test_on_images

August 19, 2020

[ ]:
```
##AUG 19,2020
# This program is to test C2TSR model on images
import cv2
import numpy as np
import time
import glob
import random
# Load Yolo v3
net = cv2.dnn.readNet("../Model/yolov3_custom_last.weights", "../miscFiles/
 ↪yolov3_custom.cfg")
classes = []
#load class file
path = r'D:\Suby\Suby\CTS2R\miscFiles\C2TSR.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 image
path = r'D:\Suby\Suby\CTS2R\Scripts\testImages'
images_path = glob.glob(r"D:\Suby\Suby\CTS2R\Scripts\testImages\*")
font = cv2.FONT_HERSHEY_PLAIN
random.shuffle(images_path)
# loop through all the images
for img_path in images_path:
    # Loading image
    frame = cv2.imread(img_path)
    frame = cv2.resize(frame, None, fx=0.4, fy=0.4)


    height, width, channels = frame.shape

    # Detecting objects
    blob = cv2.dnn.blobFromImage(frame, 0.00392, (320, 320), (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 +␣
 ↪30), font, 3, color, 3)

    cv2.imshow("Image", frame)
    key = cv2.waitKey(0)

cv2.destroyAllWindows()
```