PROJE ADI: TENSORFLOW İLE NESNELER ARASINDAN İNSANLARI AYIRACAK NESNE TANIMA VE ALGILAMA PROJESİ

YIL: 2021

AD: ÖMER

SOYAD: YANOVA

NEDEN BU PROJE?

Ben etrafını izlemeyi ve olan bitene karşı sorular sormayı seven bir insanım. Bu sebepten ötürü sürekli sorular sorar ve ona cevaplar ararım. Bu kanıya nasıl vardığımı hızlı şekilde anlatmak istiyorum.

Biliyoruz ki son zamanlarda tüm dünyayı bir virüs sardı. Ben bu virüse karşı nasıl önlem alabilirim diye düşünmeye başladım, hani dedim ya ben sorunlara karşı düşünen ve onlara çözüm üretmeye çalışan birisiyim. Yine böyle bir sorun gelince karşıma ben de düşünmeden edemedim. Güzel bir şeyler yapmam lazım ve bazı şirketlerin benim geliştirdiğim projeye talip olması gerekiyordu, bunun için aklımda daima yapay zeka(deep learning) vardı. Ben de hocamızın bize böyle ödev vermesine çok memnun kalarak aklımda ki projeyi ortaya koydum.

Bir proje geliştirmek için illa bir nedene mi gerek var? Aslıda var ama bazıları ona gerek kalmadan kendileri için proje geliştiriyor. Ben problem olunca çözüm üretmeyi seven bir insan olduğum için böyle bir yola başvurma gereği hissettim.

Yani bu projeye başlama ve aklıma geleni gerçekleştirme sebebim bu…

Eskilerden beri projeler var aklımda, fakat bunlar son zamanlarda daha elle tutulur gözle görülür şeyler olmaya başladı. Böyle ödevler ve olanaklar verilince bana ben de döktüm ortaya ve çokta güzel oldu…

İnsanlara yardım etmek ne kadar güzel bir şey, bunu hissedince bunu yaşayınca sürekli yapasın geliyor. Hele bizim sektör, daha güzel. Hem öğreniyorsun hem de yardım ediyorsun daha güzel daha mantıklı hale geliyor…

Araştırma

Araştırma için birçok yol izledim, udemy üzerinden bir ders aldım ve aldığım dersi sonuna kadar izledim ve bu süreçte yaşadıklarımı şimdi teker teker anlatacağım.

İlk öncelikle video izleyeceğim ve başka eğitimler alacağım kesindi çünkü herkesin anlatımı farklıydı ben bunu biliyordum, hangi hocadan ne kaparsam benim için o kardı. Ben bunun farkında birisi olarak ilk önce eğitimleri araştırdım. Eğitimleri araştırdıktan sonra aralarından en iyi olanı(yani en azından benim için en iyi olanı) seçtim. Dersin adı, eğitimin adı: R-CNN,GAS VB UYGULAMALAR KULLANARAK, TENSORFLOW TEKNOLOJİSİ İLE YÜZ VE NESNE TANIMA EĞİTİMİ idi. Ve ben bu eğitimi iyi ki almışım dedim. Çok kaliteli bir eğitmen ve içerik editörü. Hocanın ismini kaynakça kısmında belirteceğim. Yani aslında ben eğitim sürecini bu şekilde atlattım. Bu ilk yaşadığım ve ilk yardım aldığım yer oldu. He bir de nasıl bu kadar hızlı bitirdim eğitimi, çünkü hoca eğitimde anlattığı kaynakları paylaşıyordu, ben de onlardan yardım aldım…

Hocanın yazdığı kodlar ile aynı olmasına rağmen bazı yerlerde hatalar aldım ve bu hataları internette aratarak çözüm buldum.

Yani bazı internet sitelerinden de yardım aldım. Çoğu internet sitesi karışık anlatıyordu ama ben doğru kaynakları bulmak için uğraştım ve her seferinde başarılı sonuçlara vardım.

Ve ek olarak yabancı üniversitelerin ve BTK akademinin paylaştığı kurslar vardı onları izleyerek eğitimimi ve yardım alacağım yerleri tamamladım.

Harekete Geçme

Benim projem insanlara yönelik bir şey olduğu için düşünmem gerekiyordu. Yukarda da bahsettiğim gibi sorunlara çözüm üretmesini seven birisi olduğum için soruna böylece çözüm bulmuş oldum. Ayrıca en son bahsettiğim yerde birçok yerden yardım aldım, aldığım eğitimler ve yardımlar bazen istediğim sonucu vermedi, böylece o sorunları çözmekle de uğraştım. Böylece tamamen projeyi ele almış oldum.

Python programlama diline en öncelerden beri merakım vardı, projeler geliştirmek ve ortaya bir şeyler koymak istiyordum. Sürekli yapmak istediğimi ve sonunda koyduğumu dile getiriyorum ama gerçekten öyle. Bu benim için grur verici ve güzel bir şey…

Mesela kullandığım R-CNN adında bir algoritma türünden yararlandım. Bu algoritmalar nesne tanıma algoritmaları için neredeyse vaz geçilmez diyebiliriz. Bu algoritmalar nesne tanıma algoritmalarında bütün işi yapan algoritmalardır. Bu algoritmaların çalışma prensibi basittir. Genel olarak R-CNN programları bir fotoğraf karesini alır ve her pikselini tarar ve şüpheli bulduğu yani bir nesne veya insanın olabileceği yerleri belirli filtrelerden geçir ve bize fotoğraftaki nesneleri çıktı olarak gösterir. Ben kendi nesne tanıma algoritmamda Faster R-CNN adlı bir R-CNN çeşidini kullandım. Bu R-CNN çeşidi ise daha gelişmiş bir filtrelemeye sahip ve daha hızlı bir şekilde bize sonuç sunuyor. Bu R-CNN çeşidinin bu kadar hızlı ve isabetli olmasının sebebi ise prosedürlerin sırasıdır. Normal R-CNN algoritmasında ilk fotoğraf taranıyor ama Faster R-CNN'de fotoğraf ilk filtreden geçiriliyor. Sonra filtreden geçirilen fotoğraf taranıyor ve şüpheli bölgeler çıktı olarak sunuluyor. Bu prosedür sırasının yaptığı fark çok büyüktür. Mesela normal R-CNN algoritması bize 49 saniyede çıktı verirken, Faster R-CNN 0.2 saniye gibi hızlı bir sürede bize çıktı veriyor. Bu nedenlerden dolayı ben algoritmamda Faster R-CNN kullandım ki bize anlık ve doğru sonuçlar versin.

