|  |
| --- |
|  |
| **МИНОБРНАУКИ РОССИИ** |
| **Федеральное Государственное бюджетное образовательное учреждение высшего образования****«МИРЭА –Российский технологический университет»** |

**Лабораторный практикум**

**(Практическая часть)**

**ЦИФРОВАЯ ОБРАБОТКА ИЗОБРАЖЕНИЙ**

**Тема: НЕЙРОННЫЕ СЕТИ**

Москва 2019

**Простой классификатор изображений на Python с помощью библиотеки TensorFlow**



[Martin Leblanc](https://www.iconfinder.com/icondesigner); FlorianCassayre

Данная лаборатория посвящена созданию классификатора изображений, используя TensorFlow — библиотеку машинного обучения от Google.

**Для выполнения лабораторной работы понадобится:**

* компьютер с 64-битной операционной системой Ubuntu 16.04 или выше (если используется Windows, рекомендуется установить виртуальную машину);
* библиотека TensorFlow версии 1.13 или выше, но не 2.0;
* язык Python версии 3.6;
* утилита curl

**Постановка задачи**

Что должен выполнить алгоритм:

* проанализировать изображение и определить, есть ли на нём цветок;
* если есть — определить, к какой из предложенных категорий он относится;
* вывести процент уверенности в своём ответе.

Категории цветов, включенные во входной набор данных: ромашки, одуванчики, розы, подсолнухи и тюльпаны.

Мы будем использовать Inception v3 — готовую модель машинного обучения для тренировки на нашем собственном наборе данных. Эта нейронная сеть используется в «Кампании по широкомасштабному распознаванию образов ImageNet» (ILSVRC — ImageNet Large Scale Visual Recognition Challenge) и ранее на ней проводилось обучение описываемого алгоритма на определение категорий объектов на изображениях.

Настройка инструментов

Установите Python, менеджер пакетов pip и утилиту curl. Для этого необходимо открыть терминал Ubuntu и ввести следующие команды:

sudo apt update

sudo apt install python3 pip3 curl

Установите tensorflow

pip3 install tensorflow tensorflow-hub

**Подготовка входных данных для обучения**

Чтобы модель Inception v3 могла определять цветы на изображениях, необходимо подготовить входные данные, на которых алгоритм будет обучаться. Для этого сначала следует создать каталог на вашем основном компьютере / виртуальной машине с названием tf\_files

mkdir tf\_files

а затем перейти в него

cd tf\_files

Загрузите архив с изображениями цветов, которые будут входными данными

curl -O http://download.tensorflow.org/example\_images/flower\_photos.tgz

Разархивируйте полученный файл командой:

tar xzf flower\_photos.tgz

В каталоге flower\_photos вы увидите поддиректории daisy (ромашки), dandelion (одуванчики), roses (розы), sunflowers (подсолнухи) и tulips (тюльпаны).

**Получение примера исходного кода**

В каталоге tf\_files введите команду

curl -O \ https://raw.githubusercontent.com/tensorflow/hub/master/examples/image\_retraining/retrain.py

Для обучения введите следующую команду:

python3 ./retrain.py \

--bottleneck\_dir=bottlenecks \

--how\_many\_training\_steps 500 \

--model\_dir=inception \

--output\_graph=retrained\_graph.pb \

--output\_labels=retrained\_labels.txt \

--image\_dir flower\_photos

Пока модель Inception обрабатывается (а это займёт продолжительное время), вы увидите, что сначала создаются критические параметры (bottlenecks), как это показано на скриншоте ниже. Критические параметры — это последний слой перед основным слоем начальной модели, который будет классифицировать изображения. В них заключается содержательное, но при этом компактное резюме изображения, которое помогает классификатору сделать наиболее точный выбор:



После того, как обучение критических параметров завершится, алгоритм перейдёт на финальный этап обучения. Во время обучения в терминале будут отображены следующие параметры: точность обучения (train accuracy) и точность проверки (validation accuracy). Вы увидите аналогичную скриншоту ниже картину. В конце будет отображено значение точности от 85% до 99%:



**Тестирование классификатора изображений**

В каталог tf\_files загрузите файл label\_image.py с помощью команды

curl -O \ https://raw.githubusercontent.com/tensorflow/tensorflow/master/tensorflow/examples/label\_image/label\_image.py

Выполните следующую команду, чтобы протестировать, например, изображение, на котором присутствует ромашка:

python3 ./label\_image.py --graph=retrained\_graph.pb \

--image=flower\_photos/daisy/21652746\_cc379e0eea\_m.jpg \

--labels=retrained\_labels.txt \

--output\_layer=final\_result \

--input\_layer=Placeholder

Как видно на скриншоте ниже, при тестировании выбранного изображения было получено число 0,98929. Это указывает на то, что классификатор на 98% уверен, что изображение содержит цветок ромашки. Если аналогичным образом протестировать изображения собаки и кота, будет получен низкий процент уверенности для всех категорий цветов. Это говорит о том, что алгоритм определил, что наличие какого-либо цветка маловероятно на тестируемом изображении:

