HW5 Мельчук А.Б. Обучить детектор объектов с помощью TensorFlow Object Detection API Библиотеки: [Python, Tensorflow]

In [1]:

```
# Select TF1 in colab
%tensorflow_version 1.x
import tensorflow as tf
```

In [2]:

```
!git clone https://github.com/Dju999/TFFashionDetection.git
!pip install lxml
!pip install -U -q PyDrive
!pip install tqdm
```

```
Cloning into 'TFFashionDetection'...
remote: Counting objects: 59, done.
remote: Compressing objects: 100% (20/20), done.
remote: Total 59 (delta 21), reused 21 (delta 11), pack-reused 28
Unpacking objects: 100% (59/59), done.
Requirement already satisfied: lxml in /usr/local/lib/python3.6/dist-p
ackages (4.2.1)
Requirement already satisfied: tqdm in /usr/local/lib/python3.6/dist-p
ackages (4.23.3)
```

Монтируем датасет [DeepFashion (http://mmlab.ie.cuhk.edu.hk/projects/DeepFashion.html)](http://mmlab.ie.cuhk.edu.hk/projects/DeepFashion.html) из google drive и копируем

In [3]:

```
from google.colab import drive
drive.mount('/content/drive')
```

In [4]:

```
from TFFashionDetection.utils.colab_fs import GoogleColabFS

fs = GoogleColabFS()
```

In [5]:

```
!rm -rv ./fashion_data
!cp -rv "drive/My Drive/Category and Attribute Prediction Benchmark/" ./TEMP
!mv ./TEMP ./fashion_data
```

In [6]:

```
!unzip ./fashion_data/Img/img.zip -d ./fashion_data/Img
```

Ставим Tensorflow Object Detection API

In [7]:

```
!pip install tf_slim
```

In [8]:

```
# загружаем репозиторий, добавляем в PYTHONPATH
! cd /content; git clone https://github.com/tensorflow/models.git
# установка зависимостей для object detection тут
# https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/i
!apt-get install protobuf-compiler python-pil python-lxml python-tk
!pip install Cython
!cd /content; git clone https://github.com/cocodataset/cocoapi.git; cd cocoapi/Pytho
!cd /content/models/research; protoc object_detection/protos/*.proto --python_out=.
# проверка
!cd /content/models/research; export PYTHONPATH=$PYTHONPATH:`pwd`:`pwd`/slim; python
```

Формируем структуру

In [9]:

```
import sys
import os

import numpy as np

API_PATH = os.path.join('/content', 'models/research')
sys.path.append(API_PATH)

DETECTOR_PATH = os.path.join('/content', 'TFFashionDetection')
sys.path.append(DETECTOR_PATH)

from TFFashionDetection.data_preparator import DataPreparator
from TFFashionDetection.utils.ssd_config import write_config

data_preparator = DataPreparator()
data_preparator.build()
write_config('ssd_mobilenet_v2_coco_2018_03_29')
```

In [10]:

```
# качаем ппредоученную модель
!python /content/TFFashionDetection/utils/download_tf_zoo_model.py --name ssd_mobile
```

Обучаем модель

In [11]:

```
!export PYTHONPATH=$PYTHONPATH:/content/models/research/slim:/content/models/researc
```

```
/usr/local/lib/python3.6/dist-packages/h5py/__init__.py:36: FutureWar
ning: Conversion of the second argument of issubdtype from `float` to
`np.floating` is deprecated. In future, it will be treated as `np.flo
at64 == np.dtype(float).type`.
  from ._conv import register_converters as _register_converters
WARNING:tensorflow:From /content/models/research/object_detection/tra
iner.py:257: create_global_step (from tensorflow.contrib.framework.py
thon.ops.variables) is deprecated and will be removed in a future ver
sion.
Instructions for updating:
Please switch to tf.train.create_global_step
INFO:tensorflow:depth of additional conv before box predictor: 0
INFO:tensorflow:depth of additional conv before box predictor: 0
INFO:tensorflow:depth of additional conv before box predictor: 0
INFO:tensorflow:depth of additional conv before box predictor: 0
INFO:tensorflow:depth of additional conv before box predictor: 0
INFO:tensorflow:depth of additional conv before box predictor: 0
WARNING:root:Variable [FeatureExtractor/MobilenetV2/Conv/BatchNorm/be
ta/ExponentialMovingAverage] is not available in checkpoint
```

Сохраняем чекпоинт на Гугл-диск

In [12]:

```
!cd /content/data_dir; zip -r checkpoint_save_20180514.zip checkpoints/*
import os
fs = GoogleColabFS()

file_name = os.path.join('/content/data_dir', 'checkpoint_save_20180514.zip')

fs.load_to_drive(file_name)
```

```
  adding: checkpoints/checkpoint (deflated 78%)
  adding: checkpoints/events.out.tfevents.1526234121.2fe46401caed (def
lated 87%)
  adding: checkpoints/graph.pbtxt (deflated 97%)
  adding: checkpoints/model.ckpt-1969.data-00000-of-00001 (deflated
7%)
  adding: checkpoints/model.ckpt-1969.index (deflated 72%)
  adding: checkpoints/model.ckpt-1969.meta (deflated 94%)
  adding: checkpoints/model.ckpt-2004.data-00000-of-00001 (deflated
7%)
  adding: checkpoints/model.ckpt-2004.index (deflated 72%)
  adding: checkpoints/model.ckpt-2004.meta (deflated 94%)
  adding: checkpoints/model.ckpt-2038.data-00000-of-00001 (deflated
7%)
  adding: checkpoints/model.ckpt-2038.index (deflated 72%)
  adding: checkpoints/model.ckpt-2038.meta (deflated 94%)
  adding: checkpoints/model.ckpt-2073.data-00000-of-00001 (deflated
7%)
  adding: checkpoints/model.ckpt-2073.index (deflated 72%)
  adding: checkpoints/model.ckpt-2073.meta (deflated 94%)
  adding: checkpoints/model.ckpt-2108.data-00000-of-00001 (deflated
7%)
  adding: checkpoints/model.ckpt-2108.index (deflated 72%)
  adding: checkpoints/model.ckpt-2108.meta (deflated 94%)
  adding: checkpoints/pipeline.config (deflated 75%)
```

