
Windows için Tensorflow 1.15 ile Tensorflow Object Detection API Kurulumu

Bu kurulum bilgileri 29.08.2020 tarihinde hazırlanmıştır. Kurulumu daha kolay yapabilmek için şu linki seyrederek eş zamanlı olarak kurulum adımları takip edilirse hataları en aza indirerek kurulum tamamlanmış olur.

<https://www.youtube.com/watch?v=usR2LQuxhL4&t=1036s>

Ben karşılaştığım hataları bu dökümanda çözümleri ile birlikte derledim. Sizde bu kurulumu yaparken karşılaştığınız hataları internet üzerinden araştırarak bulabilirsiniz. Yinede destek için

rumeysaranayilmaz@gmail.com mail adresinden bana ulaşabilirsiniz yada stackoverflow üzerinden tensorflow tagı ile sorularınızı sorabilirsiniz. Kurulum yaparken karşılaştığınız hataları ve bulduğunuz çözümleri paylaşırsanız bu dökümana ekleyebiliriz.

Tensorflow 2.x sürümleri için kurulum yapmamamızın sebebi object_detection api ve protoc 3.4.0 ün desteklememesidir. Fakat dökümantasyonu incelemenizi ve konu ile alakalı güncel youtube videolarını takip etmenizi öneririm.

Önemli Uyarı:

Tensorflow Object Detection API'ı Tensorflow 2.x ile kullanmayacağız.

Kullanılacak Teknolojiler, Kütüphaneler ve Sürümleri:

- Tensorflow 1.15
- Python 3.7 den küçük bir 3.0'dan büyük bir sürüm
- Anaconda 4.8.2 yada üstü
- lxml 4.5.0, matplotlib 3.1.3, Pillow 7.0.0, jupyter 1.0.0, contextlib2 0.6.0.post1, Cython 0.29.15, tf_slim

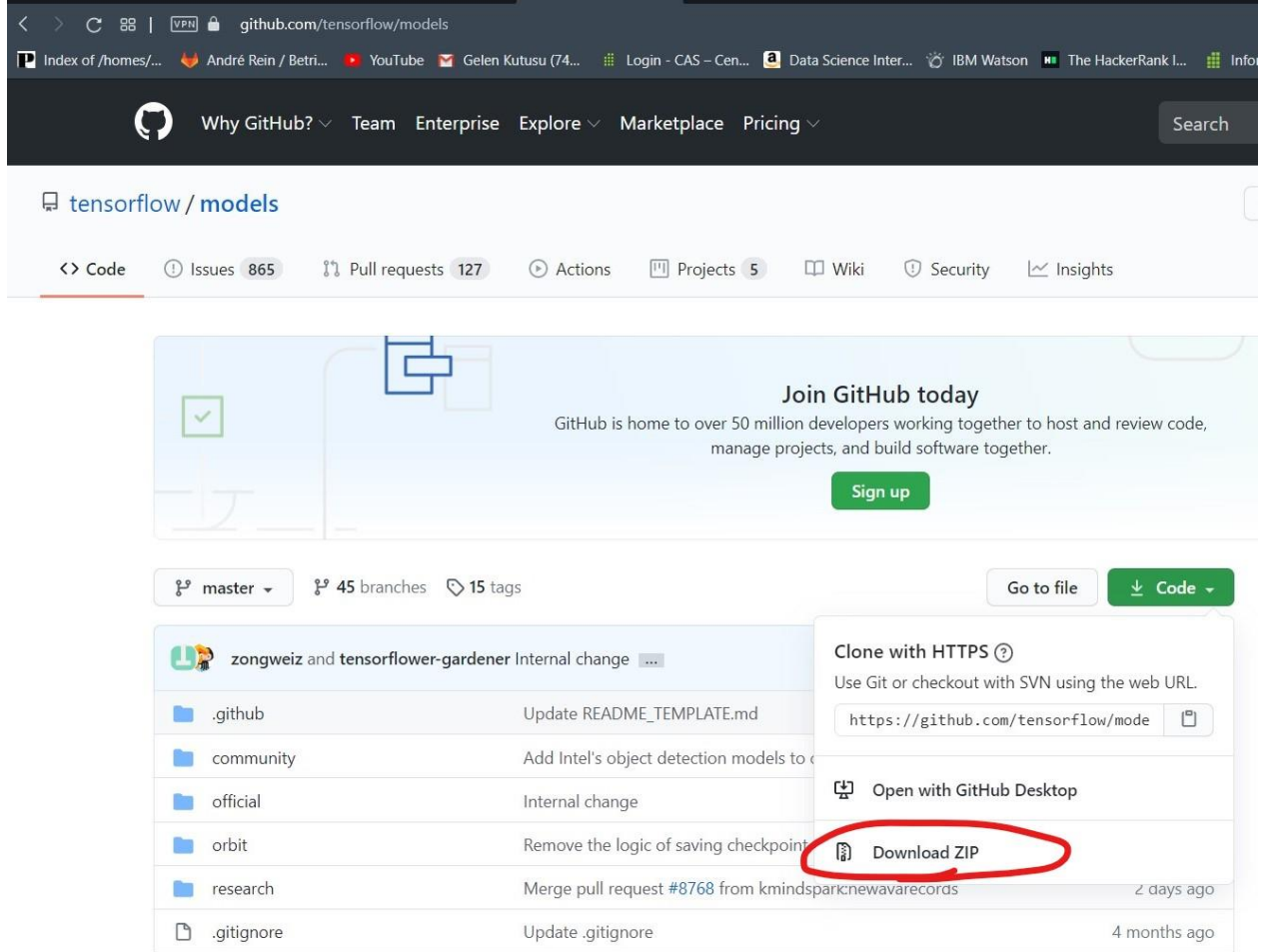
Kullanılabilecek Komutlar Listesi:

