

Ex:-15:- Implement a YOLO model to detect object

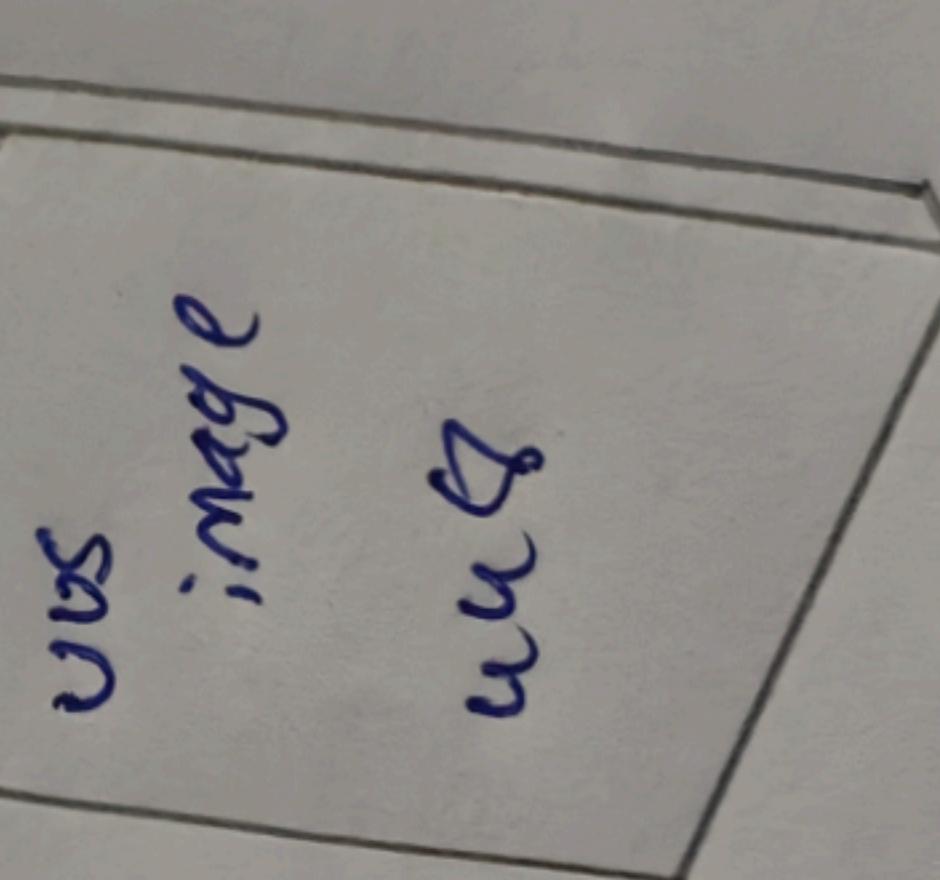
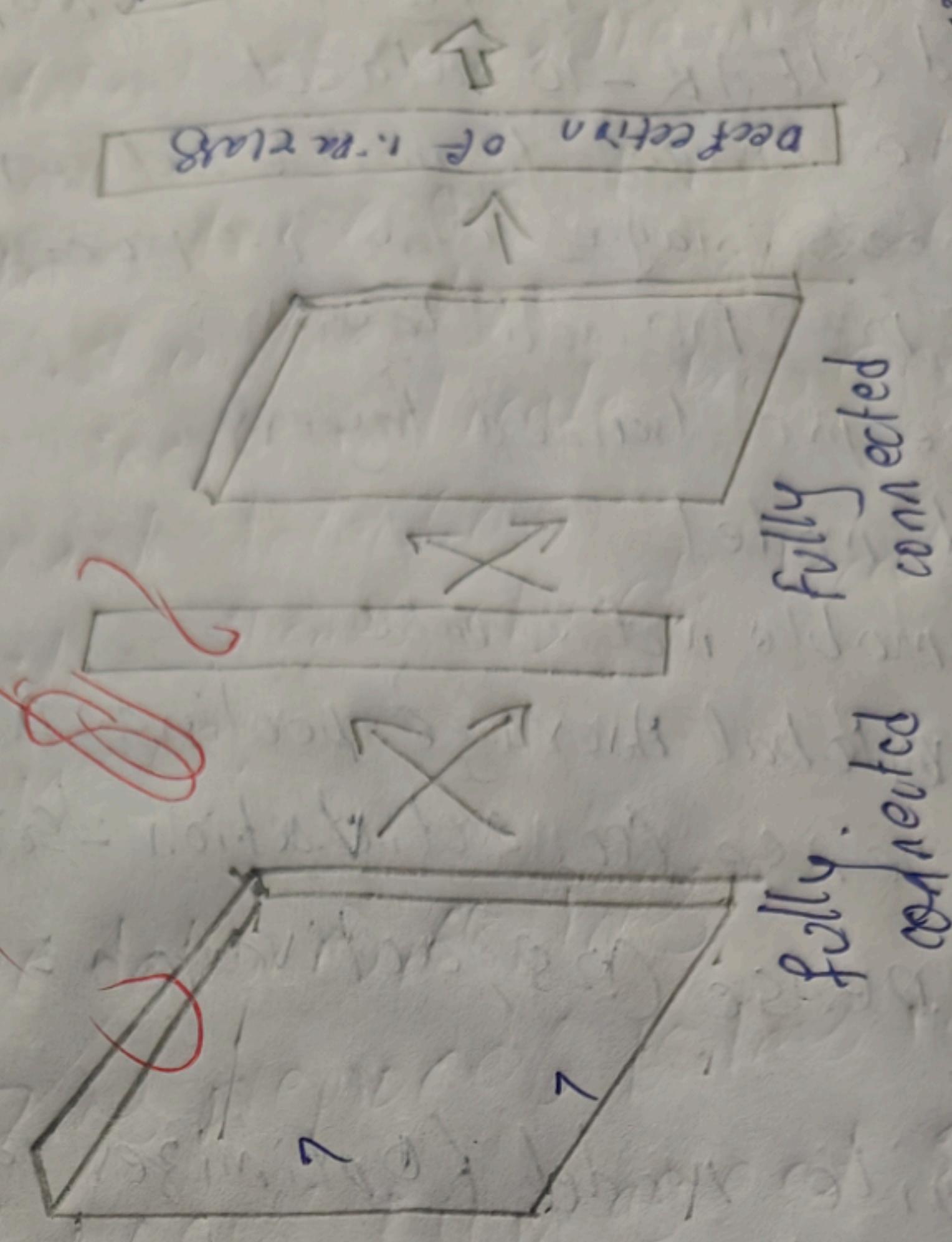
Aim:- to implement a yolov8 (you only look once) object detection model to detect and localize multiple objects in images using a pre-trained model.