Bu projede materyal olarak sadece bilgisayar ve çıkardığım notlarımı kullandım. Çünkü program yazmak materyallerle değil fikir ve bilgiyle olan bir şeydir. Aslında program yazmayı kitap yazmaya benzetebiliriz. Kitap ta düşünce, hayal gücü ve fikirlerle yazılır. Ama algoritmayı daha çok belli bir kitleye yönelik bir kitap olarak düşünebiliriz. Projemi yaparken kendimi algoritma mantığı ve programlama konusunda çok geliştirdim. Algoritma mantığı konusunda gelişmem günlük hayattaki düşünme becerimi de çok etkiledi. Artık birçok şeye daha yapıcı ve çözümsel bir biçimde yaklaşabiliyorum.

Hatadan Hata Çıkarmak

Ben bu projeyi yaparken süreç içerisinde sabırlı olmanın aslında ne kadar olayları çözücü olduğunun farkına vardım. Mesela algoritmamda bir kütüphane yüklenemediğinde hemen stres olmak yerine daha sabırla yaklaşıp problemin tam olarak nerede olduğunu bulmak ve onu çözmek için araştırma yapınca aslında sabrın projedeki büyük yerini farkına vardım. Algoritma işleri düşünce ve bilginin eseri olduğu kadar sabrın da eseri olduğunu kavradım. Örnek verecek olursak insanlar bazı büyük çaplı yapay zeka ve nesne tanıma datasetleri ve programları için yıllarını veriyor ve bu işler sabırla olmayacak şeyler değil. Ben bu projemi sabırlı olarak tamamladığımı söyleyebilirim. Hata çıkınca sabırla onu hallettim. Bu sayede stresime yenik düşüp daha çok sorunlar ortaya çıkarmadım.

Kaynakça

https://www.udemy.com/course/bilgisayar-gorusu/learn/lecture/9608942#overview

https://www.udemy.com/course/veri-bilimine-giris/learn/lecture/17750854#overview

indirme;

https://www.python.org/downloads/

https://www.anaconda.com/products/individual

https://www.jetbrains.com/pycharm/download/

http://www.thomas-krenn.com/en/wiki/Cmd_commands_under_Windows

yabancı üniversiteden izlediğim videolar;

https://www.youtube.com/watch?v=d14TUNcbn1k&list=PL3FW7Lu3i5JvHM8ljYj------zLfQRF3EO8sYv&index=4

https://www.youtube.com/watch?v=vT1JzLTH4G4&list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv&index=1

https://www.youtube.com/watch?v=OoUX-nOEjG0&list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv&index=2

https://www.youtube.com/watch?v=h7iBpEHGVNc&list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv&index=3

https://www.youtube.com/watch?v=bNb2fEVKeEo&list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv&index=5

https://www.youtube.com/watch?v=wEoyxE0GP2M&list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv&index=6

https://www.youtube.com/watch?v=_JB0AO7QxSA&list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv&index=7

https://www.youtube.com/watch?v=6SlgtELqOWc&list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv&index=8

https://www.youtube.com/watch?v=DAOcjicFr1Y&list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv&index=9

https://www.youtube.com/watch?v=6niqTuYFZLQ&list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv&index=10

https://www.youtube.com/watch?v=nDPWywWRIRo&list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv&index=11

https://www.youtube.com/watch?v=6wcs6szJWMY&list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv&index=12

https://www.youtube.com/watch?v=5WoItGTWV54&list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv&index=13

https://www.youtube.com/watch?v=lvoHnicueoE&list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv&index=14

https://www.youtube.com/watch?v=eZdOkDtYMoo&list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv&index=15

https://www.youtube.com/watch?v=CIfsB_EYsVI&list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv&index=16

numpy.org

https://medium.com/@yemreak/%EF%B8%8F-tensorflow-object-detection-api-ile-obje-bulma-5000d218bac3

Ekler

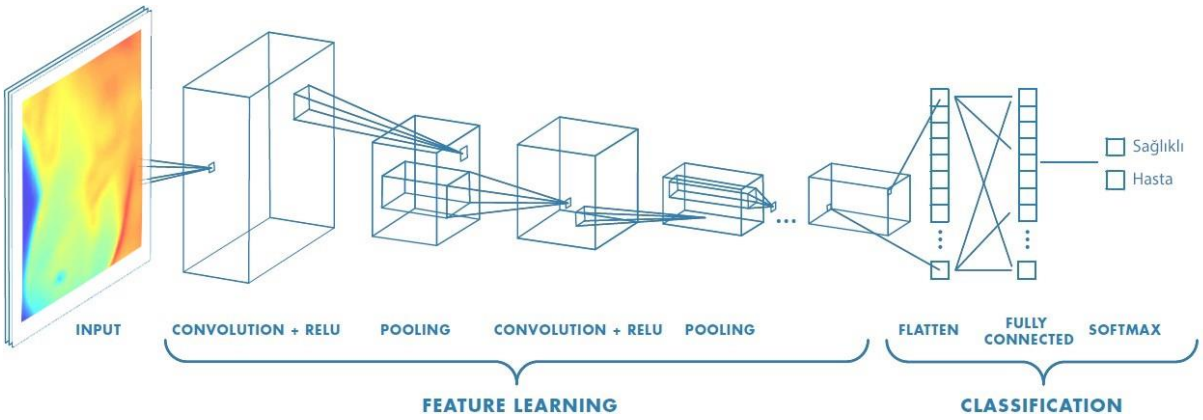
Tensorflow nedir?

TensorFlow, bir dizi görev arasında veri akışı ve türevlenebilir programlama için kullanılan ücretsiz ve açık kaynaklı bir yazılım kütüphanesidir. Sembolik bir matematik kütüphanesidir ve sinir ağları gibi makine öğrenimi uygulamaları için de kullanılır. Google'da hem araştırma hem de üretim için kullanılır.

