

# test\_on\_videos

August 19, 2020

```
[ ]: ##AUG 19,2020
# This program is to test C2TSR model on videos
import cv2
import numpy as np
import time
import os
import re

# Load Yolo v3 model
net = cv2.dnn.readNet("../Model/yolov3_custom_54000.weights", "../miscFiles/
    ↳yolov3_custom.cfg")
classes = []
#load classes
path = r'C:\Users\subys\CTS2R\miscFiles\FinalC2TSR.names'
with open(path, "r") as f:
    classes = [line.strip() for line in f.readlines()]
layer_names = net.getLayerNames()
output_layers = [layer_names[i[0] - 1] for i in net.getUnconnectedOutLayers()]
colors = np.random.uniform(0, 255, size=(len(classes), 3))
writer = None
# Loading video
video=r'../Data/Test/IMG_8471.MOV'
cap = cv2.VideoCapture(video)
#Creating output file to save the video
filename = os.path.basename(video)
names = re.split('\.',filename)
OutputFile= names[0] + "_detectedtr.mp4"
font = cv2.FONT_HERSHEY_COMPLEX_SMALL
starting_time = time.time()
frame_id = 0
while True:
    _, frame = cap.read()
    frame = cv2.resize(frame, None, fx=0.4, fy=0.4)
    frame_id += 1

    height, width, channels = frame.shape
```

```

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

net.setInput(blob)
outs = net.forward(output_layers)

# Showing informations on the screen
class_ids = []
confidences = []
boxes = []
for out in outs:
    for detection in out:
        scores = detection[5:]
        class_id = np.argmax(scores)
        confidence = scores[class_id]
        if confidence > 0.3:
            # Object detected
            center_x = int(detection[0] * width)
            center_y = int(detection[1] * height)
            w = int(detection[2] * width)
            h = int(detection[3] * height)

            # Rectangle coordinates
            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, 0.8, 0.3)

for i in range(len(boxes)):
    if i in indexes:
        x, y, w, h = boxes[i]
        label = str(classes[class_ids[i]])
        confidence = confidences[i]
        color = colors[class_ids[i]]
        cv2.rectangle(frame, (x, y), (x + w, y + h), color, 2)
        cv2.putText(frame, label + " " + str(round(confidence, 2)), (x, y),
→font, 0.5, color, 1)

    if writer is None:
        fourcc = cv2.VideoWriter_fourcc(*'mp4v')
        writer = cv2.VideoWriter(OutputFile, fourcc, 30, (frame.shape[1],frame.
→shape[0]), True)

```

```

writer.write(frame)

elapsed_time = time.time() - starting_time
fps = frame_id / elapsed_time
cv2.putText(frame, "FPS: " + str(round(fps, 2)), (10, 50), font, 1, (0, 0, 255), 3)
cv2.imshow("Image", frame)
key = cv2.waitKey(1)
if key == 27:
    break
#memory free up
cap.release()
writer.release()
cv2.destroyAllWindows()

```