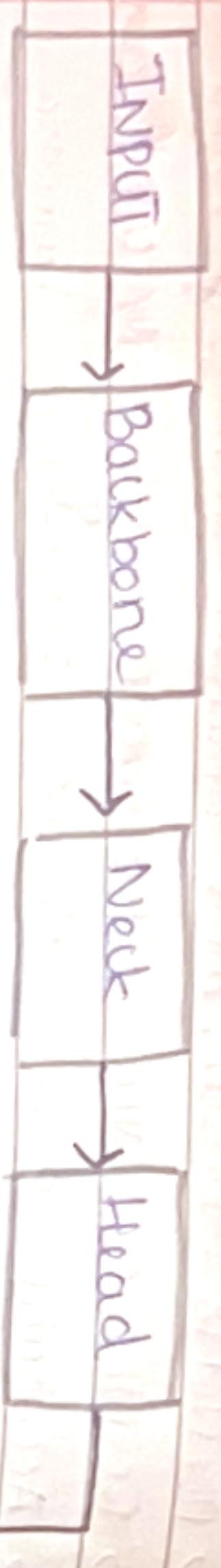


## YOLO v8:



LAB 15-

## Implement a YOLO Model to Detect Objects

27/10/25

### OUTPUT

Backbone: extracts essential visual feature from input using convolutional and CSP (Cross

Sieve Partial) Layers

Neck: connect backbone and head, combining multi-scale features using structures like PAN

Head: predict bounding boxes, object class, and confidence scores from processed features.

To implement a pre-trained YOLO (You Only Look Once) model for a real-time object detection using the analogous YOLOv8.

### OBJECTIVE:

1. Understand the concept and working principle of object detection using YOLO.
2. Use a pre-trained YOLOv8 model for detecting multiple objects in an image.
3. Visualize detection results and evaluate confidence and inference speed.
4. Interpret true detection accuracy and performance metrics.

~~Physical Implementation~~

YOLO: a deep learning-based real-time object detection algorithm. It divides the image into grids and predicts bounding boxes and class probabilities for each grid cell.

YOLOv8: Improved detection accuracy, speed optimization, and easy integration for inference and training.

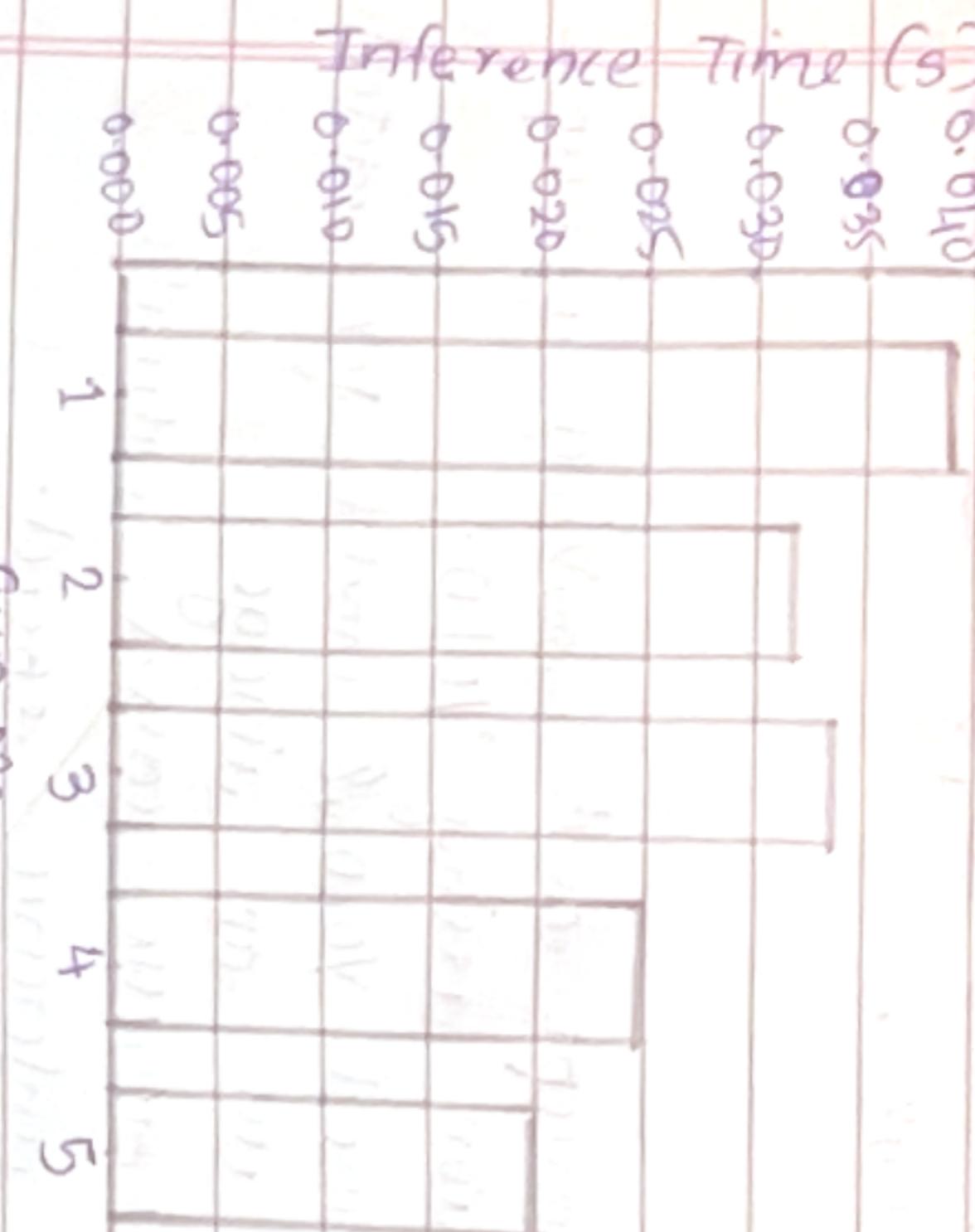
## OUTPUT:

26 Input Image whom draw a bounding box



## YOLO vs Object Detection

Inference Time



## PSEUDOCODE:

1. Install ultralytics YOLOv8 library
2. Load the YOLO vs pre-trained model (nano version)
3. Load an image for detection
4. Run detection on the image
5. Display the detected object with bounding boxes and labels
6. Measure inference time and confidence levels
7. Plot graphs

## OBSERVATION:

- > Model detects multiple objects in sample image
- > Each detected object is enclosed in a bounding box and labels
- > Inference Time Graph shows YOLO's high speed detection, maintaining low processing times across runs

- > Confidence Graph displays YOLO's confidence level for each detected object, showing reliability of prediction.

Tie

Person

Person

Person

Person

Person

Person

Person

Person

## RESULT:

Successfully implemented YOLO model for real-time object detection.