R-CNN nedir?

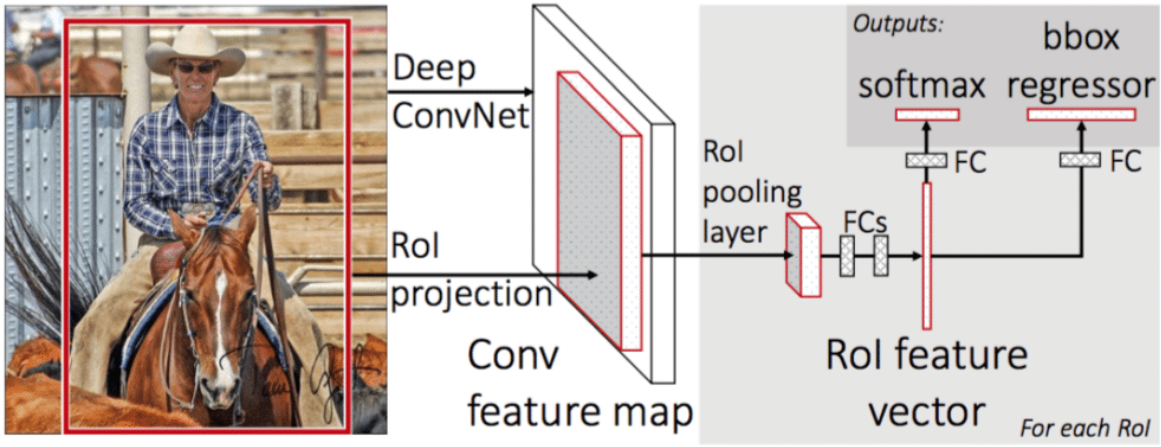
CNN nedir?

CNN resim ve video işleme için geliştirilen bir derin öğrenme ağ çeşididir. CNN genel olarak 4 katmandan oluşur. Bunlar: Convolution layer, Relu layer, Pooling layer ve Flattening layer. Convolution layer katmanında resmin üzerinde bir filtre gezdirilir. Filtre, bulunduğu piksellerde hesaplamalar yapar. Filtrenin gezdirilmesi sonucunda bir matris ortaya çıkar ve bu matrise “feature map” denir. Bir CNN ağında birden çok filtre gezdirilir ve filtreler özelliklerine göre şekillenir. Relu layer katmanı Convolution layer katmanından sonra gelir ve gelen veride eksi değerleri sıfır yapar. Bunun için Relu aktivasyon fonksiyonunu kullanır. Bu sayede 0 olan yerlerde bir nesnenin olmadığı sonucuna varılır. Pooling layer katmanında elimizdeki feature maplerinin boyutları küçültülerek havuzlama yapılır. Boyutlarının küçültülme nedeni ise elimizdeki parametrelerin sayısının azaltılması ve en marjinal ya da kritik parametreleri tutmaktır. Flattening layer katmanı elimizde matrisi tek bir vektöre çevirir ve neronlar aracılığıyla yapay sinir ağına aktarılır



R-CNN nedir?

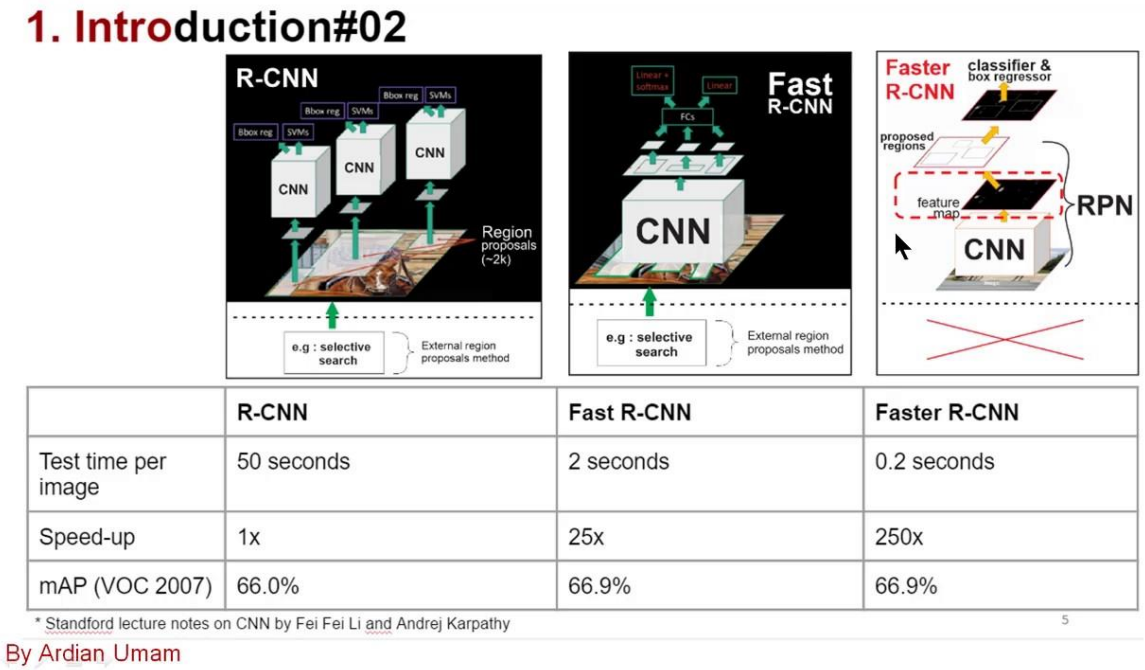
Resim öncelikle bölge önerilerine yani şüphe duyulan bölgelere bölünür. Daha sonra her bölge için CNN (ConvNet) uygulanır ve çıkan özellik mapleri için sınıflandırma algoritması olan SVM kullanılır. Sonrasında nesnenin var olup olmadığı kontrol edilip ayıklanır. Sonrasında linear regrasyon modellemesi ile bölgelerin boyutları belirlenir ve yapay sinir ağına gönderilir. Bu yöntemin en büyü sıkıntısı zamandır. Her bölge ayrı ayrı CNN'den geçirildiği için eğitim 84 saat sürer. Fotoğraflarda tahmin süresi ortalama 47 saniyedir.