И экспортируем чекпоинт

In [13]:

```
import logging
import os

logger = logging.getLogger(__name__)

fs.load_file_from_drive('/content', 'checkpoint_save_20180514.zip')
fs.unzip_file('/content', 'checkpoint_save_20180514.zip')

!mkdir /content/deep_detection_model

# экспортируем модель
!export PYTHONPATH=$PYTHONPATH:/content/models/research/slim:/content/models/researc
```

```
/usr/local/lib/python3.6/dist-packages/h5py/__init__.py:36: FutureWarn
ing: Conversion of the second argument of issubdtype from `float` to `
np.floating` is deprecated. In future, it will be treated as `np.float
64 == np.dtype(float).type`.
  from ._conv import register_converters as _register_converters
WARNING:tensorflow:From /content/models/research/object_detection/expo
rter.py:358: get_or_create_global_step (from tensorflow.contrib.framew
ork.python.ops.variables) is deprecated and will be removed in a futur
e version.
Instructions for updating:
Please switch to tf.train.get_or_create_global_step
Converted 404 variables to const ops.
```

In [14]:

```
from IPython.display import Image
import os

file_path = os.path.join('/content/data_dir/images', os.listdir('/content/data_dir/i

Image(file_path)
```

Out[15]:

Видим результаты детекции — lower_body предмет одежды

In [16]:

```
from TFFashionDetection.tf_object_detector import ObjectDetector
import numpy as np

oject_detector = ObjectDetector('/content/inference_graph/frozen_inference_graph.pb'

oject_detector.object_detection(image_dir='/content/data_dir/images', filename=file_
[[i['category_box'],i['img_name'],i['category_id'], i['category_proba']] for i in oj
#oject_detector.img_detections
np.unique([i['category_name'] for i in oject_detector.img_detections])
```

Out[17]:

```
array(['full-body', 'lower-body', 'upper-body'], dtype='<U10')
```

In [18]:

```
import sys
import os

import matplotlib.pyplot as plt
plt.switch_backend('agg')

sys.path.append(os.path.join('/content', 'models/research'))
from object_detection.utils import visualization_utils as vis_util

from PIL import Image as Pil_image

%matplotlib inline
boxes = np.array([oject_detector.img_detections[3]['category_box']])

def load_image_into_numpy_array(image):
  (im_width, im_height) = image.size
  return np.array(image.getdata()).reshape(
      (im_height, im_width, 3)).astype(np.uint8)
# загружаем картинку и превращаем в массив
image = Pil_image.open(file_path)
image_np = load_image_into_numpy_array(image)
# накладываем на массив bounding boxes
vis_util.draw_bounding_boxes_on_image_array(image_np, boxes)
# сохраняем картинку на диск
result_file_path = os.path.join('/content', 'test.png')
vis_util.save_image_array_as_png(image_np, result_file_path)
# виуализируем картинку, которую сохранили
from IPython.display import Image
Image(result_file_path)
```

Out[19]:

In [20]:

```
ls /content
```

```
checkpoints                    datalab               inference_graph
checkpoint_save_20180513.zip   deep_detection_model   models
cocoapi                        fashion_data          TFFashionDetection
data_dir                       frozen_model
```

Находим эмбеддинги изображений

In [21]:

```
from TFFashionDetection import feature_extraction as ftrs

ftrs.get_dir_embeddings(model_dir='/content/deep_detection_model', data_dir='/conter
```

```
IOPub data rate exceeded.
The notebook server will temporarily stop sending output
to the client in order to avoid crashing it.
To change this limit, set the config variable
`--NotebookApp.iopub_data_rate_limit`.

Current values:
NotebookApp.iopub_data_rate_limit=1000000.0 (bytes/sec)
NotebookApp.rate_limit_window=3.0 (secs)
```

In [22]:

```
import pickle

embeds_file = '/content/data_dir/img_embeddings.pkl'

embeds = pickle.load(open(embeds_file, 'rb'))
```

In [23]:

```
from sklearn.neighbors import KDTree
import numpy as np
num_embeds = len(embeds)
embed_index = dict(zip(embeds.keys(), range(num_embeds)))
inverted_index = {j:i for i,j in embed_index.items()}

embed_matrix = np.vstack([embeds[k] for k in embed_index])

kdt = KDTree(embed_matrix, leaf_size=30, metric='euclidean')
```

In [24]:

```
test_id = embed_index[list(embed_index.keys())[620]]

test_example = embed_matrix[test_id]
sim_query = kdt.query(test_example.reshape(1,-1), k=5, return_distance=False)
```

In [27]:

```
file_path = inverted_index[sim_query[0][1]]

Image(file_path)
```

Out[28]: