|  |  |
| --- | --- |
| Федеральное агентство железнодорожного транспорта  Омский государственный университет путей сообщения  Кафедра «Автоматика и системы управления»  ЗНАКОМСТВО С PYTHON  Лабораторная работа № 8  по теме «Информационные технологии» | |
|  | Студенты гр. 23к                           Лихтнер К.Е., Атаманчук О.А., Дубровский Н.С., Речута Д.А.  «  5  »        мая          2024 г.  Руководитель –  старший преподаватель кафедры «АиСУ»                           Т.В. Васеева  «    »                  2024 г. |
| Омск 2024 | |

# Цель работы

Знакомство с языком программирования python, а также с сортировкой фото с помощью нейронной сети.

# Выполнение работы

## Подготовка среды

## Перед началом работы над лабораторной установим нужную среду (рисунки 1-3).

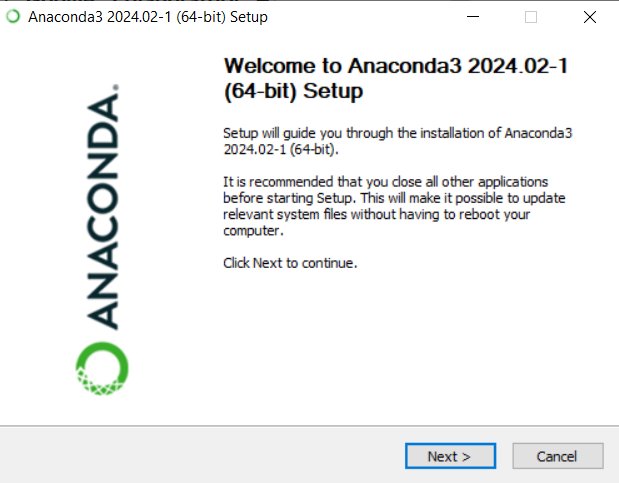


Рисунок 1 – Установка среды Anaconda

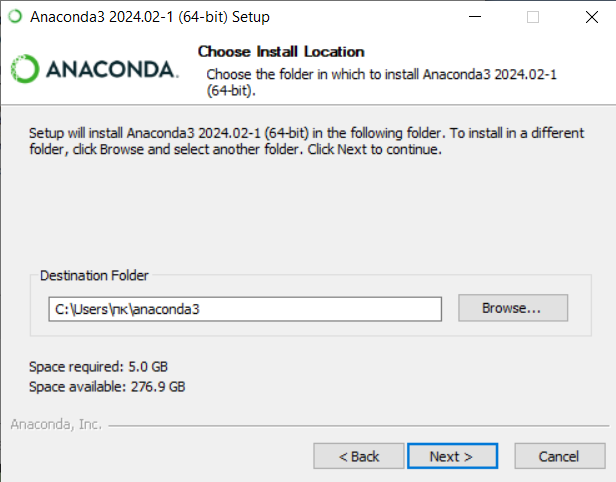


Рисунок 2 – Установка среды Anaconda

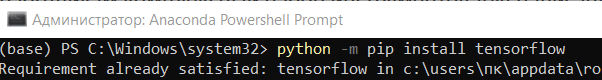


Рисунок 3 – Установка через консоль

Далее перейдем работать в Visual Studio Code, в нем открываем папку с нужными файлами (рисунок 4).

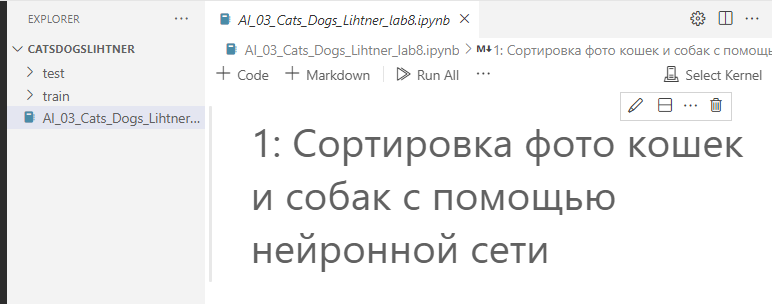


Рисунок 4 – Первый шаг

## Обучение нейронной сети

Для начала в коде исправим путь к файлу на актуальные данные (рисунок 5).

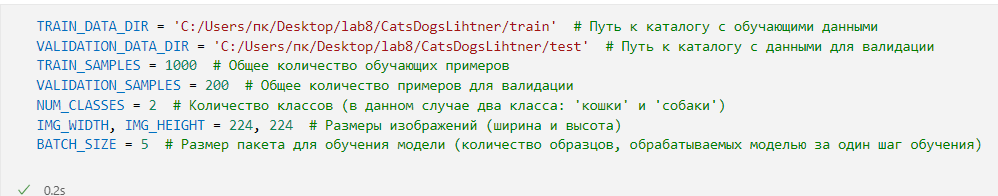


Рисунок 5 – Путь к файлу

Также, путем проб и ошибок, подберем количество эпох и нейронов для наилучшего обучения модели (рисунок 6).

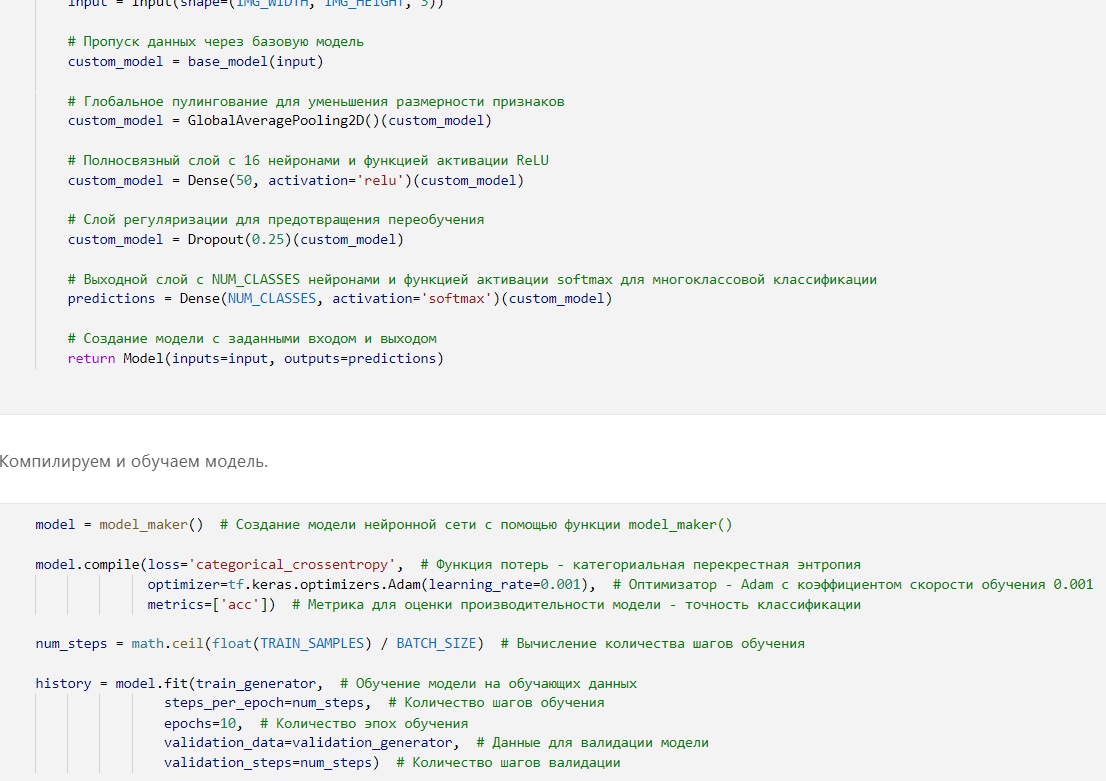


Рисунок 6 – Количество

Результат наиболее удачного подбора параметров можно наблюдать в выстроенном графике. Если обе линии на нем идут вниз следственно, обучение прошло удачно (рисунок 7).