Fast R-CNN nedir?

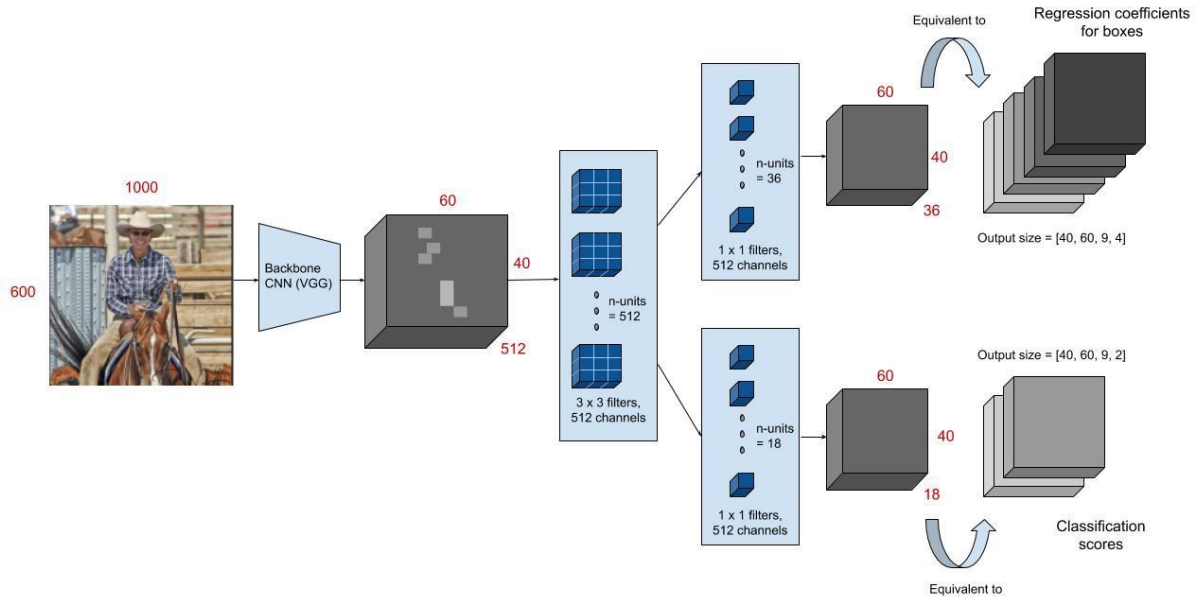
Bu yöntemin R-CNN'den farkı önce resmi bölgelere bölmek yerine CNN uygulanır ve bu sayede her bölge için ayrı ayrı uygulamaya gerek kalmaz. Sonrasında oluşan harita üzerinde bölge önerileri yapılır. Ayrıca sınıflandırma metodu olarak SVM yerine yapay sinir ağları katmanları içerisinde gerçekleşen softmax classifacition kullanılır.

CNN’ sadece bir kere kullandığı için zamandan tasarruf sağlar. Eğitim süresi 8.50 saattir. Tahmin süresi ise yaklaşık 2.5 saniyedir.



Faster R-CNN nedir?

Bu yöntemde de ilk CNN uygulanıp özellik haritası oluşturuluyor. Bölge önerileri kısmında seçici bölge araması yerine ayrı bir bölge önerisi ağı oluşturarak bölgeleri seçiyoruz. Geri kalan kısımlar Fast R-CNN ile neredeyse aynı. Bu teknikle tahmin süresini 0.3 saniyeye kadar düşürüyoruz.



Tensorflow Kurulumu

- Tensorflow anaconda üzerinden daha sağlıklı, taşınabilir ve verimli çalışabilmekte
- Anacondanın sanal ortamları, paketlerin çakışmasını engelleyecektir
-

Tensorflow CPU veya GPU Kurulumu

- Bu kurulum CPU kurulumu olarak da geçmekte
- GPU kurulumu CPU'ya nazaran oldukça hızlı eğitim seçeneği sağlar

- GPU kurulumu için gereksinimleri sağlıyorsanız GPU kurulumu (tensorflow-gpu) yapmanız tavsiye edilir

Sanal Ortam Oluşturma ve Üzerine Kurma

```
conda create -n tensorflow tensorflow # CPU kurulumu

conda create -n tensorflow tensorflow-gpu # GPU kurulumu
```

Gerekli Paketlerin Kurulumları

Tensorflow modellerini kullanabilmek için alttaki kurulumlara da ihtiyaç olabilmekte:

```
conda install opencv pillow matplotlib pandas jupyter
```

Modül bulunamaması gibi durumlarda `lxml`, `protobuf` paketlerini yüklemeyi deneyebilirsin.

OpenCv Kurulumu

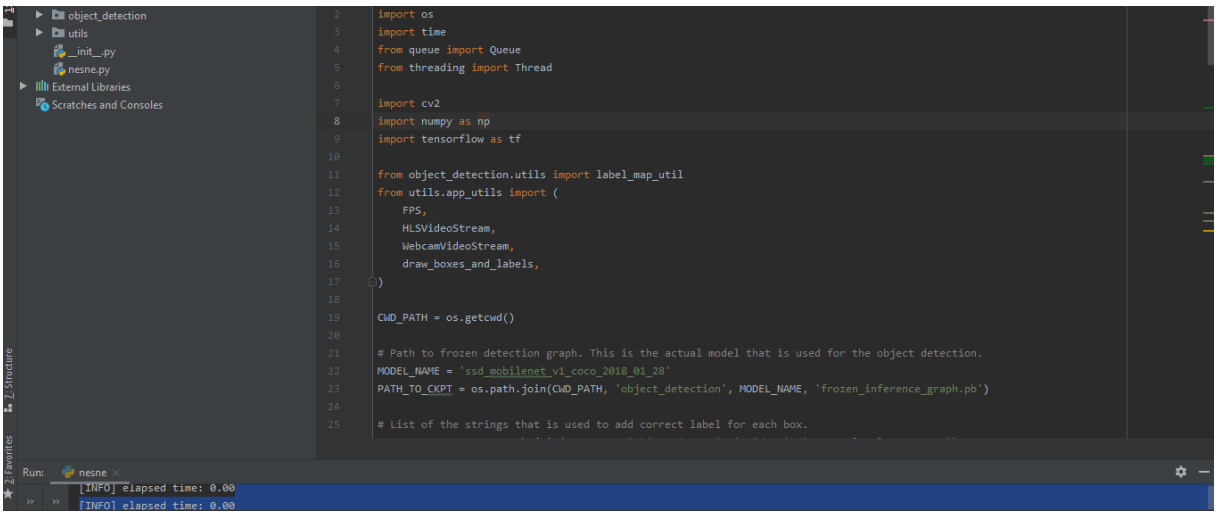
```
pip install opencv-contrib-python
```