objectives:-

- 1). Load a pre trained YOLOv8 model for object detection
- 2). Load and preprocess input images for the model
- 3). Run inference to detect objects in the image
- 4). Visualize detection results with bounding boxes, class labels and confidence scores.
- 5). Print the detection details (coordinates, classes, confidence) for further analysis.
- 6). Understand the workflow of a modern real-time object detection system.

## VLSI ARCHITECTURE

XOR customized Architecture,



62-9  
MAP  
45°C

xi-15

Aim:

Pseudo code:-

1. Install YOLOv8 library (ultralytics)
2. Import required libraries (YOLO, open cv, matplotlib)
3. Download a sample image.
4. Read & pre process the image
5. Load pre-trained YOLOv8 model.
6. Run inference on the image.
7. Get detection results.
  - Bounding boxes
  - class labels
  - confidence scores
8. Visualize the image with bounding boxes.
9. Print detection details.

Result:- successfully implemented a YOLO model to detect object.

KP

Output:-

1. the model detects objects in the image - e.g. people, sports objects.
2. Bounding boxes drawn around detected objects.
3. class tables show the type of object detected.
4. confidence scores indicate how certain the model is about each detection.
5. visualization shows objects localized accurately on the image.

~~448x640 (NO detection), 68, 1ms~~

Speed :- 10.7ms preprocess 68.1ms inference

0.8ms post process per image at shape (1, 3, 1440,

Welcome to Colab - Colab    Compression on MNIST dataset

colab.research.google.com/#scrollTo=EsuwzjwrgIk

Gmail Maps News Translate Disney+ Hotstar YouTube All Bookmarks

Welcome to Colab    Cannot save changes

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all Copy to Drive RAM Disk

Table of contents + Section

```
# Install ultralytics (YOLOv8)
!pip install ultralytics --quiet

# Import (module) ultralytics
from ultralytics import YOLO
import cv2
import matplotlib.pyplot as plt

# -----
# Step 1: Download a sample image
# -----
!wget -o sample.jpg https://github.com/ultralytics/assets/releases/download/v0.0.0/zidane.jpg

# -----
# Step 2: Read the image
# -----
image_path = "sample.jpg"
img = cv2.imread(image_path)

# Check if image is loaded correctly
if img is None:
    raise Exception("Image not found!")

# Convert BGR to RGB
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

# -----
# Step 3: Load pre-trained YOLOv8 model
# -----
model = YOLO("yolov8n.pt") # 'n' = nano, lightweight version
```

Variables Terminal ✓ 04:00 T4 (Python 3)

Welcome to Colab - Colab | Compression on MNIST dataset | +

colab.research.google.com/#scrollTo=Esuwzjwrglk

Gmail Maps News Translate Disney+ Hotstar YouTube All Bookmarks

Welcome to Colab Cannot save changes

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all Copy to Drive RAM Disk

Table of contents + Section

```
# -----  
# Step 4: Run object detection  
# -----  
results = model(img)  
  
# -----  
# Step 5: Display results  
# -----  
# Access the first result and call .plot() to get image with bounding boxes  
result_img = results[0].plot() # returns image with bounding boxes  
  
# Display with matplotlib  
plt.figure(figsize=(10,6))  
plt.imshow(result_img)  
plt.axis('off')  
plt.show()  
  
# -----  
# Step 6: Print detection details  
# -----  
for result in results:  
    boxes = result.bboxes.xyxy # bounding boxes (x1, y1, x2, y2)  
    conf = result.bboxes.conf # confidence scores  
    cls = result.bboxes.cls # class IDs  
    print("Bounding Boxes:", boxes)  
    print("Confidence Scores:", conf)  
    print("Class IDs:", cls)
```

-- 2025-10-26 22:30:09 -- https://github.com/ultralytics/assets/releases/download/v0.0.0/zidane.jpg

Variables Terminal ✓ 04:00 T4 (Python 3)

Welcome to Colab - Colab    Compression on MNIST dataset

colab.research.google.com/#scrollTo=Esuwzjwrglk

Welcome to Colab    Cannot save changes

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all Copy to Drive

RAM Disk

--2025-10-26 22:30:09-- https://github.com/ultralytics/assets/releases/download/v0.0.0/zidane.jpg  
Resolving github.com (github.com)... 20.205.243.166  
Connecting to github.com (github.com)|20.205.243.166|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Location: https://release-assets.githubusercontent.com/github-production-release-asset/521807533/416b5c91-abb1-49f0-8ab1-116c9093e950?sp=r&sv=2018-11-09&sr=b&spr=https&se=2025-10-26T23%3A08%3A17Z&rscd=attachment  
--2025-10-26 22:30:09-- https://release-assets.githubusercontent.com/github-production-release-asset/521807533/416b5c91-abb1-49f0-8ab1-116c9093e950?sp=r&sv=2018-11-09&sr=b&spr=https&se=2025-10-26T23%3A08%3A17Z  
Resolving release-assets.githubusercontent.com (release-assets.githubusercontent.com)... 185.199.108.133, 185.199.109.133, 185.199.110.133, ...  
Connecting to release-assets.githubusercontent.com (release-assets.githubusercontent.com)|185.199.108.133|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 50427 (49K) [application/octet-stream]  
Saving to: 'sample.jpg'  
  
sample.jpg 100%[=====] 49.25K --.-KB/s in 0.001s  
2025-10-26 22:30:09 (38.0 MB/s) - 'sample.jpg' saved [50427/50427]  
  
0: 384x640 2 persons, 18.3ms  
Speed: 2.2ms preprocess, 18.3ms inference, 5.5ms postprocess at shape (1, 3, 384, 640)  

```
Bounding Boxes: tensor([[ 749.4915,   42.5146, 1148.3696,  711.4277],  
                           [ 99.3115, 200.8951, 1110.9501,  713.4011]], device='cuda:0')  
Confidence Scores: tensor([0.60, 0.82], device='cuda:0')
```

Variables Terminal ✓ 04:00 T4 (Python 3)