Exp 15:

Implement a YOLO model to detect object.

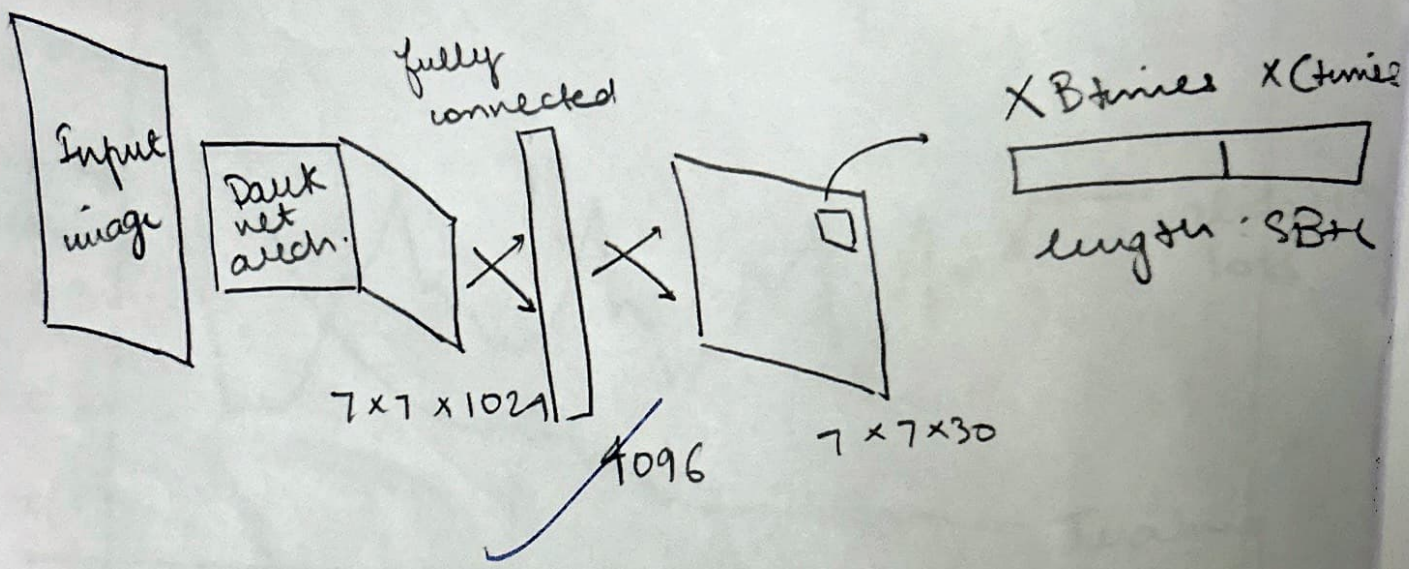
Aim: Implementing a YOLO (You only look once) model to detect object.

Pseudo code:

1. Import necessary libraries:
   - ultralytics (for YOLO model)
   - cv2 (for image handling)
   - matplotlib (for displaying results)
2. Load the pre-trained YOLO model (eg: "yolov8s.pt")
3. Read the input image from file
4. Pass the image to the YOLO model for detection.
   - YOLO divides image into grids
   - Each grids predicts:
     → bounding box (x, y, w, h)
     → confidence score
     → class label.
5. Apply Non-Maximum Suppression (NMS)
   - Removes overlapping boxes.
   - Keeps the one of highest confidence.
6. Display the image w/ detected bounding boxes and labels

Justification.

- YOLO (You only look once) detects multiple objects in one go, making it fast & efficient


Input image
Dark net arch.
fully connected
7×7×1024
4096
7×7×30
X Boxes
X Classes
length: SBC

- It uses a single CNN to predict bounding boxes & class probabilities simultaneously.
- Suitable for real-time application like CCTV, autonomous driving, & surveillance.

Result: Program implement successfully.

Output

Detected : Dog | Confidence : 0.79

Detected : Cat | Confidence : 0.77

Detected : Cat | Confidence : 0.37

Safari File Edit View History Bookmarks Window Help Tue Nov 4 7:58 PM

colab.research.google.com

clean coal technology diagrams - Google Sea... | SRM Student Portal | Untitled8.ipynb - Colab | 21AIC301J- Lab AI-A | YOLO object detection metrics

Untitled8.ipynb Saving failed since 7:38 PM

File Edit View Insert Runtime Tools Help

Share A

Commands + Code + Text Run all

RAM Disk

Next steps: Explain error

[13] 1s

```
from ultralytics import YOLO
import cv2
import matplotlib.pyplot as plt

model = YOLO("yolov8s.pt")

img = cv2.imread("sample.jpg")

results = model(img)
annotated = results[0].plot()
annotated = cv2.cvtColor(annotated, cv2.COLOR_BGR2RGB)

plt.imshow(annotated)
plt.axis("off")
plt.show()
```

```
WARNING ⚠ 'source' is missing. Using 'source=/usr/local/lib/python3.12/dist-packages/ultralytics/assets'.

image 1/2 /usr/local/lib/python3.12/dist-packages/ultralytics/assets/bus.jpg: 640x480 4 persons, 1 bus, 792.0ms
image 2/2 /usr/local/lib/python3.12/dist-packages/ultralytics/assets/zidane.jpg: 384x640 2 persons, 1 tie, 635.2ms
Speed: 8.8ms preprocess, 713.6ms inference, 1.8ms postprocess per image at shape (1, 3, 384, 640)
```

[11] 0s

```
image_files = ['/path/to/your/image1.jpg', '/other/path/image2.jpg']
```

[4] 8s

```
%pip install -U ultralytics
```

Variables Terminal 7:58 PM Python 3