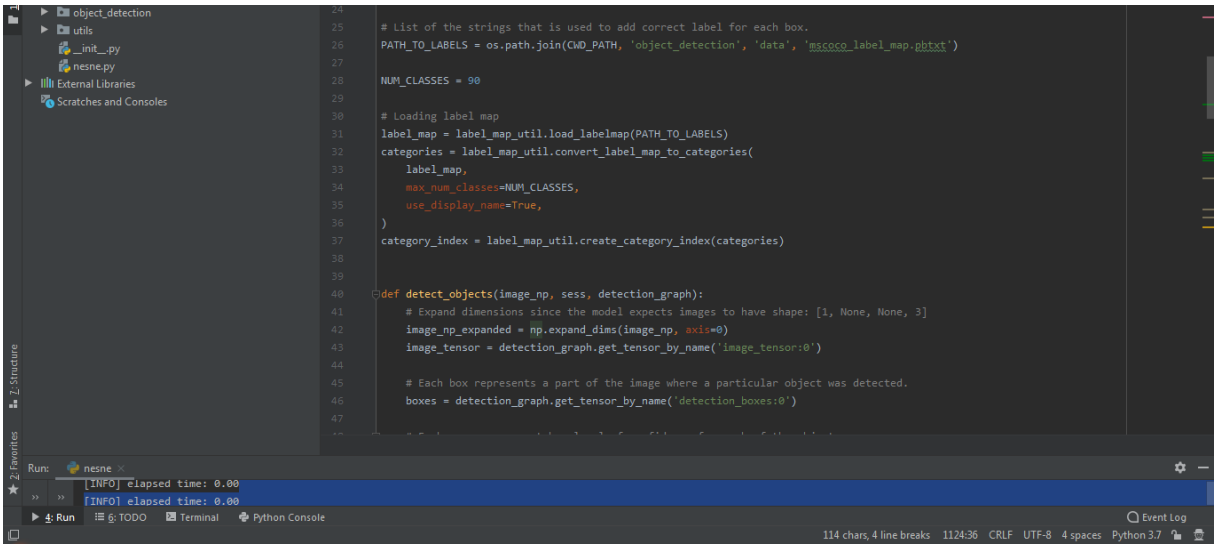
Script Dosyaları için Gerekli Modüller

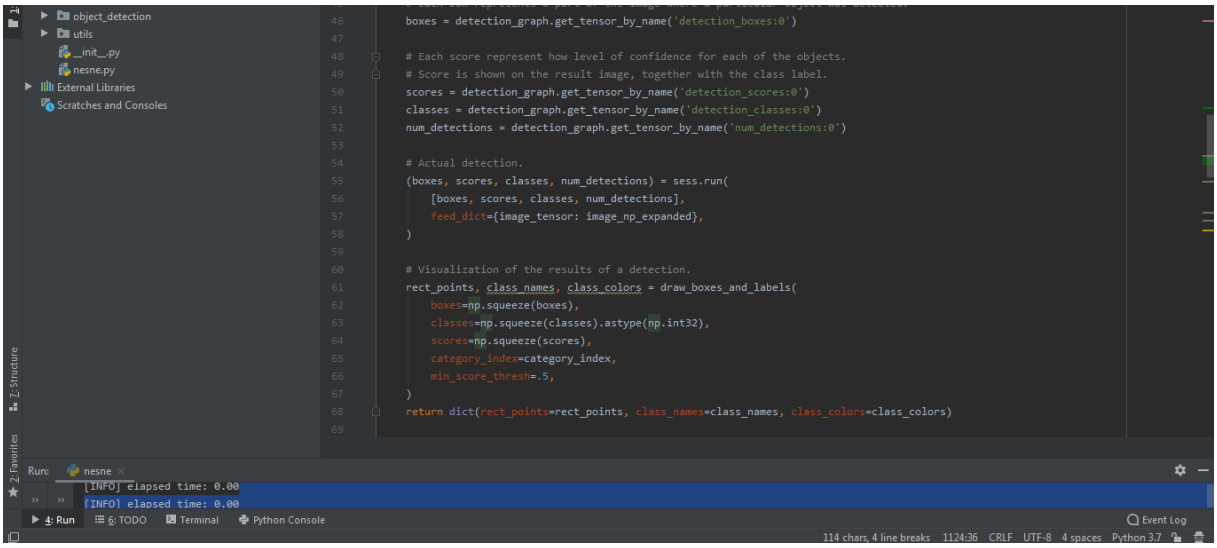
```
pip install pynput # detect_from_desktop
```

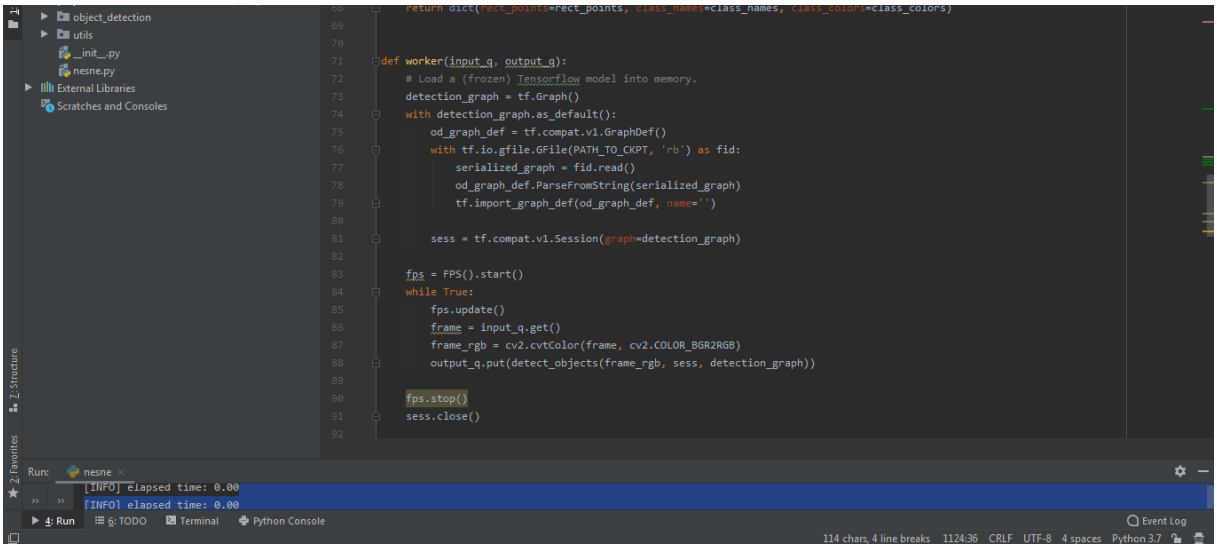
KAYNAK: https://medium.com/@yemreak/%EF%B8%8F-tensorflow-object-detection-api-ile-obje-bulma-5000d218bac3

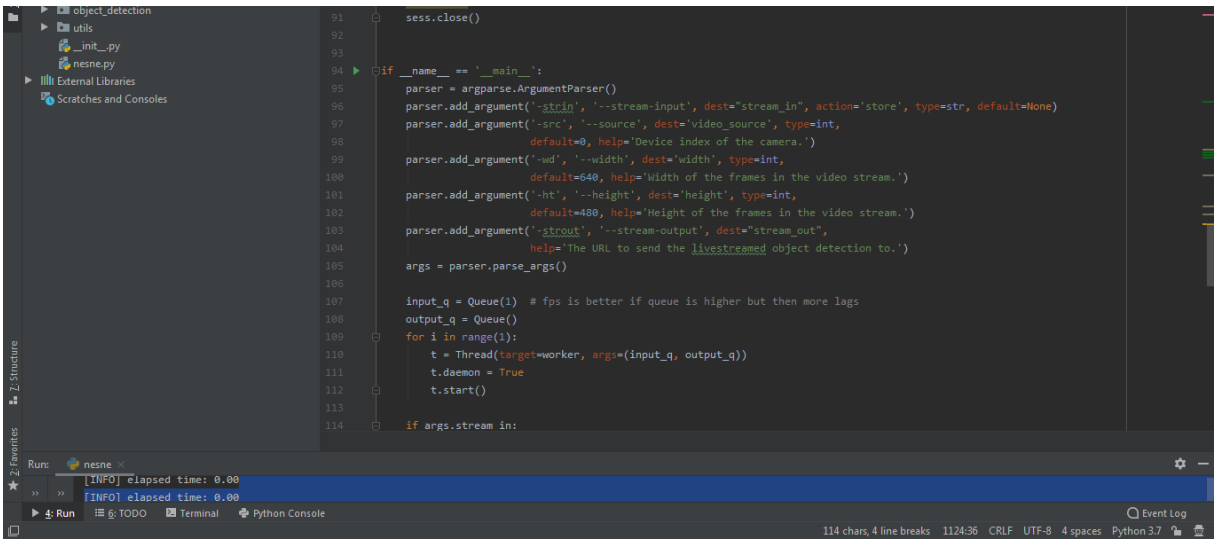
Kodlar

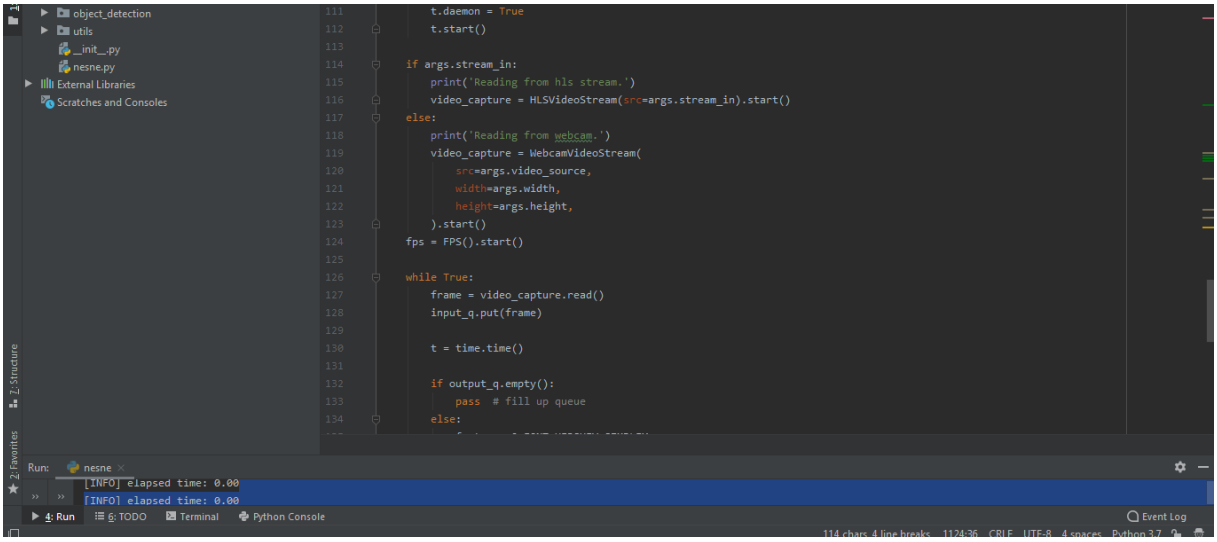

object_detection
utils
__init__.py
nesne.py
External Libraries
Scratches and Consoles
import os
import time
from queue import Queue
from threading import Thread
import cv2
import numpy as np
import tensorflow as tf
from object_detection.utils import label_map_util
from utils.app_utils import (
FPS,
HLSVideoStream,
WebcamVideoStream,
draw_boxes_and_labels,
)
CWD_PATH = os.getcwd()
# Path to frozen detection graph. This is the actual model that is used for the object detection.
MODEL_NAME = 'ssd_mobilenet_v1_coco_2018_01_28'
PATH_TO_CKPT = os.path.join(CWD_PATH, 'object_detection', MODEL_NAME, 'frozen_inference_graph.pb')
# List of the strings that is used to add correct label for each box.
Run:
nesne
[INFO] elapsed time: 0.00
[INFO] elapsed time: 0.00


