import cv2

import numpy as np

# Load YOLO

def load\_yolo():

net = cv2.dnn.readNet("yolov3.weights", "yolov3.cfg")

layer\_names = net.getLayerNames()

output\_layers = [layer\_names[i[0] - 1] for i in net.getUnconnectedOutLayers()]

return net, output\_layers

# Load COCO class labels

def load\_labels():

with open("coco.names", "r") as f:

classes = [line.strip() for line in f.readlines()]

return classes

# Process each frame

def process\_frame(frame, net, output\_layers, classes):

height, width, channels = frame.shape

blob = cv2.dnn.blobFromImage(frame, 0.00392, (416, 416), (0, 0, 0), True, crop=False)

net.setInput(blob)

outs = net.forward(output\_layers)

class\_ids = []

confidences = []

boxes = []

for out in outs:

for detection in out:

for obj in detection:

scores = obj[5:]

class\_id = np.argmax(scores)

confidence = scores[class\_id]

if confidence > 0.5:

center\_x = int(obj[0] \* width)

center\_y = int(obj[1] \* height)

w = int(obj[2] \* width)

h = int(obj[3] \* height)

x = int(center\_x - w / 2)

y = int(center\_y - h / 2)

boxes.append([x, y, w, h])

confidences.append(float(confidence))

class\_ids.append(class\_id)

indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4)

for i in indexes:

i = i[0]

box = boxes[i]

x, y, w, h = box

label = str(classes[class\_ids[i]])

confidence = confidences[i]

color = (0, 255, 0) # Green

cv2.rectangle(frame, (x, y), (x + w, y + h), color, 2)

cv2.putText(frame, f"{label} {confidence:.2f}", (x, y - 10), cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, color, 2)

return frame

# Main function

def main():

# Load YOLO

net, output\_layers = load\_yolo()

classes = load\_labels()

# Open video capture

cap = cv2.VideoCapture(0) # Use 0 for webcam, or replace with video file path

while True:

ret, frame = cap.read()

if not ret:

break

frame = process\_frame(frame, net, output\_layers, classes)

cv2.imshow("Traffic Detection", frame)

if cv2.waitKey(1) & 0xFF == ord('q'):

break

cap.release()

cv2.destroyAllWindows()

if \_\_name\_\_ == "\_\_main\_\_":

main()

Output:

Python 3.12.5 (tags/v3.12.5:ff3bc82, Aug 6 2024, 20:45:27) [MSC v.1940 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license()" for more information.

= RESTART: C:\Users\DivyaHarshini\AppData\Local\Programs\Python\Python312\traf.py

Traceback (most recent call last):

File "C:\Users\DivyaHarshini\AppData\Local\Programs\Python\Python312\traf.py", line 83, in <module>

main()

File "C:\Users\DivyaHarshini\AppData\Local\Programs\Python\Python312\traf.py", line 62, in main

net, output\_layers = load\_yolo()

File "C:\Users\DivyaHarshini\AppData\Local\Programs\Python\Python312\traf.py", line 6, in load\_yolo

net = cv2.dnn.readNet("yolov3.weights", "yolov3.cfg")

cv2.error: OpenCV(4.10.0) D:\a\opencv-python\opencv-python\opencv\modules\dnn\src\darknet\darknet\_importer.cpp:210: error: (-212:Parsing error) Failed to open NetParameter file: yolov3.cfg in function 'cv::dnn::dnn4\_v20240521::readNetFromDarknet'