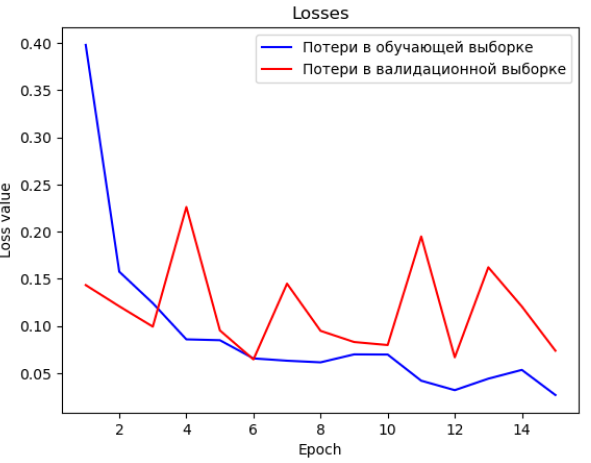


Рисунок 7 – График потерь при обучении

После коррекции кода, запустим его. Таким образом, благодаря коду мы обучаем искусственный интеллект, теперь он может отличать кошечек и собачек друг от друга (рисунок 8).

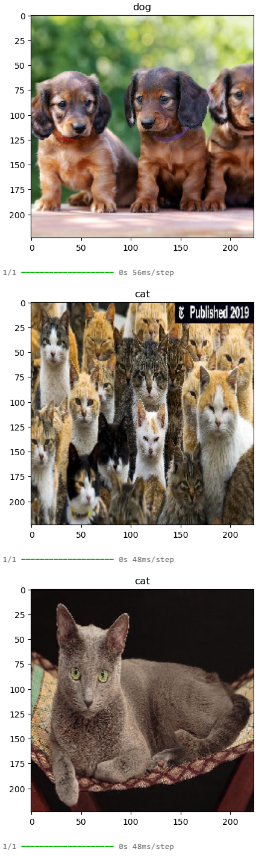


Рисунок 8 – Результат обучения

# Защита

Для того чтобы закрепить полученный знания мы решили сделать свой вариант обучения искусственного интеллекта. Теперь наша машина может отличать больших Годзилл от маленьких пушистых шиншилл с минимальной потерей в выборке (рисунки 9-11). <https://drive.google.com/drive/folders/1UR-93YkfM35Lhj7fSafolpraFlROAyTy?usp=sharing>

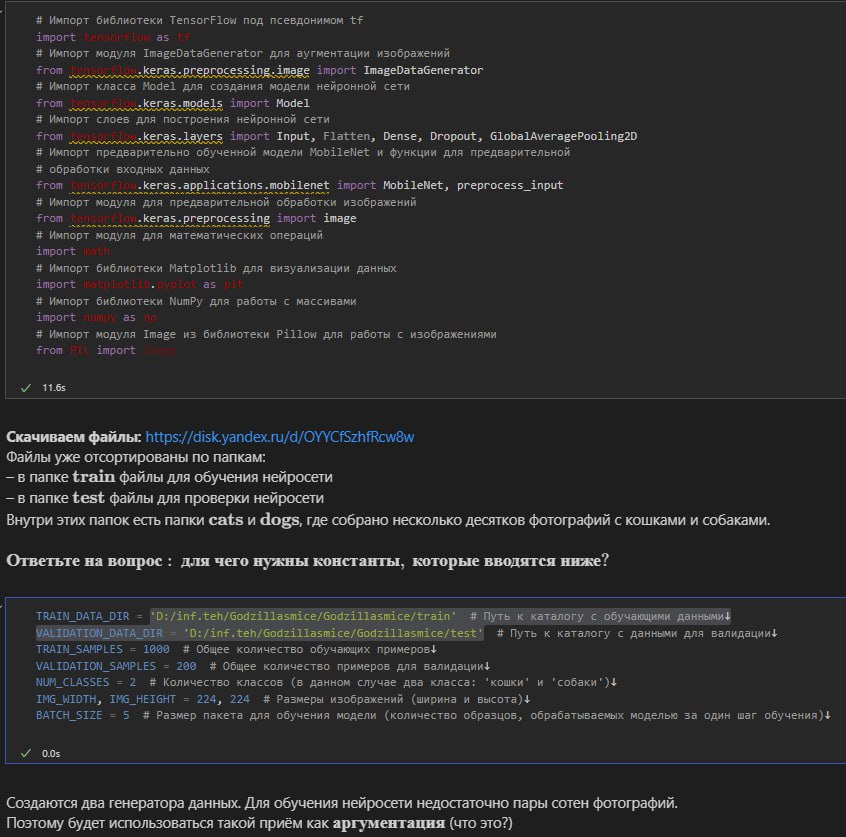


Рисунок 9 – Путь к файлам

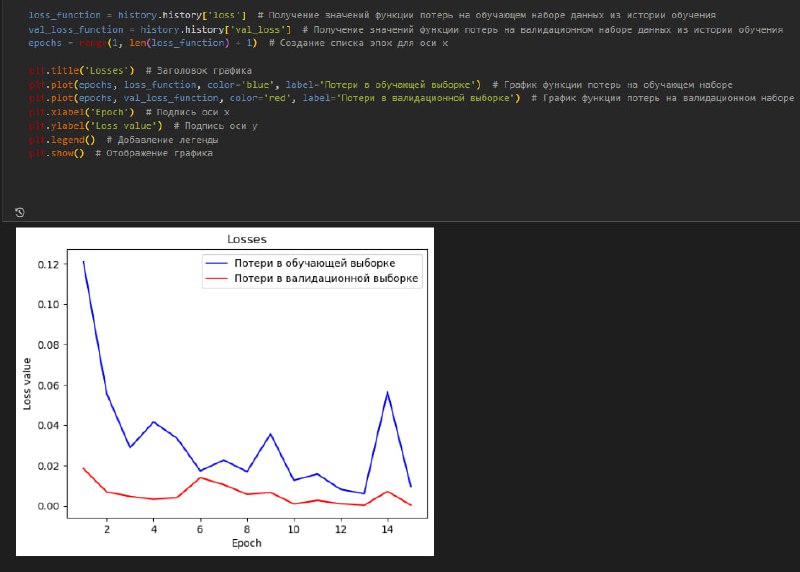


Рисунок 10 – График потерь выборки

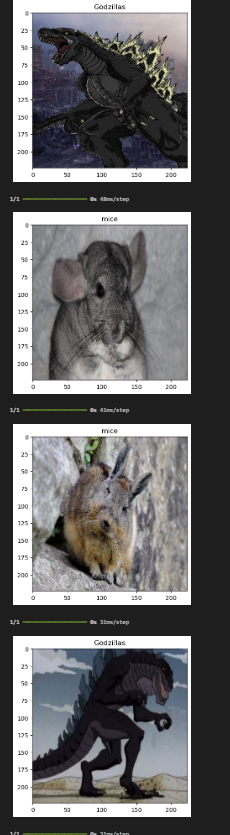


Рисунок 11 – Результат

А также людей с блондинистым цветом волос от брюнетов и брюнеток (рисунки 12-14).

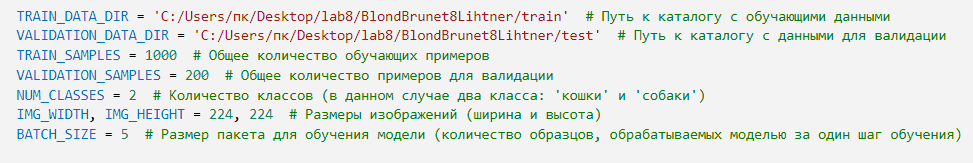


Рисунок 12 – Путь к файлам

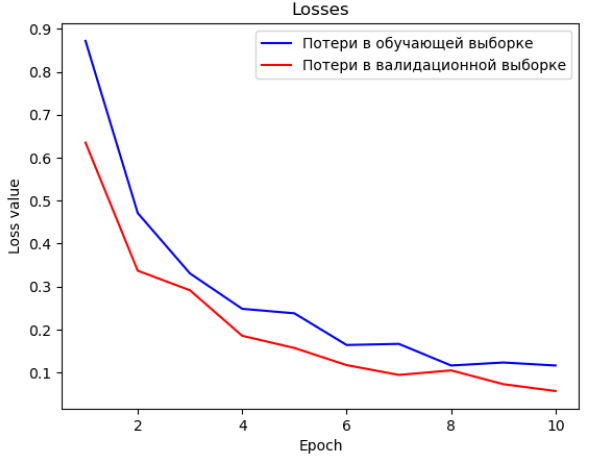


Рисунок 13 – График потерь выборки

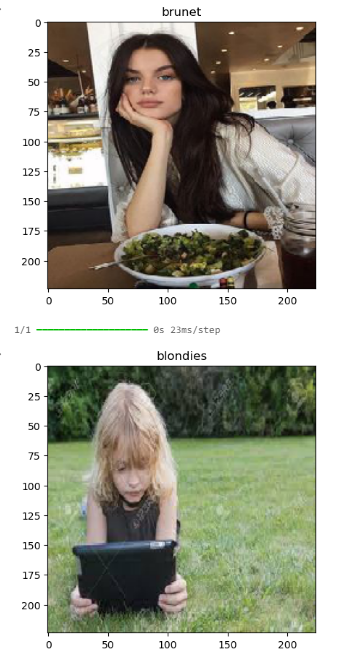


Рисунок 14 – Результат

Мы также дополнили код чтобы выборка была из всех случайных изображений папки test и вывели их отображения с предсказанием модели (рисунки 15-16). В результате чего получили ответы на фото из папки test.

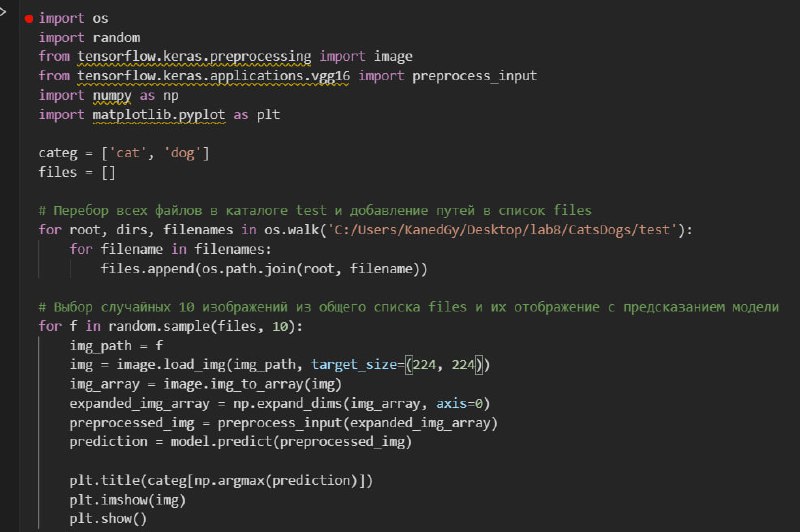


Рисунок 15 – Дополнение к коду

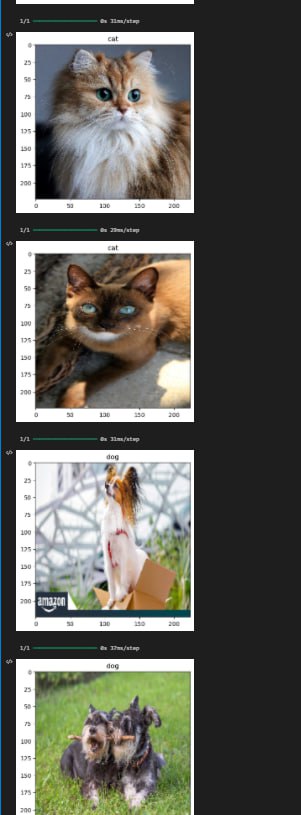


Рисунок 16 – Результат

# Контрольные вопросы

1) Вот несколько популярных архитектур нейронных сетей, помимо MobileNet:

1. ResNet (Residual Neural Network): ResNet представляет собой глубокую нейронную сеть, в которой внедрены "residual connections" для борьбы с проблемой затухания градиента при обучении глубоких сетей.