object_detection
utils
__init__.py
nesne.py
External Libraries
Scratches and Consoles
# List of the strings that is used to add correct label for each box.
PATH_TO_LABELS = os.path.join(CWD_PATH, 'object_detection', 'data', 'mscoco_label_map.pbtxt')
NUM_CLASSES = 90
# Loading label map
label_map = label_map_util.load_labelmap(PATH_TO_LABELS)
categories = label_map_util.convert_label_map_to_categories(
label_map,
max_num_classes=NUM_CLASSES,
use_display_name=True,
)
category_index = label_map_util.create_category_index(categories)
def detect_objects(image_np, sess, detection_graph):
# Expand dimensions since the model expects images to have shape: [1, None, None, 3]
image_np_expanded = np.expand_dims(image_np, axis=0)
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.
boxes = detection_graph.get_tensor_by_name('detection_boxes:0')
Run:
nesne
[INFO] elapsed time: 0.00
[INFO] elapsed time: 0.00
4: Run
6: TODO
Terminal
Python Console
Event Log
114 chars, 4 line breaks
1124:36
CRLF
UTF-8
4 spaces
Python 3.7


object_detection
utils
__init__.py
nesne.py
External Libraries
Scratches and Consoles
boxes = 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.
scores = detection_graph.get_tensor_by_name('detection_scores:0')
classes = detection_graph.get_tensor_by_name('detection_classes:0')
num_detections = detection_graph.get_tensor_by_name('num_detections:0')
# Actual detection.
(boxes, scores, classes, num_detections) = sess.run(
[boxes, scores, classes, num_detections],
feed_dict={image_tensor: image_np_expanded},
)
Visualization of the results of a detection.
rect_points, class_names, class_colors = draw_boxes_and_labels(
boxes=np.squeeze(boxes),
classes=np.squeeze(classes).astype(np.int32),
scores=np.squeeze(scores),
category_index=category_index,
min_score_thresh=.5,
)
return dict(rect_points=rect_points, class_names=class_names, class_colors=class_colors)
Run:
nesne
[INFO] elapsed time: 0.00
[INFO] elapsed time: 0.00
4: Run
6: TODO
Terminal
Python Console
Event Log
114 chars, 4 line breaks
1124:36
CRLF
UTF-8
4 spaces
Python 3.7


object_detection
utils
__init__.py
nesne.py
External Libraries
Scratches and Consoles
return dict(rect_points=rect_points, class_names=class_names, class_colors=class_colors)
def worker(input_q, output_q):
# Load a (frozen) Tensorflow model into memory.
detection_graph = tf.Graph()
with detection_graph.as_default():
od_graph_def = tf.compat.v1.GraphDef()
with tf.io.gfile.GFile(PATH_TO_CKPT, 'rb') as fid:
serialized_graph = fid.read()
od_graph_def.ParseFromString(serialized_graph)
tf.import_graph_def(od_graph_def, name='')
sess = tf.compat.v1.Session(graph=detection_graph)
fps = FPS().start()
while True:
fps.update()
frame = input_q.get()
frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
output_q.put(detect_objects(frame_rgb, sess, detection_graph))
fps.stop()
sess.close()
Run:
nesne
[INFO] elapsed time: 0.00
[INFO] elapsed time: 0.00
4: Run
6: TODO
Terminal
Python Console
Event Log
114 chars, 4 line breaks
1124:36
CRLF
UTF-8
4 spaces
Python 3.7

