Programming Assignment: Анализ изображений

You have not submitted. You must earn 1/3 points to pass.

i

It looks like this is your first programming assignment.  Learn more

|  |  |
| --- | --- |
| **Deadline** | The assignment was due on April 23, 11:59 PM PDT  You can still pass this assignment before the course ends. |

1. [**Instructions**](https://www.coursera.org/learn/data-analysis-applications/programming/CmDGQ/analiz-izobrazhienii)
2. [My submission](https://www.coursera.org/learn/data-analysis-applications/programming/CmDGQ/analiz-izobrazhienii/submission)
3. [Discussions](https://www.coursera.org/learn/data-analysis-applications/programming/CmDGQ/analiz-izobrazhienii/discussions)

Задача заключается в том, чтобы применить предобученную на imagenet нейронную сеть на практической задаче классификации автомобилей.

Учиться применять нейронные сети для анализа изображений мы будем на библиотеке TensorFlow. Это известный опенсорсный проект, разработанный инженерами Google Brain Team. Подробнее почитать о TensorFlow можно на официальном сайте, на [гитхабе](https://github.com/tensorflow/tensorflow) или на [хабре](https://habrahabr.ru/post/270543/).

**Установка окружения**

Необходимо установить **TensorFlow.**

* [Инструкции по установке на сайте](https://www.tensorflow.org/versions/r0.10/get_started/os_setup.html#download-and-setup).
* Если есть опыт работы с Docker, то можно воспользоваться готовым [докер-контейнером с тензорфлоу](https://www.tensorflow.org/versions/r0.10/get_started/os_setup.html#docker-installation). Чтобы пробросить данные в докер, можно воспользоваться флагом **-v /src/webapp:/opt/webapp** при запуске докер контейнера. Тогда директория на хосте **/src/webapp** будет доступна в контейнтере по **/opt/webapp**.

**Важно!**Если вы пользователь windows, то уставить tensorflow можно только в том случае, если вы используете Python 3.5.2 и выше. Это можно сделать с помощью команды pip install tensorflow.

Пользователи Windows 10 и выше могут воспользоваться **Docker** ([ссылка на дистрибутив](https://www.docker.com/products/docker#/windows)).

Если у вас возникли проблемы по работе с Tensorflow из Docker контейнера, воспользуйтесь [F.A.Q.](https://github.com/ton4eg/coursera_pa/blob/master/faq/tensorflow_docker_faq.ipynb)

Если же поставить Tensorflow на вашу машину никак не получается, мы предлагаем воспользоваться одним из облачных сервисов, в который необходимо установить линукс-образ. Самые популярные облачные сервисы [AWS](https://aws.amazon.com/) и [DigitalOcean](http://digitalocean.com/) предоставляют бесплатные инстансы (имейте в виду, что для того, чтобы ими воспользоваться, нужно будет привязать кредитную карту).

Чтобы освоить компьютерное зрение (или другие интересные задачи из области ML и AI), так или иначе придётся научиться работать с библиотеками нейронных сетей, линуксом и виртуальными серверами. Например, для более масштабных практических задач, крайне необходимы сервера с GPU, а с ними уже локально работать не получиться.Тем не менее, мы понимаем, что в силу временных ограничений курса кто-то может успеть установить TensorFlow. Поэтому мы сделали пункты 1 и 2 необязательными. На оценку они не повлияют — можете сразу переходить к третьему пункту.Помимо tensorflow, потребуется библиотека **scipy**. Если вы уже работали с Anaconda и/или выполняли задания в нашей специализации, то она должна присутствовать.

**Данные**

Скачать данные нужно тут: <https://yadi.sk/d/6m_KbM4HvmLfs>

Данные это часть выборки *Cars Dataset*([link](http://ai.stanford.edu/~jkrause/cars/car_dataset.html)). Исходный датасет содержит 16,185 изображений автомобилей, принадлежащих к 196 классам. Данные разделены на 8,144 тренировочных и 8,041 тестовых изображений, при этом каждый класс разделён приблизительно поровну между тестом и трейном. Все классы уровня параметров *Марка*, *Год*, *Модель* и др. (например,*2012 Tesla Model S or 2012 BMW M3 coupe*). В нашем же случае в**train**204 изображения, и в **test** — 202 изображения.

**Что делать**Помимо данных, потребуется скачать:

* [код](https://github.com/ton4eg/coursera_pa),
* веса модели [по ссылке](https://yadi.sk/d/F6_ZiHHI3GMDGZ).

Положите данные, код и модель в одну папку. У вас должна получиться такая структура:

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

/assignment-computer-vision/

|

|-- test # папки

| `---- ... # с

|-- train # картинками

| `---- ...

|

|-- class\_names.txt # имена классов, номер строки соответствует id класса

|-- results.txt # соответствие имя картинки — id класса

|-- vgg16\_weights.npz # веса модели в формате tensorflow

|

|-- vgg16.ipynb # основной скрипт

|-- imagenet\_classes.py

|

`-- beach.jpg # картиночка с пляжем

**Задание 1.**

Для начала нужно запустить готовую модель **vgg16**, предобученную на **imagenet**. Модель обучена с помощью **caffe** и сконвертирована в формат**tensorflow** - **vgg16\_weights.npz**. Скрипт, иллюстрирующий применение этой модели к изображению, находится внутри ноутбука **vgg16.ipynb** и возвращает топ-5 классов из **imagenet**, а так же уверенность в этих классах.

***Задание:***

Загрузите уверенность первого класса для изображения **train/00002.jpg** с точностью до 1го знака после запятой.

**Задание 2.**

Научитесь извлекать**fc2** слой. Для этого нужно модифицировать **process\_image**, чтобы вместо последнего слоя извлекались выходы **fc2**.

***Задание:***

Посчитайте **fc2** для картинки **train/00002.jpg**. Запишите первые 20 компонент (каждое число с новой строки, т.е. в загружаемом файле должно получиться 20 строк).

**Задание 3.**

Теперь необходимо дообучить классификатор на нашей базе. В качестве бейзлайна предлагается воспользоваться классификатором **svm** из пакета **sklearn**.

* Модифицировать функцию **get\_features** и добавить возможность вычислять **fc2**. (Мы это уже сделали в**process\_image**).
* Применить **get\_features**, чтобы получить **X\_test** и **Y\_test**.
* Воспользоваться классификатором, **SVC** с **random\_state=0**.

**Важно!** Если вам не удалось поставить **tensorflow**, то необходимо вместо использования функции **get\_features**, загрузить предпосчитанные **X**, **Y**, **X\_test**, **Y\_test** и воспользоваться функцией**np.load** для их загрузки, а после этого два последних пункта.

[X.npy](https://d3c33hcgiwev3.cloudfront.net/_103342682036906d018098b7fc6a86b2_X.npy?Expires=1493683200&Signature=HExneh4AbrID3ZceXDTCowj9gxqRphmmsDWWNtYZBGLLIgXXNafNMk27Lut-ALJWOsh5ViA5suRoSlT5uw0x9IygLXNoxcJ2Oldw59tmG6rQJ-HODAbfm-qj1AAhfbkC8xevwOu76KfuKYDhBDlfN4B5B815iZtDrCkfCYFQvAg_&Key-Pair-Id=APKAJLTNE6QMUY6HBC5A)

[Y.npy](https://d3c33hcgiwev3.cloudfront.net/_103342682036906d018098b7fc6a86b2_Y.npy?Expires=1493683200&Signature=geJJlf26lY5-56mElPy9eD4ZPnX9X7rOwH1KeE3O1wFr86iIgnDCxkDS-IFbXs7q0bIBrv45uTERrFHq587iF6Q98IdIdR61rO~U2f5uXLrN6kNDk0ncroSC9LgFarxBRKPbXifXUCUOQPspLLA~SbLVRGIGWkzUKzOoxT8brPU_&Key-Pair-Id=APKAJLTNE6QMUY6HBC5A)

[X\_test.npy](https://d3c33hcgiwev3.cloudfront.net/_103342682036906d018098b7fc6a86b2_X_test.npy?Expires=1493683200&Signature=AgET2du2-vKMMMKlkmqVNS7XcyALrCmcabpmDERYS1Kr~82oirI6THP5~gcXErUGWz3ScEsihi07N2H5pqx1Q6YXPO-eul4YO4SRIVIGNHDhGNmxkEPNoDOzijVn1GR6683x3aStsjT6BY5zPz-3qj8Tfj8BUFvuEAdTskbYPn0_&Key-Pair-Id=APKAJLTNE6QMUY6HBC5A)

[Y\_test.npy](https://d3c33hcgiwev3.cloudfront.net/_103342682036906d018098b7fc6a86b2_Y_test.npy?Expires=1493683200&Signature=DF6JJfhYIuvQV1oCbaBS1PXqw6rB-jIUEn08EthEqNFDxePWvoOfmNJbnzgXuGp61zZuv9vb1TZTD3KutnETDb0LihkSXvzoWwPQOmycnls~5RHYJHLGZ6slrdCY3JBolFq21Z7Qgc4LtX7n99UNT5kEu7NZ33qGPlikKTCkYO0_&Key-Pair-Id=APKAJLTNE6QMUY6HBC5A)

***Задание:***

Сколько правильных ответов получается на валидационной выборке из папки **test**? Запишите в файл.

How to submit

When you're ready to submit, you can upload files for each part of the assignment on the "My submission" tab.