- `conda create --name sneakss python==3.15`
- `conda activate sneakss`
- `pip install <kütüphane_ismi>`
- `pip uninstall <kütüphane_ismi>`
- `pip install TensorFlow==1.15 lxml pillow matplotlib jupyter contextlib2 cython tf_slim`
- `git clone https://github.com/tensorflow/models.git`
- `protoc object_detection/protos/*.proto --python_out=.`
- `python setup.py build`
- `python setup.py install`
- `pip install pycocotools` yada
`pip install git+https://github.com/philferriere/cocoa...^&subdirectory=PythonAPI` yada
`pip install pycocotools-windows`

Kullanılabilecek Linkler

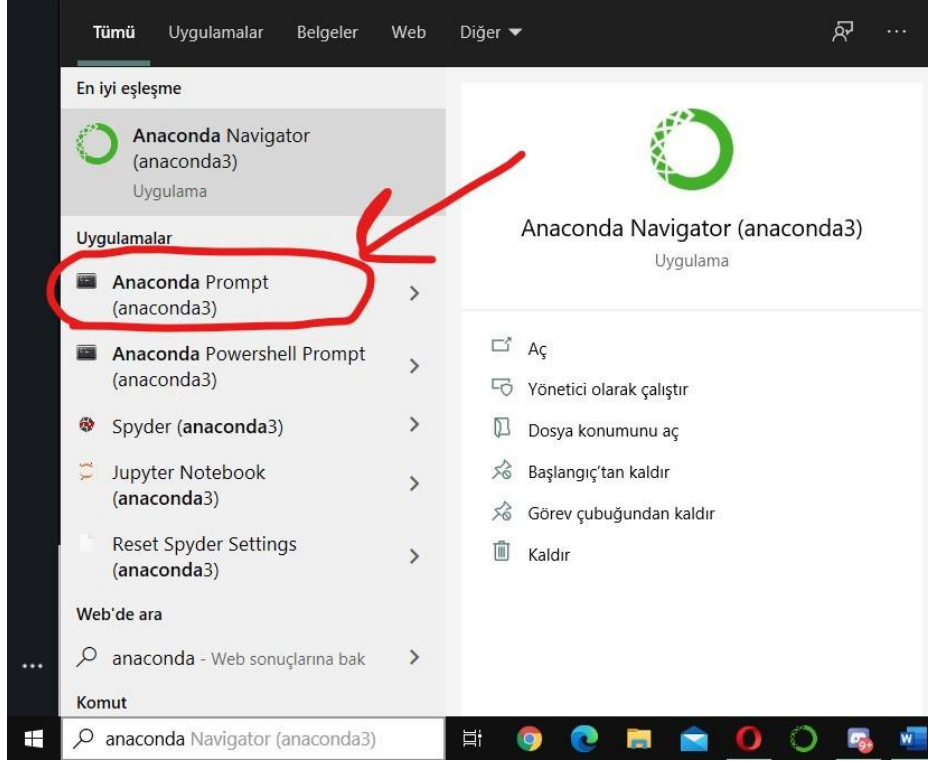
- Anaconda → <https://www.anaconda.com/products/individual#Downloads>
- Tensorflow/Models → <https://github.com/tensorflow/models>
- Protobuf → <https://github.com/protocolbuffers/protobuf/releases/tag/v3.4.0>

1. Yerel disk C'de models adında bir dosya oluřturunuz. Daha sonra řu linkten Models dökümanlarını Zip olarak yada git ile models klasörüne indiriniz.



2. Anacondayı indiriniz.

3. Anacondayı indirdikten sonra Anaconda Prompt (Anaconda Komut İstemcisini Açınız.)



4. Promptu açtıktan sonra -- `conda create --name sneakss python==3.5` komutunu girerek sneakss adında python 3.5 versiyonunu kullanan bir environment oluşturacağız. (sneakss dışında istediğiniz herhangi bir ismi de verebilirsiniz.)

Bu environmenti oluşturmamızın sebebi bazen Anacondanın base environmentinde pythonun 3.7'den daha yüksek bir sürümü çalıştığından şu hatayı veriyor olmasıdır.

ERROR: Could not find a version that satisfies the requirement tensorflow==1.15 (from versions: 2.2.0rc1, 2.2.0rc2, 2.2.0rc3)

ERROR: No matching distribution found for tensorflow==1.15

Bu hatayı alıyorsanız ve pythonun eski sürümlerinden biri ile environment **oluşturamıyorsanız Anacondayı silip tekrar yükleyiniz**

Bu komut ekranınız

```
Anaconda Prompt (anaconda3)

(base) C:\Users\suses>conda create --name sneakss python=3.5
```

sorunsuz çalıştıysa şöyle olacaktır.

The following NEW packages will be INSTALLED:

certifi	pkgs/main/win-64::certifi-2018.8.24-py35_1
pip	pkgs/main/win-64::pip-10.0.1-py35_0
python	pkgs/main/win-64::python-3.5.6-he025d50_0
setuptools	pkgs/main/win-64::setuptools-40.2.0-py35_0
vc	pkgs/main/win-64::vc-14.1-h0510ff6_4
vs2015_runtime	pkgs/main/win-64::vs2015_runtime-14.16.27012-hf0eaf9b_3
wheel	pkgs/main/noarch::wheel-0.35.1-py_0
wincertstore	pkgs/main/win-64::wincertstore-0.2-py35hfbbdb8_0

Proceed ([y]/n)? y

```
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#   $ conda activate sneakss
#
# To deactivate an active environment, use
#
#   $ conda deactivate
#
(base) C:\Users\suses>conda activate sneakss
(sneakss) C:\Users\suses>
```

Bu aşamadan sonra oluşturduğumuz environmenti kullanabilmek için şu komutu girmeliyiz.

-- `conda activate sneakss`

Bu komutu çalıştırdıktan sonra base environmentten sneakss environmente **geçiş yapıldığına emin olunuz.**

5. Oluşturduğumuz environmente gerekli kütüphaneleri kurabilmek için prompt'a şu kodu giriniz. İndirme internetinizin hızınıza göre 5-10 dakika sürebilir.

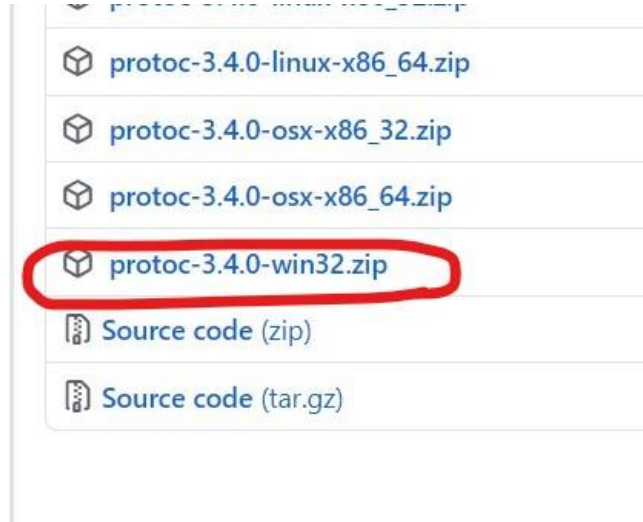
```
-- pip install TensorFlow==1.15 lxml pillow matplotlib jupyter contextlib2 cython tf_slim
```

```
(base) C:\Users\suses>conda activate sneakss  
(sneakss) C:\Users\suses>pip install TensorFlow==1.15 lxml pillow matplotlib jupyter contextlib2 cython tf_slim
```