2. VGG (Visual Geometry Group): VGG - это набор сверточных нейронных сетей, разработанных в Университете Оксфорда, известных своей простой структурой и хорошими результатами на задачах классификации изображений.

3. Inception (GoogLeNet): Inception представляет собой серию нейронных сетей, разработанных Google, которые используют модули "Inception" для эффективного извлечения признаков изображений.

4. AlexNet: AlexNet была одной из первых сверточных нейронных сетей, которая дала толчок развитию глубокого обучения и победила на соревновании ImageNet в 2012 году.

5. DenseNet: DenseNet предлагает архитектуру, в которой каждый слой соединен со всеми последующими слоями в блоке, что способствует лучшему использованию градиентов и повышению эффективности обучения.

6. Transformer: Transformer - это архитектура, изначально разработанная для задач обработки естественного языка, но успешно применяемая и в области компьютерного зрения для задач, таких как распознавание изображений и сегментация.

2) Количество эпох обучения - это один из важных гипер параметров при обучении нейронных сетей, включая MobileNet. Эпоха представляет собой один проход по всем обучающим данным нейронной сети. Вот как количество эпох обучения может повлиять на результаты нейронной сети MobileNet:

1. Недостаточное количество эпох: Если выбрано слишком малое количество эпох, нейронная сеть может не успеть достаточно "увидеть" данные и выучить закономерности в них. Это может привести к не дообучению, когда модель не способна хорошо обобщать данные.

2. Оптимальное количество эпох: Правильно подобранное количество эпох позволяет модели улучшить свою производительность, выучить закономерности в данных и достичь хорошей обобщающей способности.

3. Избыточное количество эпох: слишком много эпох может привести к переобучению модели, когда она начинает "запоминать" обучающие данные вместо того, чтобы обобщать их. Это может привести к плохой производительности модели на новых данных.

Для MobileNet и других нейронных сетей важно проводить эксперименты с разным количеством эпох и следить за метриками производительности модели на валидационном наборе данных. Оптимальное количество эпох обучения будет зависеть от сложности задачи, объема данных, архитектуры модели и других факторов. Тщательный подбор количества эпох поможет достичь хороших результатов при обучении нейронной сети.

3) TRAIN\_DATA\_DIR = 'CatsDogs/train' -- Путь к каталогу с обучающими данными

VALIDATION\_DATA\_DIR = 'CatsDogs/test' -- Путь к каталогу с данными для валидации

TRAIN\_SAMPLES = 1000 -- Общее количество обучающих примеров

VALIDATION\_SAMPLES =200 -- Общее количество примеров для валидации

NUM\_CLASSES = 2 -- Количество классов (в данном случае два класса: 'кошки' и 'собаки')

IMG\_WIDTH, IMG\_HEIGHT = 224, 224 -- Размеры изображений (ширина и высота)

BATCH\_SIZE = 5 -- Размер пакета для обучения модели (количество образцов, обрабатываемых моделью за один шаг обучения)

4) Аугментация данных (data augmentation) - это процесс искусственного создания новых обучающих примеров путем внесения небольших изменений в существующие данные. В контексте изображений это может включать в себя повороты, отражения, изменения масштаба, сдвиги, изменения яркости, добавление шума и другие подобные преобразования. Эти изменения позволяют создать разнообразие входных данных для обучения модели.

## Вывод

В данной лабораторной работе, мы начали знакомиться и работать с языком программирования питон. Научились обучать искусственный интеллект, способом без учителя. Теперь мы умеем обучать нейронную сеть различать предметы на фото. Подбирать для этого наиболее удачные параметры и тому подобное. Также научились устанавливать такие среды для работы как: Anaconda, Jupyter Notebook и им подобные.