Изначально качаем вску, ставим расширения:

1. Russian Language Pack for Visual Studio Code  
2. Python Extension Pack  
3. Python Environment Manager  
4. Python Debugger

5. Python

6. Code Runner

**Потом ide 3.9:**

[Python Release Python 3.9.0 | Python.org](https://www.python.org/downloads/release/python-390/)

Создаем **venv** в нашем проекте, и погнали по гайду.

[TensorFlow 2 Object Detection API tutorial — TensorFlow 2 Object Detection API tutorial documentation (tensorflow-object-detection-api-tutorial.readthedocs.io)](https://tensorflow-object-detection-api-tutorial.readthedocs.io/en/latest/) на всякий.

Качаем **tenserflow**  
pip install --ignore-installed --upgrade tensorflow==2.5.0  
  
Проверяем установку  
python -c "import tensorflow as tf;print(tf.reduce\_sum(tf.random.normal([1000, 1000])))"

На выходе получим +- такое   
Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

По хорошему надо:

Обновите драйверы графического процессора (необязательно)

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание Создаем папку **Tensorflow**

### Отесда качаем **api** для обнаружения объектов <https://github.com/tensorflow/models/archive/master.zip> там архив будэ распаковали закинули в **tensorflow**

### Установка/компиляция **Protobuf**

Качаем зип на нашу винду [Releases · protocolbuffers/protobuf (github.com)](https://github.com/protocolbuffers/protobuf/releases)  
любой версии, лишь бы на питухон был.

Потом нужно созать папку **C:\Program Files\Google Protobuf гугл протобаф**, в ней создать еще одну **protoc** распаковать все туда.

Дальше заходим в изменение переменных среды текущего пользвоателя добавляем в патч

Ссылку на эту папку, путь крч.

Изображение выглядит как текст, программное обеспечение, снимок экрана, Значок на компьютере

Автоматически созданное описание

Вводим команду

Cd TensorFlow/models/research/

Потом в терминале

protoc object\_detection/protos/\*.proto --python\_out=.

Ниче не произойдет это норма.

**COCO API** скипаем

### Но если нас лед этапе крашниться то качаем его сначало обновляем **Visual Studio ++** крч там по гайду Установка API обнаружения объектов

Cd TensorFlow/models/research

cp object\_detection/packages/tf2/setup.py .

python -m pip install .

Тестируем установку

Cd Tensorflow\models\research  
python object\_detection/builders/model\_builder\_tf2\_test.py

Тут выходит ошибка.

Пофиксить я не смог поэтому делаем так. Я на гит кину .venv и models Делаем все так же, обновляем дрова создаем папку **tenserflow** качаем **protobuf** и в **patch** его кидаем. После импортируем с гита венв и модели. Тестим командой.

python object\_detection/builders/model\_builder\_tf2\_test.py

и балдеем.

Теперь создаем две папки

Изображение выглядит как текст, снимок экрана, Мультимедийное программное обеспечение, программное обеспечение

Автоматически созданное описание

TensorFlow/

├─ addons/ (Optional)

│ └─ labelImg/

├─ models/

│ ├─ community/

│ ├─ official/

│ ├─ orbit/

│ ├─ research/

│ └─ ...

└─ workspace/

└─ training\_demo/

training\_demo/

├─ annotations/

├─ exported-models/

├─ images/

│ ├─ test/

│ └─ train/

├─ models/

├─ pre-trained-models/

└─ README.md

Теперь качаем   
**pip install labelImg**

Запускаем его командой   
**labelImg**

Создаем папку **Scripts** Кидаем туда тему котурую написали с юрой для фотографий на cv2 логично cv2 нужно импортнуть  
**pip install opencv-python**  
запускаем и делаем фотографии для нашего датасета. Он создает папку **images.**

Размечаем все фото.  
далее делим фотки на те которые будут в тесте для проверки и на те которые будут тренить. В тест 1/10 от всех фото обычно.

В папке scripts создаем **preprocessing** папку  
Создаем в **annotations** карту меток label\_map.pbtxt вот название ее   
item {

id: 1

name: 'cat'

}

item {

id: 2

name: 'dog'

}

внутри что-то такое

качаем скрипт **generate\_tfrecord.py**

кидаем его в **scripts/preprocessing**  
качаем **pandas   
pip install pandas**

python generate\_tfrecord.py -x [PATH\_TO\_IMAGES\_FOLDER]/train -l [PATH\_TO\_ANNOTATIONS\_FOLDER]/label\_map.pbtxt -o [PATH\_TO\_ANNOTATIONS\_FOLDER]/train.record

*# Create test data:*

python generate\_tfrecord.py -x [PATH\_TO\_IMAGES\_FOLDER]/test -l [PATH\_TO\_ANNOTATIONS\_FOLDER]/label\_map.pbtxt -o [PATH\_TO\_ANNOTATIONS\_FOLDER]/test.record

Махаем наши файлы в .record  
  
Качаем файл обучения  
<https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md> отседава [SSD ResNet50 V1 FPN 640x640](http://download.tensorflow.org/models/object_detection/tf2/20200711/ssd_resnet50_v1_fpn_640x640_coco17_tpu-8.tar.gz)

Распаковывем его и кидаем в **pre-traid-model**Махаем наш **cfg**, на скаченный.

Меняем шаги и тд  
python model\_main\_tf2.py --model\_dir=models/my\_ssd\_resnet50\_v1\_fpn --pipeline\_config\_path=models/my\_ssd\_resnet50\_v1\_fpn/pipeline.config

запускаем обучение и ждем, епта

После этого копируем в скрипты **exporter\_main\_v2**

python model\_main\_tf2.py --model\_dir=models/my\_ssd\_resnet50\_v1\_fpn --pipeline\_config\_path=models/my\_ssd\_resnet50\_v1\_fpn/pipeline.config --checkpoint\_dir=models/my\_ssd\_resnet50\_v1\_fpn

Запускаем команду появиться папочка

**Exporter\_models** в ней будет наши чекпоинты и вся нейронка.

**Качаем Detection** Указываем пути, запускаем и наслаждаемся.

git remote add origin https://github.com/nike2005231/Tenserflow\_ws.git