TÜM İŞLEMLER BİTENE KADAR PROMPT'U KAPATMAYINIZ!

6. Yerel Disk C de **Protoc** diye bir klasör oluşturun. Şu linkten <https://github.com/protocolbuffers/protobuf/releases/tag/v3.4.0> protobuf 3.4.0 sürümünü Windows için **Protoc** klasörüne indirip Zip'ten çıkarınız. Daha Sonra bin klasörünü açıp içinden **protoc.exe** dosyasını daha önce açtığımız models klasörünün içinde **research** klasörüne atalım.

– protoc.exe → research dosyasının içine atılmalı.



7. Pycocotools kütüphanesini kurmak için prompt üzerinde şu komutları sıra ile deneyerek çalışanı bulana kadar indirmeyi deneyiniz.

```
pip install pycocotools yada
```


```
pip install git+https://github.com/philferriere/cocoa...^&subdirectory=PythonAPI yada
```

```
pip install pycocotools-windows
```

```
(sneakss) c:\>pip install pycocotools
```

8. Prompt ekranından `cd` komutu ile models'in altındaki research klasörüne locate olun.
- `cd C:\models\research`

```
(sneakss) C:\Users>cd C:\models\research
(sneakss) C:\models\research>
```



9. Alttaki komutları research klasörüne locate olduktan sonra sırası ile çalıştırınız.
- `python setup.py build`
 - `python setup.py install`
10. Daha sonra research klasörünün altındaki object_detecton klasörüne locate olun.
- `cd object_detection`
11. Daha sonra sonra `jupyter notebook` yazarak notebookumuzu tarayıcımız üzerinde çalıştırmak için komut verelim.

```
(sneakss) c:\models\research\object_detection>jupyter notebook
```

12. `colab_tutorials` dosyasından `object_detection_tutorial.ipynb` notebookunu açalım.

The screenshot shows the JupyterLab interface in a web browser. The address bar indicates the URL is `localhost:8888/tree`. The Jupyter logo is visible at the top. Below the logo, there are tabs for 'Files', 'Running', and 'Clusters'. The 'Files' tab is active, showing a file explorer view. The file explorer displays a list of folders and files. The folder `colab_tutorials` is highlighted with a red box. Below the file explorer, there is a list of notebooks. The notebook `object_detection_tutorial.ipynb` is highlighted with a red box.

Files Running Clusters

Select items to perform actions on them.

☐ 0 /

- ☐ `anchor_generators`
- ☐ `box_coders`
- ☐ `builders`
- ☐ `colab_tutorials`
- ☐ `configs`
- ☐ `core`
- ☐ `data`

- ☐ `context_rcnn_tutorial.ipynb`
- ☐ `eager_few_shot_od_training_tf2_colab.ipynb`
- ☐ `inference_from_saved_model_tf2_colab.ipynb`
- ☐ `inference_tf2_colab.ipynb`
- ☐ `object_detection_tutorial.ipynb`

13. Loader Kısmından model değişkenini
`model = tf.compat.v2.saved_model.load(str(model_dir))`
olarak değiştiriniz.

Loader

```
In [9]: def load_model(model_name):  
        base_url = 'http://download.tensorflow.org/models/object_detection/'  
        model_file = model_name + '.tar.gz'  
        model_dir = tf.keras.utils.get_file(  
            fname=model_name,  
            origin=base_url + model_file,  
            untar=True)  
  
        model_dir = pathlib.Path(model_dir)/"saved_model"  
  
        model = tf.compat.v2.saved_model.load(str(model_dir))  
  
        return model
```



Eğer bu değişimi yapmazsanız şu hatayı alabilirsiniz.