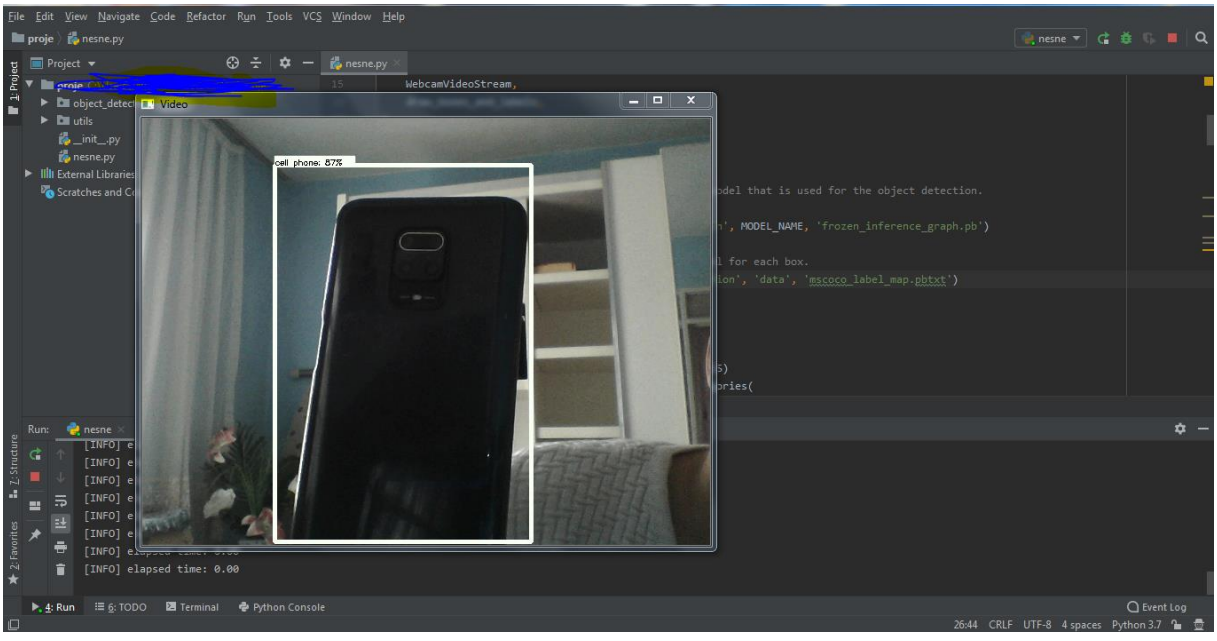

object_detection
utils
__init__.py
nesne.py
External Libraries
Scratches and Consoles
sess.close()
if __name__ == '__main__':
parser = argparse.ArgumentParser()
parser.add_argument('-strin', '--stream-input', dest="stream_in", action='store', type=str, default=None)
parser.add_argument('-src', '--source', dest='video_source', type=int,
default=0, help='Device index of the camera.')
parser.add_argument('-wd', '--width', dest='width', type=int,
default=640, help='Width of the frames in the video stream.')
parser.add_argument('-ht', '--height', dest='height', type=int,
default=480, help='Height of the frames in the video stream.')
parser.add_argument('-strout', '--stream-output', dest="stream_out",
help='The URL to send the livestreamed object detection to.')
args = parser.parse_args()
input_q = Queue(1) # fps is better if queue is higher but then more lags
output_q = Queue()
for i in range(1):
t = Thread(target=worker, args=(input_q, output_q))
t.daemon = True
t.start()
if args.stream in:
Run:
nesne
[INFO] elapsed time: 0.00
[INFO] elapsed time: 0.00
4: Run
6: TODO
Terminal
Python Console
Event Log
114 chars, 4 line breaks
1124:36
CRLF
UTF-8
4 spaces
Python 3.7

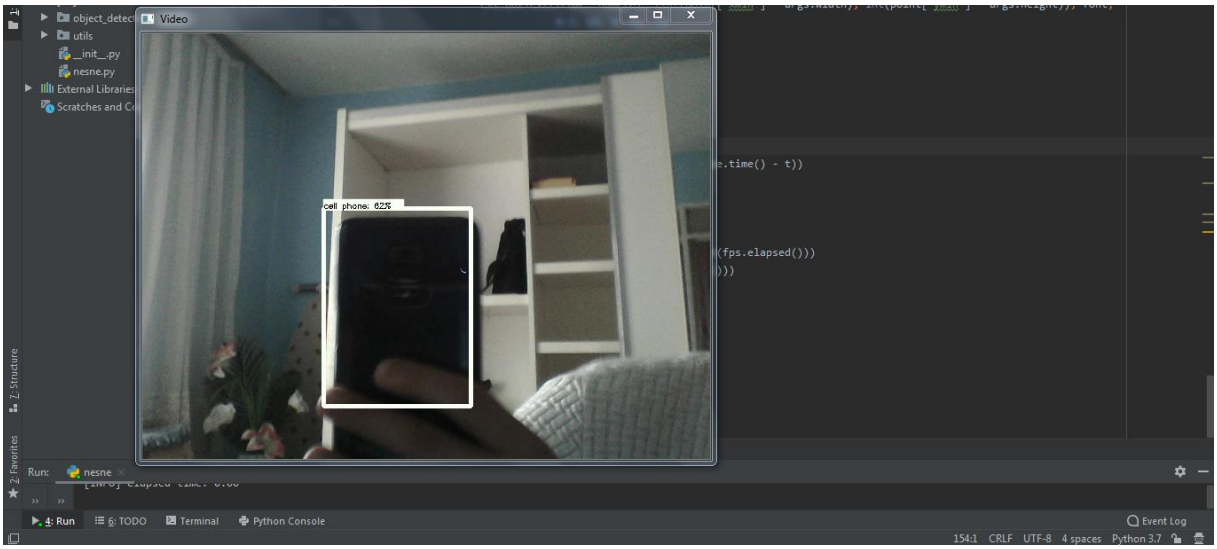

object_detection
utils
__init__.py
nesne.py
External Libraries
Scratches and Consoles
t.daemon = True
t.start()
if args.stream_in:
print('Reading from hls stream.')
video_capture = HLSVideoStream(src=args.stream_in).start()
else:
print('Reading from webcam.')
video_capture = WebcamVideoStream(
src=args.video_source,
width=args.width,
height=args.height,
).start()
fps = FPS().start()
while True:
frame = video_capture.read()
input_q.put(frame)
t = time.time()
if output_q.empty():
pass # fill up queue
else:
Run:
nesne
[INFO] elapsed time: 0.00
[INFO] elapsed time: 0.00
4: Run
6: TODO
Terminal
Python Console
Event Log

```
        if output_q.empty():
            pass  # fill up queue
        else:
            font = cv2.FONT_HERSHEY_SIMPLEX
            data = output_q.get()
            rec_points = data['rect_points']
            class_names = data['class_names']
            class_colors = data['class_colors']
            for point, name, color in zip(rec_points, class_names, class_colors):
                cv2.rectangle(frame, (int(point['xmin'] * args.width), int(point['ymin'] * args.height)),
                              (int(point['xmax'] * args.width), int(point['ymax'] * args.height)), color, 3)
                cv2.rectangle(frame, (int(point['xmin'] * args.width), int(point['ymin'] * args.height)),
                              (int(point['xmin'] * args.width) + len(name[0]) * 6,
                               int(point['ymin'] * args.height) - 10), color, -1, cv2.LINE_AA)
                cv2.putText(frame, name[0], (int(point['xmin'] * args.width), int(point['ymin'] * args.height)), font,
                            0.3, (0, 0, 0), 1)
            if args.stream_out:
                print('Streaming elsewhere!')
            else:
                cv2.imshow('Video', frame)

        fps.update()
```

```
                            0.3, (0, 0, 0), 1)
            if args.stream_out:
                print('Streaming elsewhere!')
            else:
                cv2.imshow('Video', frame)

        fps.update()

        print('[INFO] elapsed time: {:.2f}'.format(time.time() - t))
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

    fps.stop()
    print('[INFO] elapsed time (total): {:.2f}'.format(fps.elapsed()))
    print('[INFO] approx. FPS: {:.2f}'.format(fps.fps()))

    video_capture.stop()
    cv2.destroyAllWindows()
```

Ekran Çıktısı




Video
cell phone: 62%
object_detect
utils
__init__.py
nesne.py
External Libraries
Scratches and C
.time() - t))
(fps.elapsed()))
)))
Run:
nesne
4: Run
6: TODO
Terminal
Python Console
Event Log
154:1
CRLF
UTF-8
4 spaces
Python 3.7