

Инструкция по обучению детектора гаджетов и прочих устройств с маркированным датасетом

Введение

Данная инструкция предоставляет возможность запустить обучение на основе сформированных маркированных датасетов используя виртуальную машину. При хорошем датасете можно получать уникальные классы объектов, не только девайсы, но и актуальные объекты, например, маски на лице и прочее.

Основные информация перед запуском

1. Поскольку большинство используют OS Windows, запустить первое обучение мы попробуем через [Google Colab](#) сделанный под Linux.

P.S. Если вы все же предпочитаете использовать персональный компьютер, то прочитайте инструкции в следующих разделах репозитория для запуска `darknet` :

- [How to compile on Linux \(using make\)](#).
- [How to compile on Windows \(using CMake\)](#).
- [How to compile on Windows \(using vcpkg\)](#).

```
https://github.com/AlexeyAB/darknet
```

Дальнейшие шаги для всех ОС являются аналогичными.

2. **ВАЖНО.** Если у вас внезапно отключилось обучение или прервалось по причине слабого интернета, то вам необходимо заново выполнить шаги 2, 4, 5 в разделе Запуск программы. Иначе исполняемый файл `darknet` работать не будет. Так же не забудьте снова выбрать GPU для запуска скриптов в Google Collab.
3. Все данные, скрипты и датасеты мы будем хранить в Google Drive.

Создание необходимых компонентов

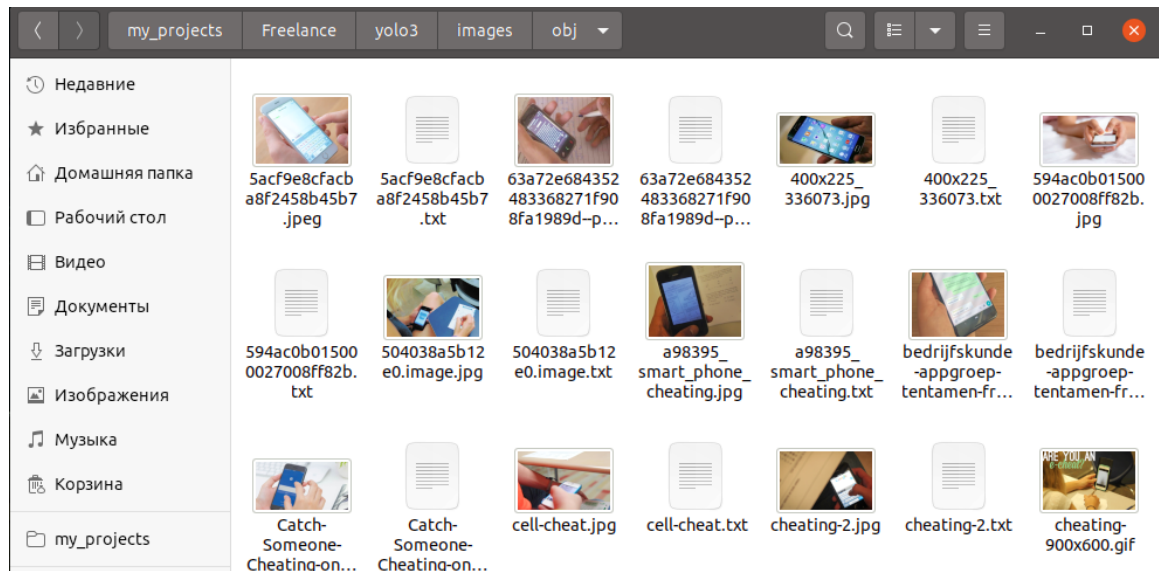
Перед запуском основного обучения необходимо подготовить “почву” в Google Drive.

1. В Google Drive создайте пустую папку (Например, `yolo4`). Внутри созданной папки создайте еще одну папку с именем `training`. Папка `training` предназначена для тренировочных весов. В терминале перейдите в папку `yolo4`:

```
cd yolo4/
```

2. Так же нам нужен датасет полученный по PDF “Инструкция по маркировке датасета для распознавания гаджетов и прочих устройств”.

