Name – Akshat Bhargava

Roll no 22563

ANN – SL 2

Prac 10

Code –

import tensorflow as tf

import numpy as np

import cv2

# Load the pre-trained model

model = tf.saved\_model.load('D:\\faster\_rcnn\_resnet50\_v1\_640x640\_coco17\_tpu-8\\saved\_model')

# Load the label map

label\_map = {1: 'person', 3: 'car', 18: 'dog'}  # Replace with your label map

# Load and preprocess the input image

image = cv2.imread('D:\\image.jpg')

image = cv2.cvtColor(image, cv2.COLOR\_BGR2RGB)

# Convert to uint8 (Fix for TypeError)

image = tf.convert\_to\_tensor(image, dtype=tf.uint8)

# Expand dimensions for batch processing

image = tf.expand\_dims(image, axis=0)

# Run object detection

detections = model(image)

# Extract results

boxes = detections['detection\_boxes'][0].numpy()

scores = detections['detection\_scores'][0].numpy()

classes = detections['detection\_classes'][0].numpy().astype(int)

# Set confidence threshold

threshold = 0.5

filtered\_boxes = boxes[scores >= threshold]

filtered\_classes = classes[scores >= threshold]

# Convert tensor back to numpy for OpenCV drawing

image\_np = image.numpy()[0]

# Draw bounding boxes and labels

for box, cls in zip(filtered\_boxes, filtered\_classes):

    ymin, xmin, ymax, xmax = box

    h, w, \_ = image\_np.shape

    ymin, xmin, ymax, xmax = int(ymin \* h), int(xmin \* w), int(ymax \* h), int(xmax \* w)

    label = label\_map.get(cls, 'Unknown')

    cv2.rectangle(image\_np, (xmin, ymin), (xmax, ymax), (0, 255, 0), 2)

    cv2.putText(image\_np, label, (xmin, ymin - 10), cv2.FONT\_HERSHEY\_SIMPLEX, 0.9, (0, 255, 0), 2)

# Display the image with detections

cv2.imshow('Object Detection', image\_np)

cv2.waitKey(0)

cv2.destroyAllWindows()

Output –





 