

Instalación

martes, 29 de enero de 2019 15:56

Última Fecha Actualización: 31/01/2019

Objetivo: Pasos para instalación de **TF Object Detection API** en PC con *Windows 10 + Anaconda*.

Fuentes:

- Real Time Object Detection with TensorFlow Detection Model: <https://towardsdatascience.com/real-time-object-detection-with-tensorflow-detection-model-e7fd20421d5d>
- TensorFlow Object Detection API: https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/installation.md
- TensorFlow Object Detection API tutorial: <https://tensorflow-object-detection-api-tutorial.readthedocs.io/en/latest/>
- <https://github.com/tensorflow/models/issues/1591#issuecomment-384633225>

Pasos de instalación:

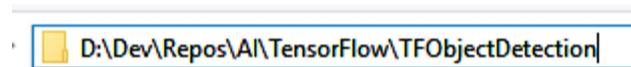
1. Instalar las siguientes librerías en el environment de Anaconda correspondiente:
 - pillow.
 - lxml.
 - matplotlib.
 - tensorflow.
 - cython.
 - contextlib2.

Nota: Las librerías anteriores se instalan todas por medio del comando: **pip install <librería>**.

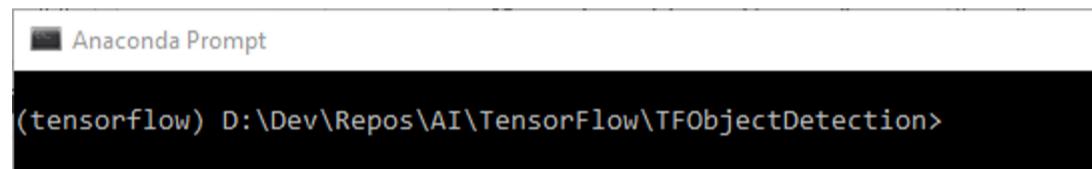
2. Instalar Protobuf: En el caso de este manual se descarga la versión "protoc-3.6.1-win32.zip" de la página GitHub del proyecto <https://github.com/protocolbuffers/protobuf/releases>.

Nota: El archivo .zip descargado se usará más adelante.

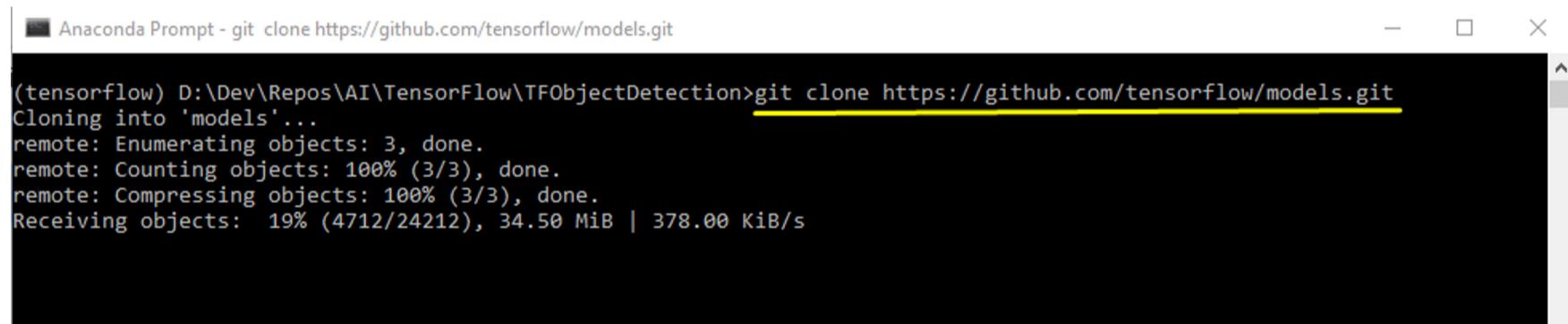
3. Ahora es momento de instalar los "*TensorFlow Models*" que se encuentran en <https://github.com/tensorflow/models>:
 - a. Crear carpeta donde se manejará la instalación de **TF Object Detection API**. Para el caso de este manual se crea la siguiente ruta:



- b. Una vez se haya creado la estructura de directorios anterior, en **Anaconda Prompt** ir a esa ruta:



- c. Clonar proyecto de *models* en el directorio local como se muestra a continuación:

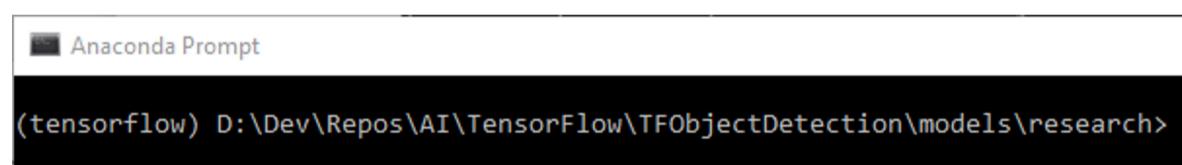


- d. Una vez terminada la descarga del proyecto *models*, verificar que en la ruta "*models\research\object_detection\protos*" **NO** se encuentren ninguno de los siguientes archivos:

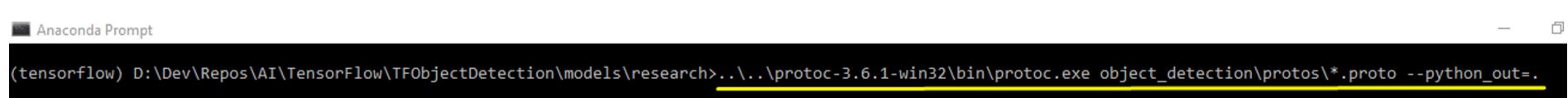
- *box.predictor_pb2.py*.
- *faster_rcnn_box_coder_pb2.py*.

Nota: Para obtener los archivos anteriores se deben compilar las librerías Protobuf, lo cual se realizará en el siguiente punto.

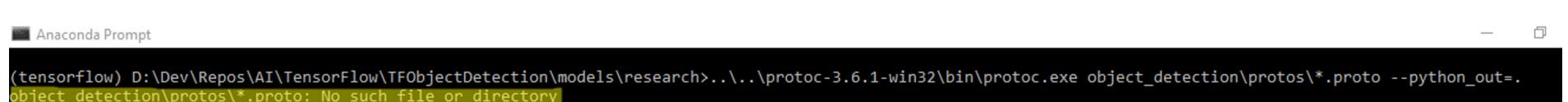
4. Compilar librerías Protobuf:
 - a. Descomprimir el archivo "protoc-3.6.1-win32.zip" en la carpeta "/TFOBJECTDETECTION".
 - b. Navegar en la consola a la ruta "TFOBJECTDETECTION\models\research":



- c. Compilar con el ejecutable que se encuentra en la ruta "..\..\protoc-3.6.1-win32\bin\protoc.exe" los archivos ".proto" que se encuentran en la ruta "*object_detection\protos*":
 - Con base en la documentación del API de TensorFlow se debe ejecutar el siguiente comando:



- Pero al ejecutarlo en Windows sale el siguiente error:



- Despues de hacer una investigación, al parecer esta parte del comando "protos*.proto" **NO** es soportada en Windows

(<https://github.com/tensorflow/models/issues/1591#issuecomment-384633225>). Por lo cual, se tienen las siguientes opciones:

- Ejecutar manualmente cada uno de los archivos ".proto" que se encuentren en el directorio como se muestra a continuación:

```
Anaconda Prompt
(tensorflow) D:\Dev\Repos\AI\TensorFlow\TFObjectDetection\models\research>..\..\protoc-3.6.1-win32\bin\protoc.exe object_detection\protos\anchor_generator.proto --python_out=.
```

Nota: Lo anterior se debe ejecutar para cada uno de los siguientes archivos:

Name	Date modified	Type	Size
anchor_generator.proto	29/01/2019 18:06	PROTO File	1 KB
argmax_matcher.proto	29/01/2019 18:06	PROTO File	2 KB
bipartite_matcher.proto	29/01/2019 18:06	PROTO File	1 KB
box_coder.proto	29/01/2019 18:06	PROTO File	1 KB
box_predictor.proto	29/01/2019 18:06	PROTO File	8 KB
eval.proto	29/01/2019 18:06	PROTO File	4 KB
faster_rcnn.proto	29/01/2019 18:06	PROTO File	9 KB
faster_rcnn_box_coder.proto	29/01/2019 18:06	PROTO File	1 KB
graph_rewriter.proto	29/01/2019 18:06	PROTO File	1 KB
grid_anchor_generator.proto	29/01/2019 18:06	PROTO File	2 KB
hyperparams.proto	29/01/2019 18:06	PROTO File	5 KB
image_resizer.proto	29/01/2019 18:06	PROTO File	3 KB
input_reader.proto	29/01/2019 18:06	PROTO File	6 KB
keypoint_box_coder.proto	29/01/2019 18:06	PROTO File	1 KB
losses.proto	29/01/2019 18:06	PROTO File	8 KB
matcher.proto	29/01/2019 18:06	PROTO File	1 KB
mean_stddev_box_coder.proto	29/01/2019 18:06	PROTO File	1 KB
model.proto	29/01/2019 18:06	PROTO File	1 KB
multiscale_anchor_generator.proto	29/01/2019 18:06	PROTO File	1 KB
optimizer.proto	29/01/2019 18:06	PROTO File	4 KB
pipeline.proto	29/01/2019 18:06	PROTO File	1 KB
post_processing.proto	29/01/2019 18:06	PROTO File	2 KB
preprocessor.proto	29/01/2019 18:06	PROTO File	17 KB
region_similarity_calculator.proto	29/01/2019 18:06	PROTO File	1 KB
square_box_coder.proto	29/01/2019 18:06	PROTO File	1 KB
ssd.proto	29/01/2019 18:06	PROTO File	9 KB
ssd_anchor_generator.proto	29/01/2019 18:06	PROTO File	3 KB
string_int_label_map.proto	29/01/2019 18:06	PROTO File	1 KB
train.proto	29/01/2019 18:06	PROTO File	6 KB

- Crear un archivo ".bat":

- Crear el archivo ".bat" con el siguiente código:

```
run_protoc.bat - Notepad
File Edit Format View Help
@echo off
setlocal
echo Searching for new .proto files...

for %%F in (object_detection\protos\*.proto) do (
    echo %%F
    ..\..\protoc-3.6.1-win32\bin\protoc.exe %%F --python_out=.
)
echo Complete!
```

- Guardar el archivo anterior en la ruta "TFObjectDetection\models\research". El nombre con el que se guardó el archivo en este manual es "run_protoc.bat".
 - Ejecutar el archivo ".bat" desde la consola de la siguiente manera:

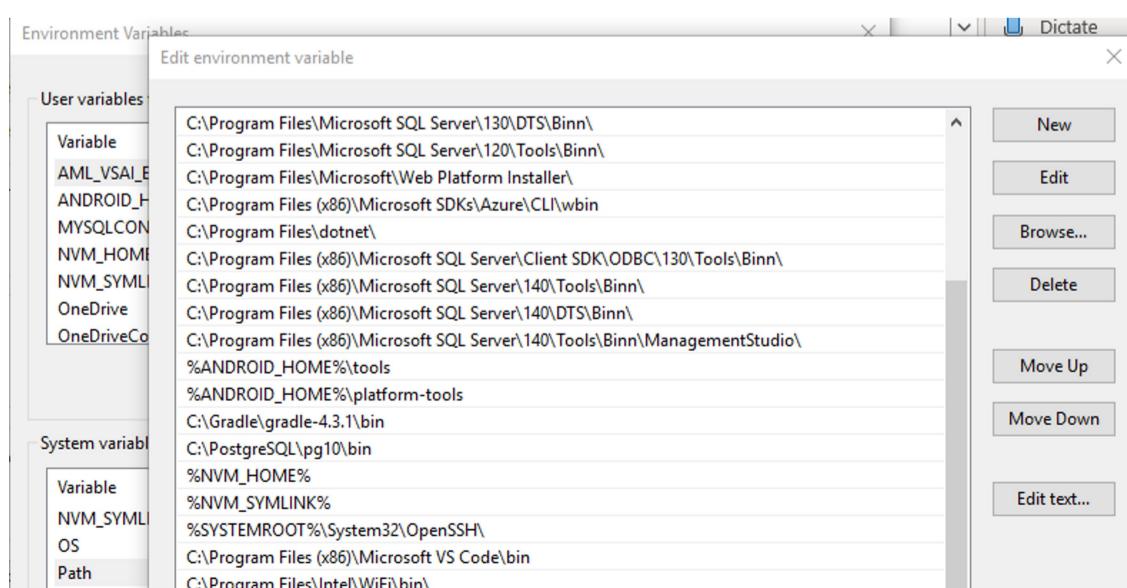
```
(tensorflow) D:\Dev\Repos\AI\TensorFlow\TFObjectDetection\models\research>run_protoc.bat
Searching for new .proto files...
object_detection\protos\anchor_generator.proto
object_detection\protos\argmax_matcher.proto
object_detection\protos\bipartite_matcher.proto
object_detection\protos\box_coder.proto
object_detection\protos\box_predictor.proto
object_detection\protos\eval.proto
object_detection\protos\faster_rcnn.proto
object_detection\protos\faster_rcnn_box_coder.proto
object_detection\protos\graph_rewriter.proto
object_detection\protos\grid_anchor_generator.proto
object_detection\protos\hyperparams.proto
object_detection\protos\image_resizer.proto
object_detection\protos\input_reader.proto
object_detection\protos\keypoint_box_coder.proto
object_detection\protos\losses.proto
object_detection\protos\matcher.proto
object_detection\protos\mean_stddev_box_coder.proto
object_detection\protos\model.proto
object_detection\protos\multiscale_anchor_generator.proto
object_detection\protos\optimizer.proto
object_detection\protos\pipeline.proto
object_detection\protos\post_processing.proto
object_detection\protos\preprocessor.proto
object_detection\protos\region_similarity_calculator.proto
object_detection\protos\square_box_coder.proto
object_detection\protos\ssd.proto
object_detection\protos\ssd_anchor_generator.proto
object_detection\protos\string_int_label_map.proto
object_detection\protos\train.proto
Complete!
```

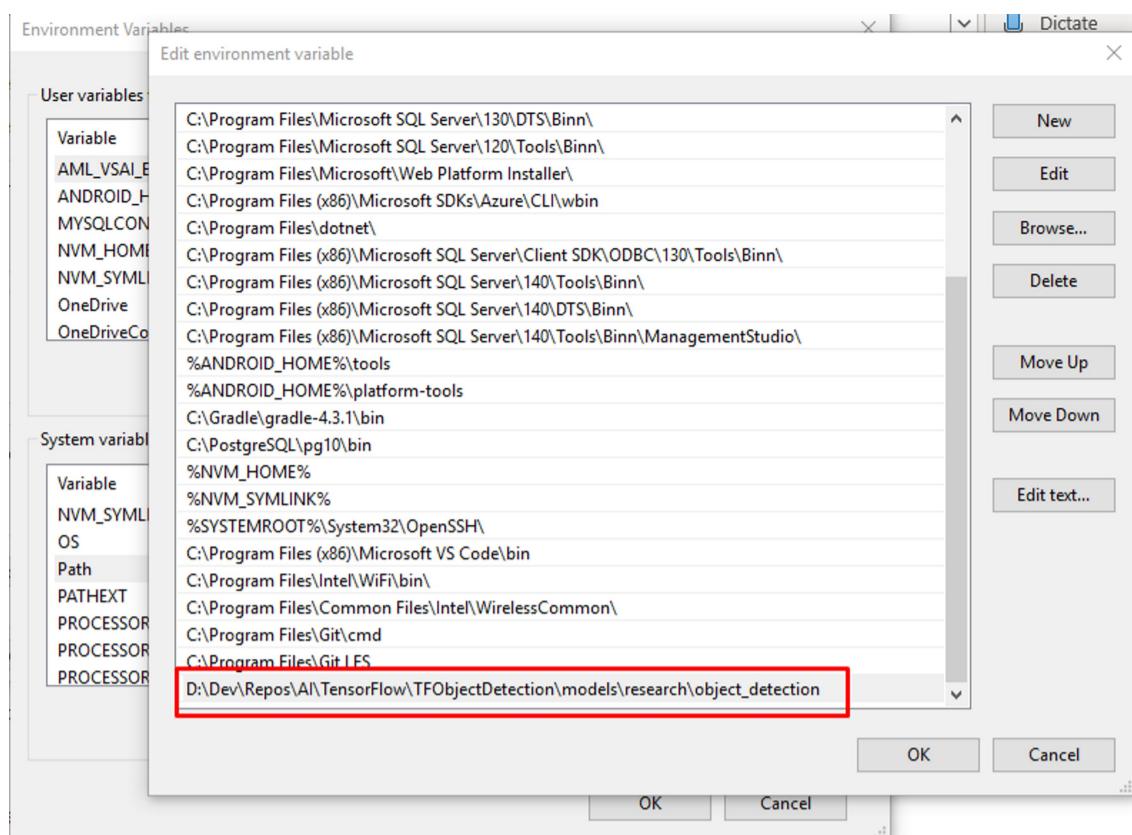
- De esta manera ya quedaron compilados los archivos ".proto" y debe existir por cada archivo proto un archivo ".py" con el mismo nombre:

Name	Date modified	Type	Size
__pycache__	30/01/2019 08:57	File folder	
__init__.py	29/01/2019 18:06	PY File	0 KB
anchor_generator.proto	29/01/2019 18:06	PROTO File	1 KB
anchor_generator_pb2.py	30/01/2019 10:37	PY File	6 KB
argmax_matcher.proto	29/01/2019 18:06	PROTO File	2 KB
argmax_matcher_pb2.py	30/01/2019 10:37	PY File	5 KB
bipartite_matcher.proto	29/01/2019 18:06	PROTO File	1 KB
bipartite_matcher_pb2.py	30/01/2019 10:37	PY File	3 KB
box_coder.proto	29/01/2019 18:06	PROTO File	1 KB
box_coder_pb2.py	30/01/2019 10:37	PY File	7 KB
box_predictor.proto	29/01/2019 18:06	PROTO File	8 KB
box_predictor_pb2.py	30/01/2019 10:37	PY File	34 KB
eval.proto	29/01/2019 18:06	PROTO File	4 KB
eval_pb2.py	30/01/2019 10:37	PY File	12 KB
faster_rcnn.proto	29/01/2019 18:06	PROTO File	9 KB
faster_rcnn_box_coder.proto	29/01/2019 18:06	PROTO File	1 KB
faster_rcnn_box_coder_pb2.py	30/01/2019 10:37	PY File	4 KB
faster_rcnn_pb2.py	30/01/2019 10:37	PY File	24 KB
graph_rewriter.proto	29/01/2019 18:06	PROTO File	1 KB
graph_rewriter_pb2.py	30/01/2019 10:37	PY File	5 KB
grid_anchor_generator.proto	29/01/2019 18:06	PROTO File	2 KB
grid_anchor_generator_pb2.py	30/01/2019 10:37	PY File	6 KB
hyperparams.proto	29/01/2019 18:06	PROTO File	5 KB
hyperparams_pb2.py	30/01/2019 10:37	PY File	27 KB
image_resizer.proto	29/01/2019 18:06	PROTO File	3 KB
image_resizer_pb2.py	30/01/2019 10:37	PY File	12 KB
input_reader.proto	29/01/2019 18:06	PROTO File	6 KB
input_reader_pb2.py	30/01/2019 10:37	PY File	17 KB
keypoint_box_coder.proto	29/01/2019 18:06	PROTO File	1 KB

5. Adicionar variables de entorno necesarias:

- Adicionar "<PATH_TO_TF>\models\research\object_detection" en la variable de sistema **Path**:

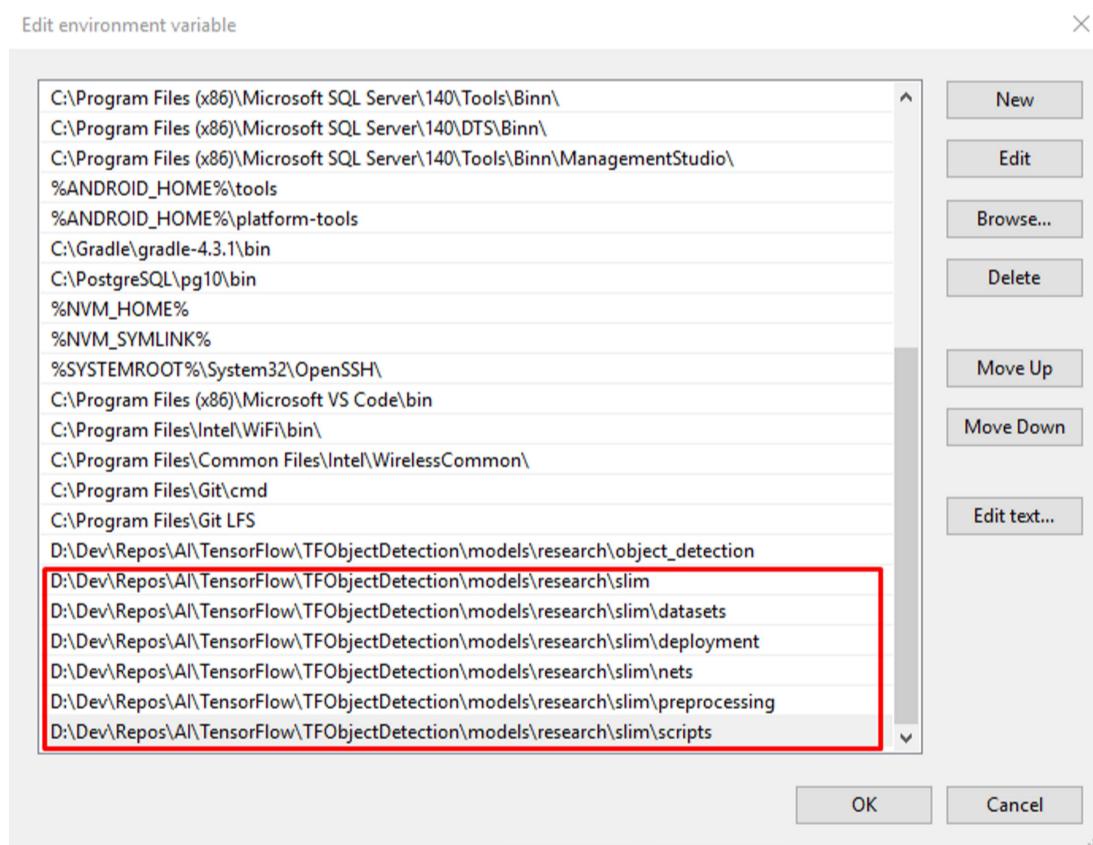




- b. Algunos de los paquetes que se requieren para realizar *object detection* NO vienen pre-instalados con el paquete de TensorFlow. Por esto debemos adicionar las siguientes rutas a la variable de entorno **Path**, tal como se realizó en el paso anterior:

- <PATH_TO_TF>\models\research\slim
- <PATH_TO_TF>\models\research\slim\datasets
- <PATH_TO_TF>\models\research\slim\deployment
- <PATH_TO_TF>\models\research\slim\nets
- <PATH_TO_TF>\models\research\slim\preprocessing
- <PATH_TO_TF>\models\research\slim\scripts

En nuestro caso, <PATH_TO_TF> lo vamos a reemplazar por "D:\Dev\Repos\AI\TensorFlow\TFObjectDetection\".



Nota: Se debe abrir una nueva instancia de **Anaconda Prompt** para que los cambios en las variables de entorno tengan efecto.

6. En este momento ya se encuentran instalados los diferentes requerimientos para empezar a probar **TF Object Detection API**.
 Ir a la sección **PoCs** donde se muestra cómo empezar a probar este API.