Для этого на локальном компьютере создайте папку `obj`, поместите туда все изображения в `jpg` формате и соответствующие им `txt` файлы. Помните что названия `jpg` и `txt` файлов должны быть одинаковыми. Преобразуйте папку `obj` в `zip-файл` и поместите набор данных в папку `yolo4` на диске.



3. Создайте файл `obj.names`, поместите данные в том же порядке, что вы указывали в папке `class_list.txt` при маркировке датасета и загрузите на свой диск. Например:

```
mouse
smartphone
keyboard
smartwatch
calculator
```

4. Создайте файл `obj.data`, поместите следующие данные и загрузите их на свой диск.

```
classes = 5
train = data/train.txt
valid = data/test.txt
names = data/obj.names
backup = /mydrive/yolov4/training
```

P.S. На первой строке, укажите то количество классов, которые вы обозначали при маркировке датасета. Остальные данные остаются в таком же виде. В дальнейшем в папку `training` будут попадать обученные веса.

5. Создайте файл `process.py` и загрузите в нее следующий скрипт:

```
import glob, os

# Current directory
current_dir = os.path.dirname(os.path.abspath(__file__))

print(current_dir)

current_dir = 'data/obj'

# Percentage of images to be used for the test set
percentage_test = 10;

# Create and/or truncate train.txt and test.txt
file_train = open('data/train.txt', 'w')
file_test = open('data/test.txt', 'w')

# Populate train.txt and test.txt
counter = 1
index_test = round(100 / percentage_test)
for pathAndFilename in glob.iglob(os.path.join(current_dir, "*.jpg")):
    title, ext = os.path.splitext(os.path.basename(pathAndFilename))

    if counter == index_test:
        counter = 1
        file_test.write("data/obj" + "/" + title + '.jpg' + "\n")
    else:
        file_train.write("data/obj" + "/" + title + '.jpg' + "\n")
        counter = counter + 1
```

Данный скрипт мы запустим чуть позже. Его необходимо так же поместить в папку `yolov4` на своем диске. Этот скрипт поможет разделить набор данных с изображениями на две части: 90% для обучения и 10% для теста. Он создает два текстовых файла `train.txt` и `test.txt` которые имеют пути к вашим изображениям.

6. И последнее, нужно создать свой собственный `config` файл под названием `yolov4-custom.cfg` и аналогично положить его в свой диск в папку `yolov4`. Готовый `config` файл можно получить из каталога `darknet/cfg`, изменив некоторые параметры. Или заново скачать из следующего репозитория файл под названием `cfg/yolov4-custom.cfg`, так же заменив параметры:







<https://github.com/AlexeyAB/darknet>

Какие именно изменения необходимо сделать?

- изменить строку `batch` на `batch=64` (возможно нет необходимости).
- изменить строку `subdivisions` на `subdivisions=16` (возможно нет необходимости).
- поставить значение `width=416` и `height=416` (возможно нет необходимости).
- изменить строку `max_batches` на `max_batches = (количество классов * 2000)`. Однако значение должно быть не ниже 6000 или количества обучающих изображений. К примеру, в нашем случае мы имеем 5 классов, значит `max_batches = 10000`.
- изменить строку `steps` на 80% и 90% от ранее полученного `max_batches`. В нашем случае `steps=8000, 9000`.
- изменить строки `filters=255` (должны быть 3 строки, остальные числа изменять не нужно) на `filters=(количество классов + 5) * 3`. Например, у нас бы вышло `filters=30`.
- изменить строки `classes=80` (должны быть 3 строки, остальные числа изменять не нужно) на количество классов. Например, у нас `classes=5`.

Итак, мы лишь подготовили все необходимые компоненты, далее будем работать над запуском обучения. В итоге наша папка `yolov4` должна выглядеть следующим образом.

Мой диск > yolov4 ▾

Название ↑	Владелец	Последнее изме...
 training	я	23:30
 obj.data	я	25 мая 2022 г.
 obj.names	я	25 мая 2022 г.
 obj.zip	я	25 мая 2022 г.
 process.py	я	23:31
 yolov4-custom.cfg	я	25 мая 2022 г.

Запуск программы

Нижеперечисленные шаги продублированы в [GoogleCollab](#). Далее переходим на [сайт](#) в котором лежит виртуальная машина.

