|  |  |
| --- | --- |
| Федеральное агентство железнодорожного транспорта  Омский государственный университет путей сообщения  Кафедра «Автоматика и системы управления»  ЗНАКОМСТВО С PYTHON  Лабораторная работа № 8  по дисциплине «Информационные технологии» | |
|  | Студент гр. 23М                            М.П. Сидоренко                            А.Н. Терехов                            А.С. Ципичев                            И.А. Черноморов  «  23  »        Мая         2024 г.  Руководитель –  старший преподаватель кафедры «АиСУ»                            Т.В. Васеева  «    »                  2024 г. |
| Омск 2024 | |

# Цель работы

Целью данной лабораторной работы является знакомство языком программирования Python, а также с сортировкой фото с помощью нейронной сети.

# Выполнение работы

## Для начала среду разработки Anaconda (рисунок 1)

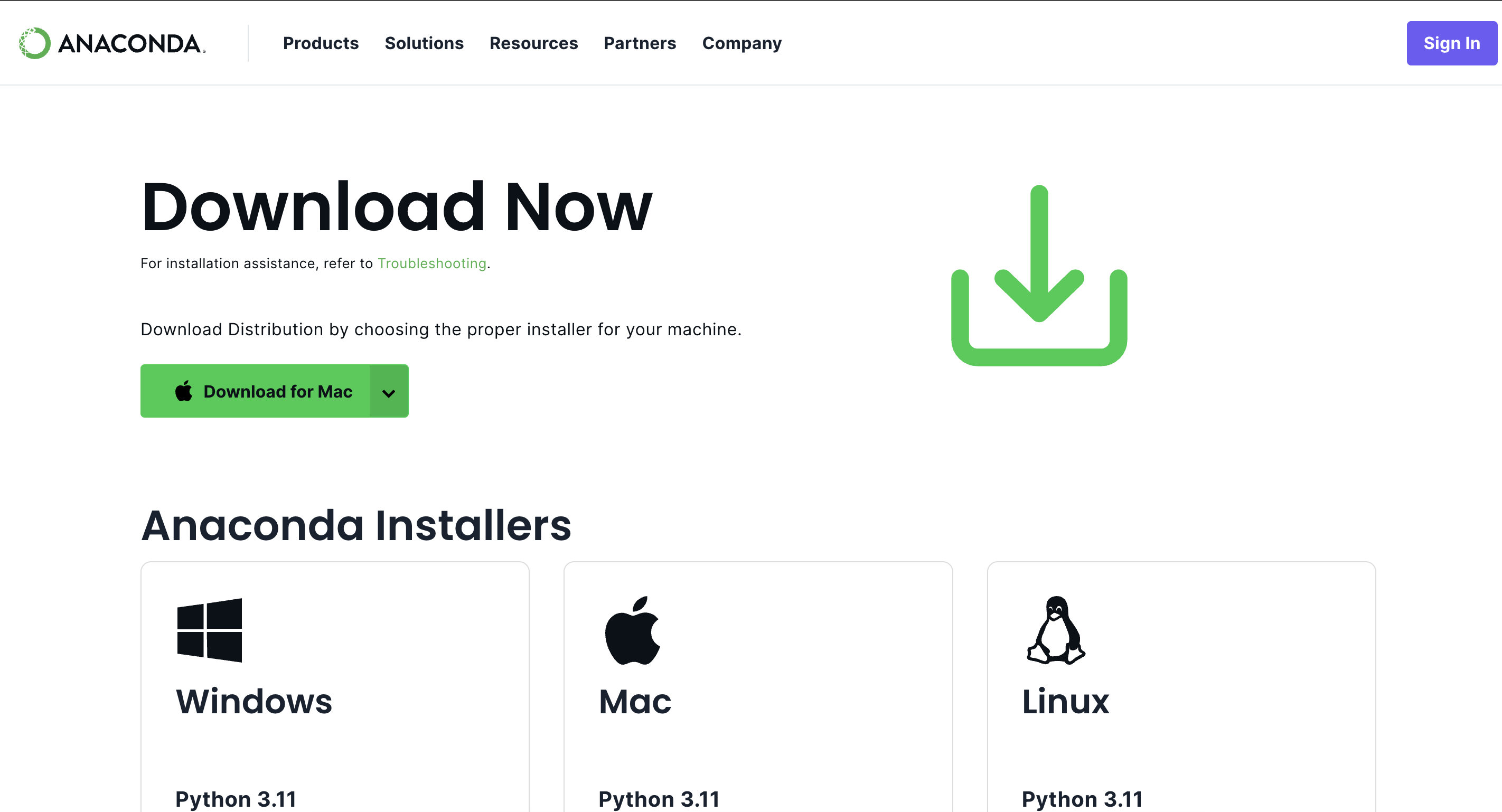


Рисунок 1 – Установка Anaconda

## Подгрузим нужные нам библиотеки (рисунок 2)

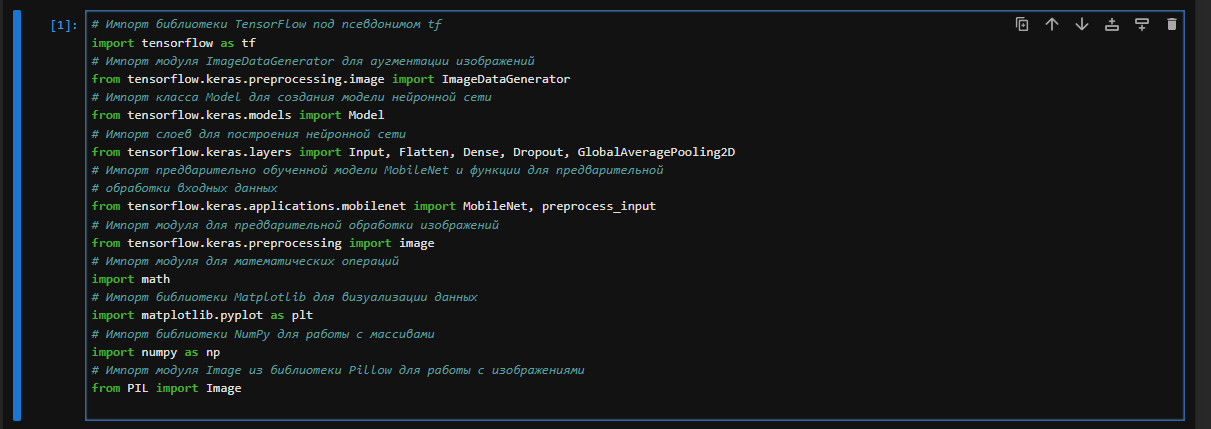


Рисунок 2 – Библиотеки