`TypeError: load() missing 2 required positional arguments: 'tags' and 'export_dir'`

Ayrıntılı bilgi için şu linki ziyaret edebilirsiniz.

- <https://github.com/tensorflow/models/issues/7773>

14. Kütüphane importlarının altına şu kodu ekleyiniz.

- `tf.compat.v1.enable_eager_execution()`

Import the object detection module.

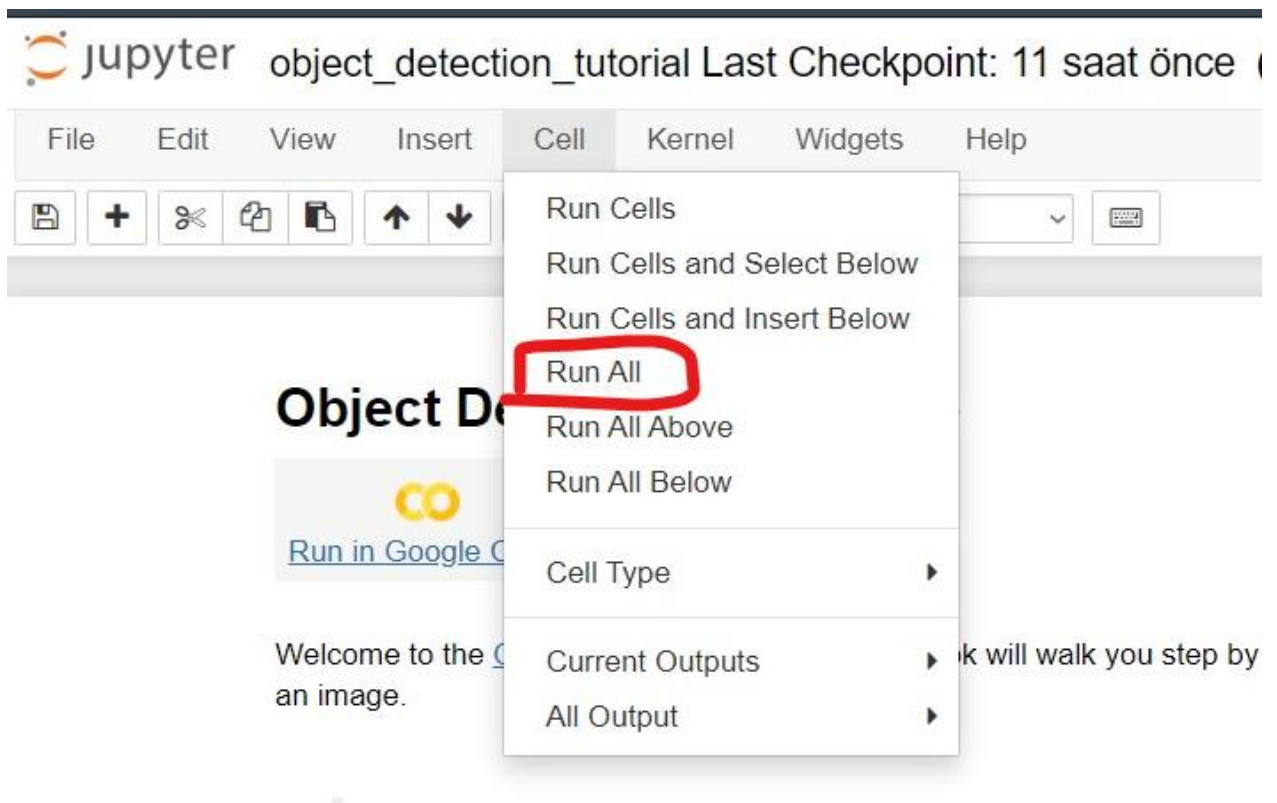
```
In [7]: from object_detection.utils import ops as utils_ops
        from object_detection.utils import label_map_util
        from object_detection.utils import visualization_utils as vis_util

        tf.compat.v1.enable_eager_execution()
```

Bunu yapmazsanız şu hatayı alabilirsiniz.

`int() argument must be a string, a bytes-like object or a number, not 'Tensor'`

15. Notebook'un en üst kısmından Cell'i açın Run All seçeneğini seçip tüm kodların çalışmasını sağlayın.



16. Tüm kodların çalıştırılması biraz uzun sürebilir.