1. Для начала.
 - а. Сделайте копию “Инструкции по обучению детектора гаджетов и прочих устройств с маркированным датасетом.ipynb” себе на диск, нажав “Файл” в строке “меню” и “Сохранить копию на диске”. Теперь вы сможете подключиться к виртуальной машине Google Colab и/или запускать скрипты данного файла.
 - б. Нажмите “Среда выполнения” в строке “меню” и нажмите “Изменить тип среды выполнения”. Выберите “GPU” и нажмите “Сохранить”.
2. Откройте диск и перейдите в созданную папку `yolov4`
3. Сделайте клон git-репозитория `darknet` в папку `yolov4`

Мой диск > yolov4 ▾			
Название ↑	Владелец	Последнее изме...	
darknet	я	23:48	
training	я	23:30	
obj.data	я	25 мая 2022 г.	
obj.names	я	25 мая 2022 г.	
obj.zip	я	25 мая 2022 г.	
process.py	я	23:31	
yolov4-custom.cfg	я	25 мая 2022 г.	

4. Внесите изменения в `makefile`, чтобы включить `OPENCV` и `GPU`
5. Запустите команду `make` для создания даркнета


```
5. Запустите команду make для создания даркнета





# build darknet
!make

In file included from src/yolo_v2_class.cpp:2:0:
include/yolo_v2_class.hpp: In member function 'void track kalman t::clear old_states()':
include/yolo_v2_class.hpp:879:50: warning: comparison between signed and unsigned integer expres
    if ((result_vec_pred[state_id].x > img_size.width) ||
include/yolo_v2_class.hpp:880:50: warning: comparison between signed and unsigned integer expres
    (result_vec_pred[state_id].y > img_size.height))
include/yolo_v2_class.hpp: In member function 'track kalman t::tst t track kalman t::get state i
include/yolo_v2_class.hpp:900:30: warning: comparison between signed and unsigned integer expres







✓ 1m 44s completed at 11:53 PM
```

6. Скопируйте все созданные нами ранее файлы из папки `yolov4` в директорию `darknet`

Мой диск > yolov4 > darknet > cfg ▾		
Название ↑	Владелец	Последнее изме...
 yolov4-custom.cfg	я	25 мая 2022 г.

Мой диск > yolov4 > darknet > data ▾		
Название ↑	Владелец	Последнее изме...
 labels	я	23:48
 obj	я	23:56
 obj.data	я	25 мая 2022 г.
 obj.names	я	25 мая 2022 г.

7. Запустите скрипт `process.py`, чтобы создать файлы `train.txt` и `test.txt` внутри папки data

Мой диск > yolov4 > darknet > data ▾		
Название ↑	Владелец	Последнее изменение
 labels	я	23:48 я
 obj	я	23:56 я
 obj.data	я	25 мая 2022 г. я
 obj.names	я	25 мая 2022 г. я
 test.txt	я	00:07 я
 train.txt	я	00:07 я

8. Загрузите предварительно обученные `yolov4` веса
9. Приступаем к обучению.