## Скачаем файлы для обучения нашей нейросети (рисунок 3)

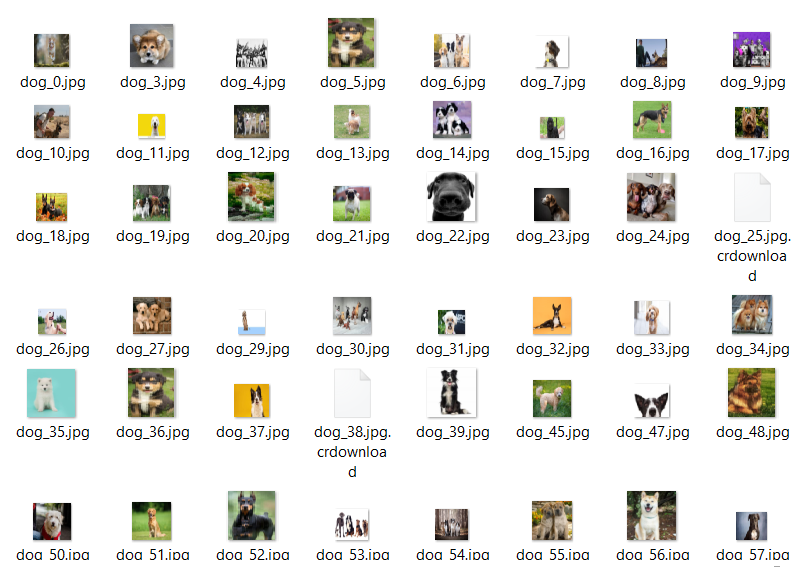


Рисунок 3 – Файлы для обучения

## Подключаемся к серверу Jupiter

В приложении Anaconda-Navigator нажимаем Jupiter notebook после чего появляется терминал, из которого мы копируем адрес сервера   
(рисунок 4).

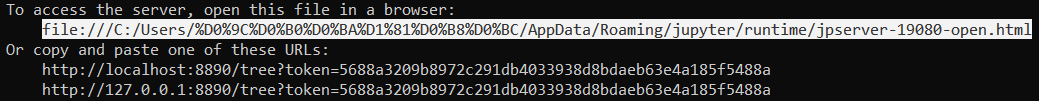


Рисунок 4 – Адрес сервера

## Эти константы (рисунок 5) служат для установления параметров и структуры обучения нейронной сети в процессе машинного обучения, особенно при работе с изображениями

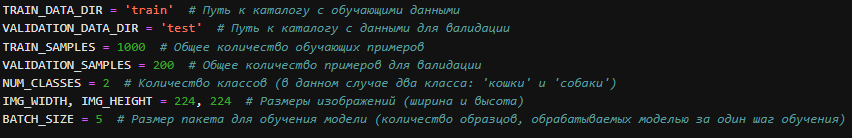


Рисунок 5 – Константы

## Создаются два генератора данных. Для обучения нейросети недостаточно пары сотен фотографий.

Поэтому будет использоваться такой прием как аргументация – это техника, используемая в машинном обучении, особенно при работе с изображениями, для искусственного увеличения объема обучающих данных путем создания модифицированных версий уже существующих данных. Это особенно важно в ситуациях, когда количество доступных данных ограничено, чтобы обеспечить достаточное разнообразие и предотвратить переобучение модели (рисунок 6).

