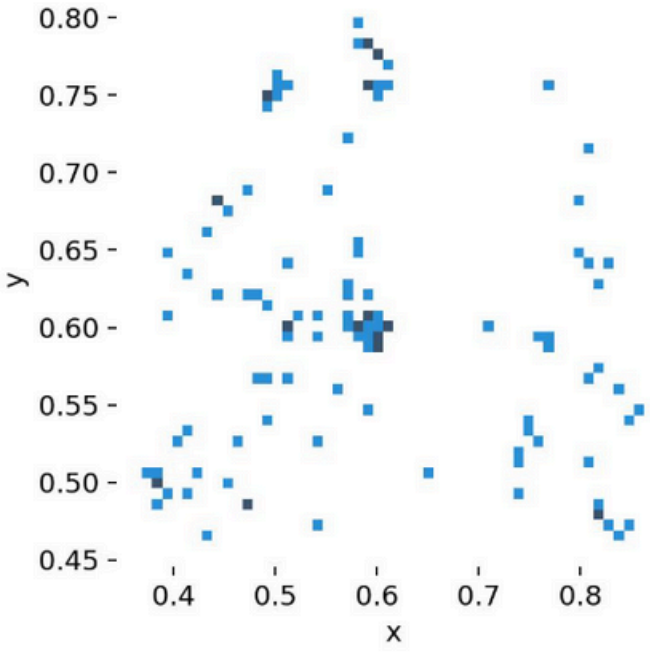
Results Visualization :-

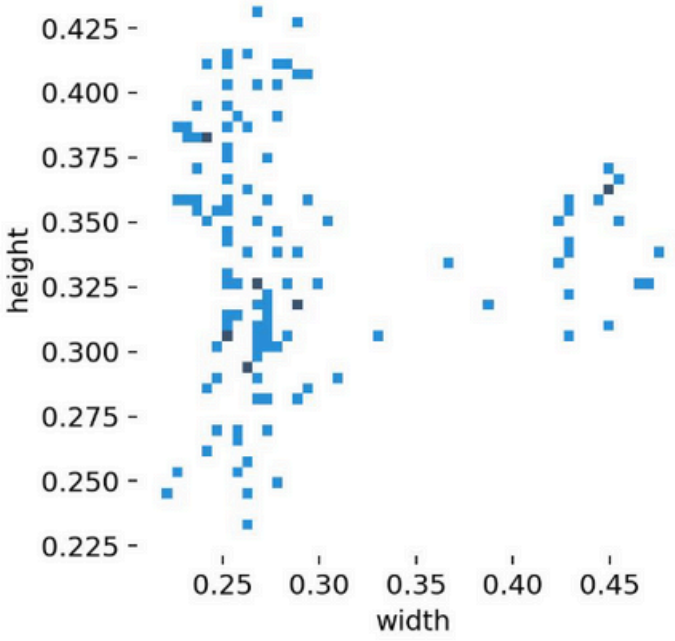
5.1Displaying Annotated Labels

Code:- **Image("/content/runs/detect/train/labels.jpg", width=600)**







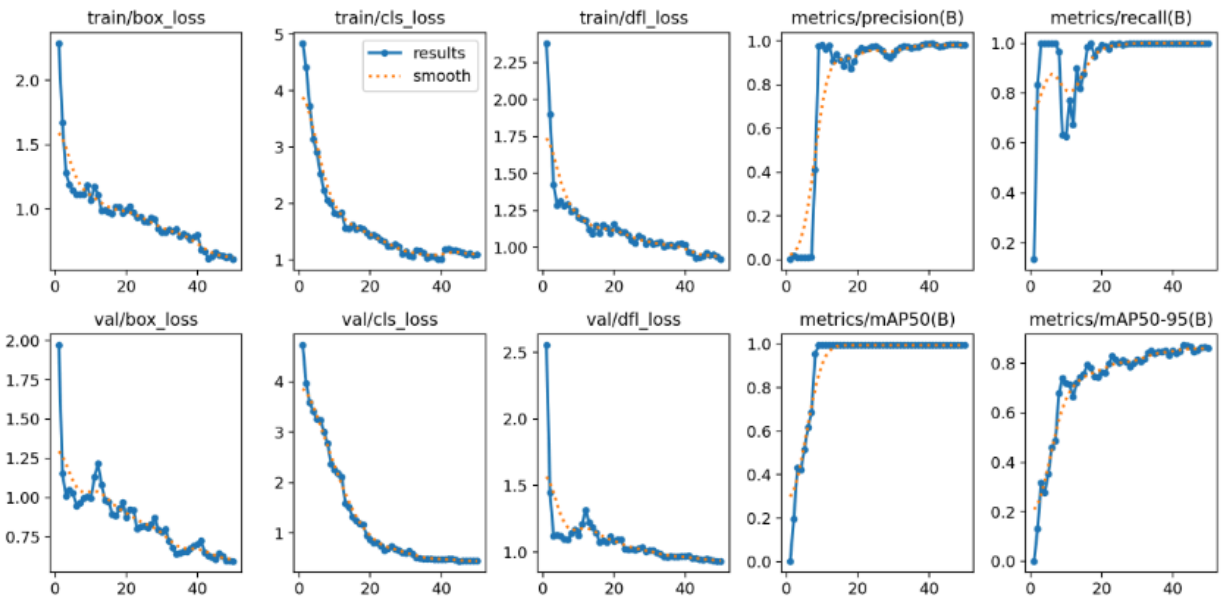


5.2 Training Performance Graphs

code:- Image("/content/runs/detect/train/results.png", width=600)

- **Loss Curves in figure** It shows how the model's prediction error is decreasing over time.
- **mAP (Mean Average Precision)**: Evaluates accuracy of detection and show in curve .

```
Image("/content/runs/detect/train/results.png", width=600)
```



6. Model Prediction on Video

After training the model, it is tested on a **video(capstone.mp4)** to verify its performance. The best-trained weights (best.pt) are used for prediction.(conf=0.25) is the Confidence threshold for detecting an object. (save=True) is used to save the detected results as output.

```
!yolo task=detect mode=predict model= "/content/runs/detect/train/weights/best.pt" conf=0.25 source="/content/runs/detect/train/weights/capstone.mp4" save=True
```

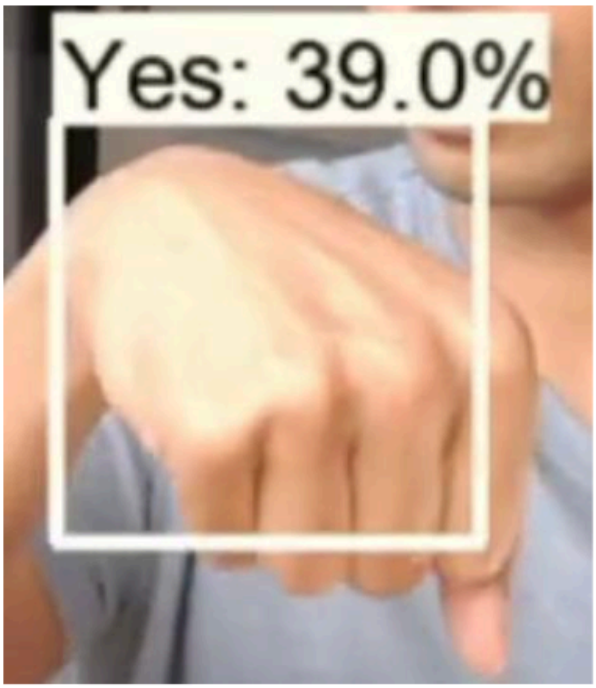
```
Ultralytics 8.3.94 🚀 Python-3.11.11 torch-2.6.0+cu124 CUDA:0 (Tesla T4, 15095MiB)
YOLO11n summary (fused): 100 layers, 2,583,322 parameters, 0 gradients, 6.3 GFLOPs
```

```
video 1/1 (frame 2/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 1 Hello, 9.7ms
video 1/1 (frame 3/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 1 Hello, 8.5ms
video 1/1 (frame 4/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 1 Hello, 9.3ms
video 1/1 (frame 5/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 1 Hello, 8.0ms
video 1/1 (frame 6/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 1 Hello, 8.2ms
video 1/1 (frame 7/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 1 Hello, 8.0ms
video 1/1 (frame 8/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 1 Hello, 8.0ms
video 1/1 (frame 9/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 1 Hello, 10.6ms
video 1/1 (frame 10/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 1 Hello, 7.9ms
video 1/1 (frame 11/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 (no detections), 8.0ms
video 1/1 (frame 12/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 1 Hello, 8.1ms
video 1/1 (frame 13/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 1 Hello, 7.9ms
video 1/1 (frame 14/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 1 Hello, 7.9ms
video 1/1 (frame 15/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 1 Hello, 10.2ms