9. Приступаем к обучению

```
# train your custom detector! (uncomment %%capture below if you run into memory issues or your Colab
# %%capture

!./darknet detector train data/obj.data cfg/yolov4-custom.cfg yolov4.conv.137 -dont_show -map

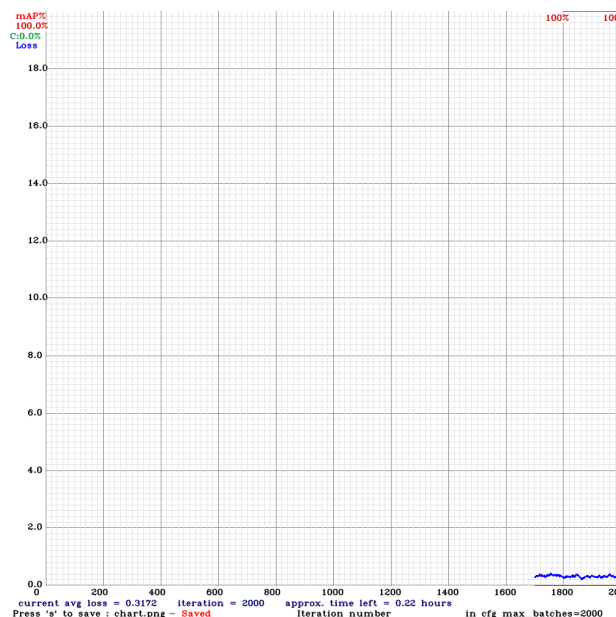
125 conv 512 3 x 3/ 1 26 x 26 x 256 -> 26 x 26 x 512 1.595 BF
126 conv 256 1 x 1/ 1 26 x 26 x 512 -> 26 x 26 x 256 0.177 BF
127 conv 128 1 x 1/ 1 26 x 26 x 256 -> 26 x 26 x 128 0.044 BF
128 upsample 2x 26 x 26 x 128 -> 52 x 52 x 128
129 route 54 -> 52 x 52 x 256
130 conv 128 1 x 1/ 1 52 x 52 x 256 -> 52 x 52 x 128 0.177 BF
131 route 130 128 -> 52 x 52 x 256
132 conv 128 1 x 1/ 1 52 x 52 x 256 -> 52 x 52 x 128 0.177 BF
133 conv 256 3 x 3/ 1 52 x 52 x 128 -> 52 x 52 x 256 1.595 BF
134 conv 128 1 x 1/ 1 52 x 52 x 256 -> 52 x 52 x 128 0.177 BF
135 conv 256 3 x 3/ 1 52 x 52 x 128 -> 52 x 52 x 256 1.595 BF
136 conv 128 1 x 1/ 1 52 x 52 x 256 -> 52 x 52 x 128 0.177 BF
137 conv 256 3 x 3/ 1 52 x 52 x 128 -> 52 x 52 x 256 1.595 BF
138 conv 18 1 x 1/ 1 52 x 52 x 256 -> 52 x 52 x 18 0.025 BF
139 yolo
[yolo] params: iou loss: ciou (4), iou_norm: 0.07, obj_norm: 1.00, cls_norm: 1.00, delta_norm: 1.00
nms kind: greedy (1), beta = 0.600000
140 route 136 -> 52 x 52 x 128
141 conv 256 3 x 3/ 2 52 x 52 x 128 -> 26 x 26 x 256 0.399 BF
142 route 141 126 -> 26 x 26 x 512
143 conv 256 1 x 1/ 1 26 x 26 x 512 -> 26 x 26 x 256 0.177 BF
144 conv 512 3 x 3/ 1 26 x 26 x 256 -> 26 x 26 x 512 1.595 BF
145 conv 256 1 x 1/ 1 26 x 26 x 512 -> 26 x 26 x 256 0.177 BF
146 conv 512 3 x 3/ 1 26 x 26 x 256 -> 26 x 26 x 512 1.595 BF
147 conv 256 1 x 1/ 1 26 x 26 x 512 -> 26 x 26 x 256 0.177 BF
148 conv 512 3 x 3/ 1 26 x 26 x 256 -> 26 x 26 x 512 1.595 BF
149 conv 18 1 x 1/ 1 26 x 26 x 512 -> 26 x 26 x 18 0.012 BF
150 yolo
[yolo] params: iou loss: ciou (4), iou_norm: 0.07, obj_norm: 1.00, cls_norm: 1.00, delta_norm: 1.00
nms kind: greedy (1), beta = 0.600000
151 route 147 -> 26 x 26 x 256
152 conv 512 3 x 3/ 2 26 x 26 x 256 -> 13 x 13 x 512 0.399 BF
153 route 152 116 -> 13 x 13 x 1024
154 conv 512 1 x 1/ 1 13 x 13 x 1024 -> 13 x 13 x 512 0.177 BF
```

Если вы видите такую картину на 9-ом этапе, поздравляю, у вас получилось запустить процесс обучения без ошибок! :) Этот этап может занять от 1 до 3 часов. Если у вас очень большой датасет и большое количество классов, то будьте готовы к долгому процессу который может продлиться еще дольше.

