SmallSEOTools

PLAGIARISM SCAN REPORT

Words	447	Date	September 17, 2021
Characters	5213	Excluded URL	

47% Plagiarism	53% Unique	8 Plagiarized Sentences	9 Unique Sentences

Content Checked For Plagiarism

```
import numpy as np
import tensorflow as tf
import cv2
import pafy
import time
import youtube_dl
tf.compat.v1.disable_v2_behavior()
class DetectorAPI:
def __init__(self, path_to_ckpt):
self.path_to_ckpt = path_to_ckpt
self.detection_graph = tf.compat.v1.Graph()
with self.detection_graph.as_default():
od_graph_def = tf.compat.v1.GraphDef()
with tf.compat.v1.gfile.GFile(self.path_to_ckpt, 'rb') as fid:
serialized_graph = fid.read()
od_graph_def.ParseFromString(serialized_graph)
tf.compat.v1.import_graph_def(od_graph_def, name='')
self.default_graph = self.detection_graph.as_default()
self.sess = tf.compat.v1.Session(graph=self.detection_graph)
# Definite input and output Tensors for detection_graph
self.image_tensor = self.detection_graph.get_tensor_by_name('image_tensor:0')
# Each box represents a part of the image where a particular object was detected.
self.detection_boxes = self.detection_graph.get_tensor_by_name('detection_boxes:0')
# Each score represent how level of confidence for each of the objects.
# Score is shown on the result image, together with the class label.
self.detection_scores = self.detection_graph.get_tensor_by_name('detection_scores:0')
self.detection_classes = self.detection_graph.get_tensor_by_name('detection_classes:0')
self.num_detections = self.detection_graph.get_tensor_by_name('num_detections:0')
print(self.detection_scores)
def processFrame(self, image):
# Expand dimensions since the trained_model expects images to have shape: [1, None, None, 3]
image_np_expanded = np.expand_dims(image, axis=0)
# Actual detection.
start_time = time.time()
(boxes, scores, classes, num) = self.sess.run(
[self.detection_boxes, self.detection_scores, self.detection_classes, self.num_detections],
feed_dict={self.image_tensor: image_np_expanded})
```

```
end_time = time.time()
# print("Elapsed Time:", end_time-start_time)
im_height, im_width, _ = image.shape
boxes_list = [None for i in range(boxes.shape[1])]
for i in range(boxes.shape[1]):
boxes_list[i] = (int(boxes[0, i, 0] * im_height),
int(boxes[0, i, 1] * im_width),
int(boxes[0, i, 2] * im_height),
int(boxes[0, i, 3] * im_width))
return boxes_list, scores[0].tolist(), [int(x) for x in classes[0].tolist()], int(num[0])
def close(self):
self.sess.close()
self.default_graph.close()
if __name__ == "__main__":
model_path = 'frozen_inference_graph.pb'
odapi = DetectorAPI(path_to_ckpt=model_path)
# threshold defines the value of over which an identified pedestrian is recognized as a pedestrian
threshold = 0.7
webcam = 0
cctv = 'rtsp://192.168.0.169/live/ch00_1'
url = "https://www.youtube.com/watch?v=1EiC9bvVGnk"
video = pafy.new(url)
best = video.getbest(preftype="mp4")
cap = cv2.VideoCapture('pump.mp4')
car_cascade = cv2.CascadeClassifier('car.xml')
while True:
# capture frame by frame
r, img = cap.read()
img = cv2.resize(img, (1280, 720))
# convert video into gray scale of each frames
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
# detect cars in the video
cars = car_cascade.detectMultiScale(gray, 1.1, 3)
boxes, scores, classes, num = odapi.processFrame(img)
# Visualization of the results of a detection.
total_ped = 0
cs = 0
font = cv2.FONT_HERSHEY_SIMPLEX
# uncomment this for car detection
for (x, y, w, h) in cars:
cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)
cs += 1
for i in range(len(boxes)):
# Class 1 represents human
if classes[i] == 1 and scores[i] > threshold:
box = boxes[i]
cv2.rectangle(img, (box[1], box[0]), (box[3], box[2]), (255, 0, 0), 2)
total_ped = total_ped + 1
k = str(cs)
# uncomment this for car count
cv2.putText(img, 'Car count : ' + k, (0, 200), font, 2, (0, 0, 0), 5, cv2.LINE_AA)
total_ped = str(total_ped)
cv2.putText(img, 'Bike count : ' + total_ped, (0, 130), font, 2, (0, 0, 0), 5, cv2.LINE_AA)
cv2.imshow("preview", img)
key = cv2.waitKey(1)
if key & 0xFF == ord('q'):
break
cap.release()
cv2.destroyAllWindows()
```

Sources	Similarity
python - Tensorflow object detection API: output boxes for ...	25%

with detection_graph.as_default(): with tf.Session(graph=detection_graph) as sess: # Definite input and output Tensors for detection_graph image_tensor = detection_graph.get_tensor_by_name('image_tensor:0') # Each box represents a part of the image where a particular object was detected. https://stackoverflow.com/questions/48917447/tensorflow-object-detection-api-output-boxes-for-probability-less-than-50	
object detections class name Â· Issue #7458 Â· tensorflow ... Â· Each box represents a part of the image where a particular object was detected. detection_boxes = detection_graph.get_tensor_by_name('detection_boxes:0') Each score â€¦ https://github.com/tensorflow/models/issues/7458	50%
Score is shown on the result image together with the class label ... Score is shown on the result image together with the class label from EG 1 at 283-st. Louis Park Alc. https://www.coursehero.com/file/p8ta2j/Score-is-shown-on-the-result-image-together-with-the-class-label/	25%
tensorflow-human-detection.py - gists Â· GitHub # Each box represents a part of the image where a particular object was detected. self.detection_boxes = self.detection_graph.get_tensor_by_name ... https://gist.github.com/halil/36074a1d26902c8fe6e3364ca63ec1db	13%
#import libraries of python opencv import cv2 import numpy as np ... ... cap.read() #convert video into gray scale of each frames gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) #detect cars in the video cars = car_cascade. http://fab.academany.org/2018/labs/fablabpuebla/students/yvonne-lomasmontaudon/assets/car_detection.py	9%
visualize boxes in tensorflow object detection - Stack Overflow Feb 23, 2018 â€” ... detection_graph) # Visualization of the results of a detection. vis_util.visualize_boxes_and_labels_on_image_array(image_np, ... https://stackoverflow.com/questions/48942079/visualize-boxes-in-tensorflow-object-detection	9%
Vehicle detection in Python using OpenCV - CodeSpeedy Vehicle detection in Python using OpenCV. This program will detect the cars and the bicycles from given footage and draws rectangle over it. https://www.codespeedy.com/vehicle-detection-in-python-using-opencv/	9%