video 1/1 (frame 16/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 1 Hello, 7.8ms
video 1/1 (frame 17/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 1 Hello, 8.5ms
video 1/1 (frame 18/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 1 Hello, 8.0ms
video 1/1 (frame 19/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 1 Hello, 8.2ms
video 1/1 (frame 20/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 1 Hello, 8.4ms
video 1/1 (frame 21/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 1 Hello, 8.5ms
video 1/1 (frame 22/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 1 Hello, 8.5ms
video 1/1 (frame 23/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 1 Hello, 8.1ms
video 1/1 (frame 24/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 1 Hello, 9.6ms
video 1/1 (frame 25/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 1 Hello, 10.4ms
video 1/1 (frame 26/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 1 Hello, 8.3ms
video 1/1 (frame 27/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 1 Hello, 8.0ms
video 1/1 (frame 28/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 1 Hello, 8.1ms
video 1/1 (frame 29/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 1 Hello, 8.1ms
video 1/1 (frame 30/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 1 Hello, 8.2ms
video 1/1 (frame 31/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 1 Hello, 8.0ms
video 1/1 (frame 32/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 1 Hello, 8.0ms
video 1/1 (frame 33/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 1 Hello, 7.9ms
video 1/1 (frame 34/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 1 Hello, 7.9ms

video 1/1 (frame 592/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 (no detections), 8.0ms
video 1/1 (frame 593/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 1 No, 8.0ms
video 1/1 (frame 594/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 1 No, 8.0ms
video 1/1 (frame 595/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 1 No, 7.9ms

video 1/1 (frame 613/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 (no detections), 8.1ms
video 1/1 (frame 614/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 (no detections), 8.6ms
video 1/1 (frame 615/615) /content/runs/detect/train/weights/capstone.mp4: 384x640 (no detections), 8.3ms
Speed: 2.5ms preprocess, 9.1ms inference, 1.3ms postprocess per image at shape (1, 3, 384, 640)
Results saved to runs/detect/predict3
```

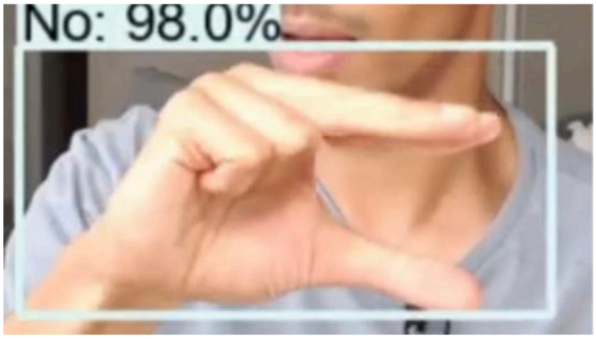
7. Predecting individual words through images

```
Image("/content/English-Sentences-1/train/images/No-sign-language1_31_jpeg.rf.f4c73e0601d40bfcd58387d276f55455.jpg", width=600)
```

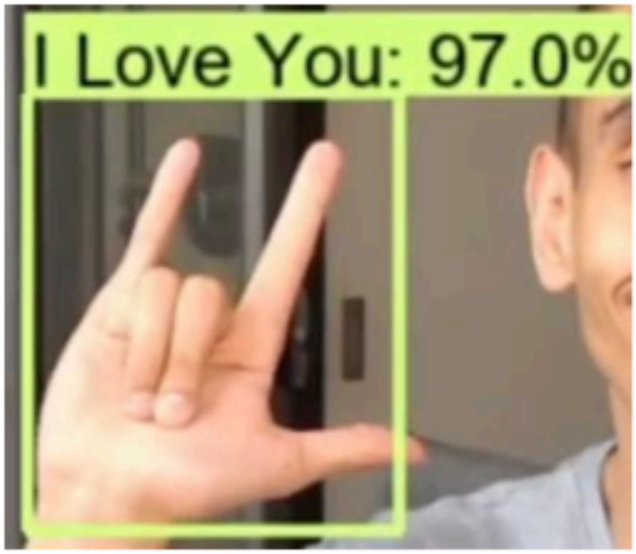

```
Image("/content/English-Sentences-1/train/images/WhatsApp-Image-2024-04-29-at-8-22-38-PM_jpeg.rf.cd9deea280307da3718b09593106a6df.jpg",width = 600)
```



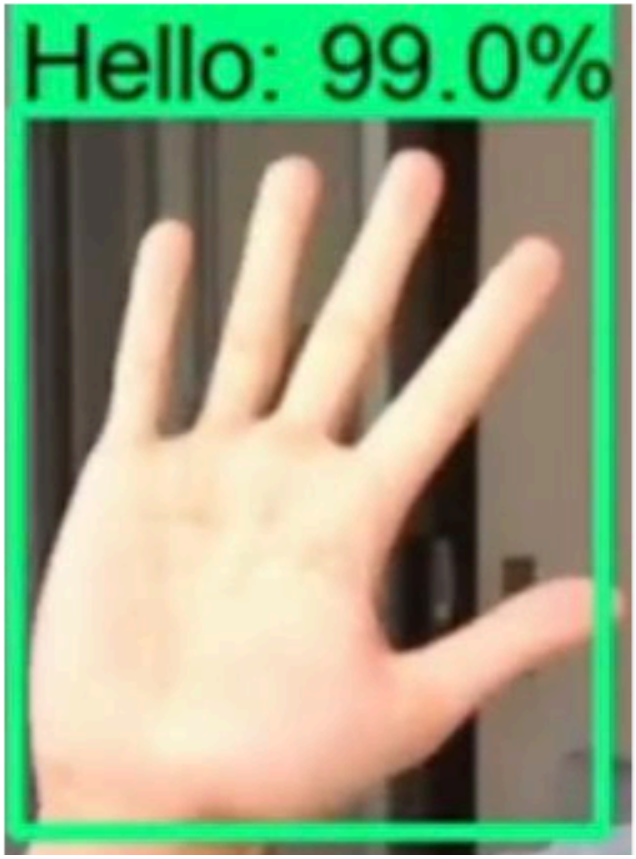

```
Image("/content/English-Sentences-1/train/images/WhatsApp-Image-2024-04-29-at-8-22-39-PM_jpeg.rf.f754c3f0f7b7403f504e0e4667715878.jpg",width = 600)
```

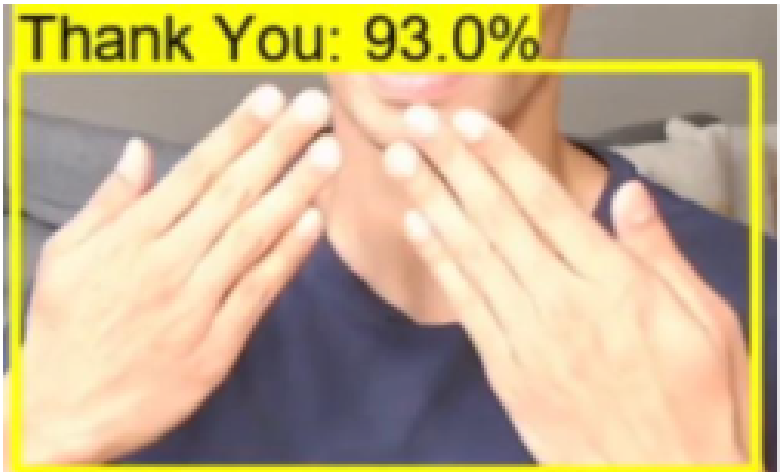


```
[ ] Image("/content/English-Sentences-1/train/images/WhatsApp-Image-2024-04-29-at-8-22-45-PM_jpeg.rf.6855bd7644027780b6d577758fdf4c6fa.jpg",width = 600)
```




```
[ ] Image("/content/English-Sentences-1/train/images/WhatsApp-Image-2024-04-29-at-8-22-46-PM-1-_jpeg.rf.5282f48b013873Sb11d5c393f544e51f4.jpg",width = 600)
```





```
Image("/content/English-Sentences-1/train/images/i-love-you-sign-language1_14_jpeg.rf.3223fe1b02985d39677b2f0ddaf37fff.jpg",width = 600)
```


```
Image("/content/English-Sentences-1/train/images/i-love-you-sign-language1_31_png.rf.88b43eb790d6482e262ffd125050e41f.jpg",width = 600)
```


```
Image("/content/English-Sentences-1/train/images/thank-you-sign-language1_58_jpeg.rf.7bf5b093a14f41d0e0ac614f6f13db91.jpg",width = 600)
```

8. challenges faced and Future Improvements

8.2 Challenges Faced during making project include :-

- **Dataset size limitations**: More data is required for improved accuracy.
- **Generalization**: The model may not perform well on unseen images if trained on a limited dataset.
- these problems can be solved with big data models and powerful GPU but it would not be feasible & cost effective compared to pre-existing solutions.

8.3 Future Scope

- **Deploying the model on embedded systems (Raspberry Pi, Jetson Nano)**.
- **Improving accuracy with larger and more diverse datasets**.
- **Integrating object tracking for real-time applications**.

9. Conclusion

The implemented object detection system based on YOLOv11 achieved successful sign recognition using the detection framework. The system processed a built-in dataset and conducted evaluation tests using operational video files. Deep learning object detection research demonstrates effective real-time performance which serves as a basis for advancing autonomous systems and smart surveillance and assistive technology development.