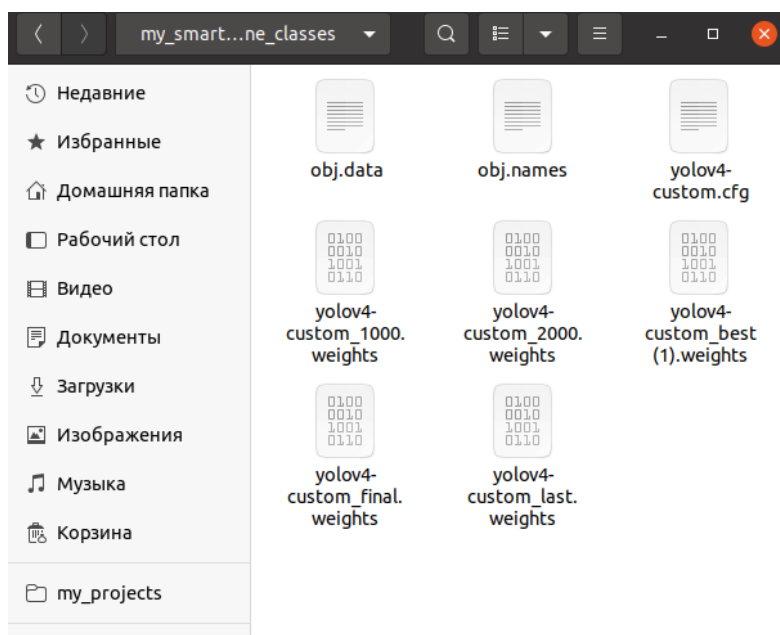
Если у вас вдруг случился форс-мажор и обучение было прервано по каким то причинам, то начинать все заново нет никакой необходимости. Вам лишь нужно обновить страницу, заново выбрать GPU, запустить 2, 4 и 5 шаги, и запустить код указанный в [“Инструкции по обучению детектора гаджетов и прочих устройств с маркированным датасетом.ipynb”](#) в конце 9-го этапа.

10. Проверить производительность и правильность полученных результатов

На данном этапе вы получите график производительности, который выводится только если вы выполнили все предыдущие шаги верно и не было прерываний во время обучения. Картинка примерно выглядит вот так:



Если у вас появилась эта картинка, то обучение было завершено успешно. Чтобы проверить что у вас в итоге получилось, запустите предпочитаемый способ из 11. Если у вас уже есть готовый проект, скачайте все необходимые файлы, в том числе сформированный **weight** файлы в свой проект.

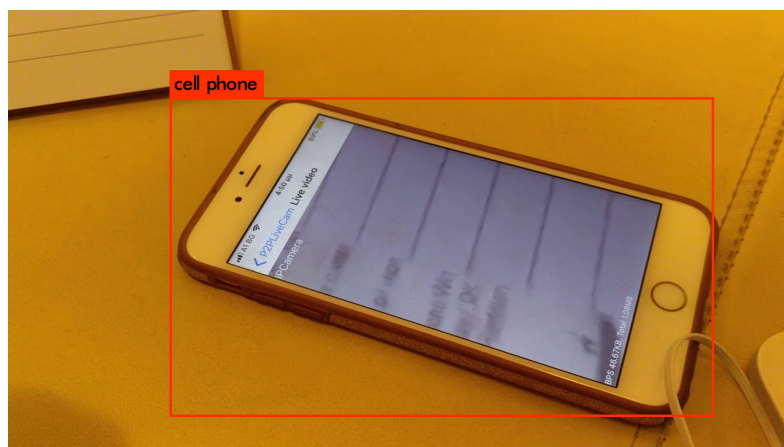


В качестве выполненного задания для проверки принимаются данные файлы, а так же **obj.zip** файл. Архивируйте все файлы и отправьте преподавателю.

11. Протестируйте свой пользовательский детектор объектов

- a. Запустите детектор на изображении
- b. Запустите детектор на изображении с веб-камеры

- c. Запустить детектор на видео
- d. Запустить детектор на видео с веб-камеры



Источники

- ☐ TRAIN A CUSTOM YOLOv4 OBJECT DETECTOR (Using Google Colab).
<https://medium.com/analytics-vidhya/train-a-custom-yolov4-object-detector-using-google-colab-61a659d4868>