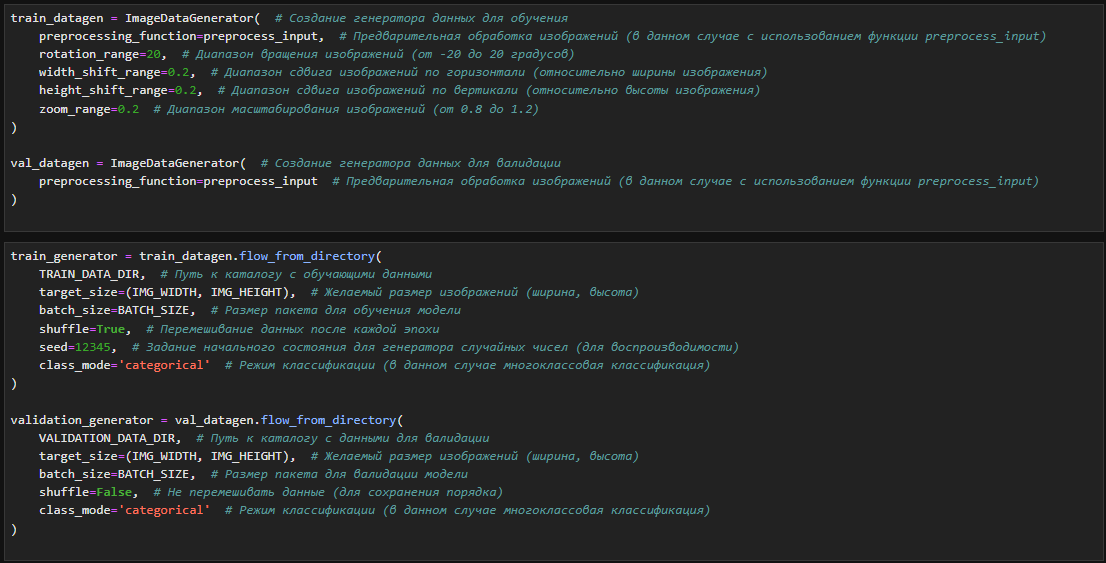


Рисунок 6 – Аргументация

## Создание модели на основе нейронной сети

В базовой модели нужно отключить обучение (для всех слоев устанавливается trainable = False). Входное изображение поступает на вход базовой модели. После базовой модели добавляется несколько своих слоев:слой пулинга GlobalAveragePooling2D; полносвязный слой из 16 нейронов Dense; используется механизм Дропаут (случайное отключение нейронных связей во время обучения); последний слой должен выдавать два значения (вероятности того, что на фото кошка или собака – два класса). Таким образом мы решаем задачу двухклассовой классификации (рисунок 7).

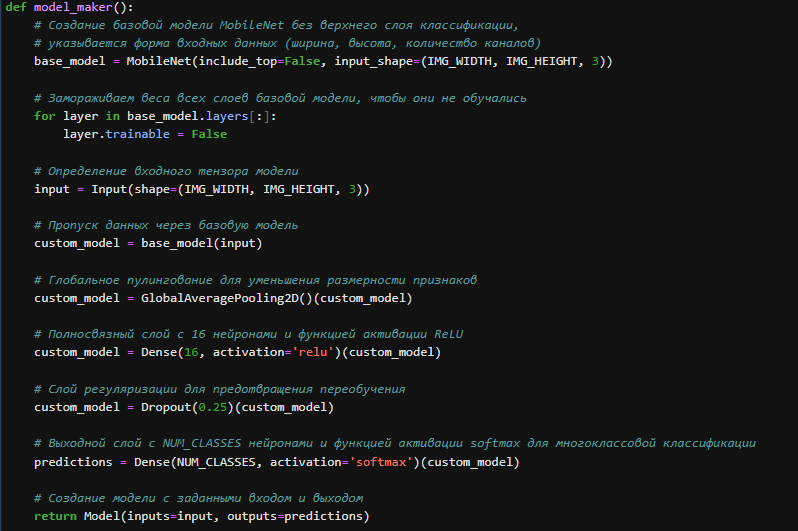


Рисунок 7 – Создание модели

## Компилируем и обучаем модель (рисунок 8)

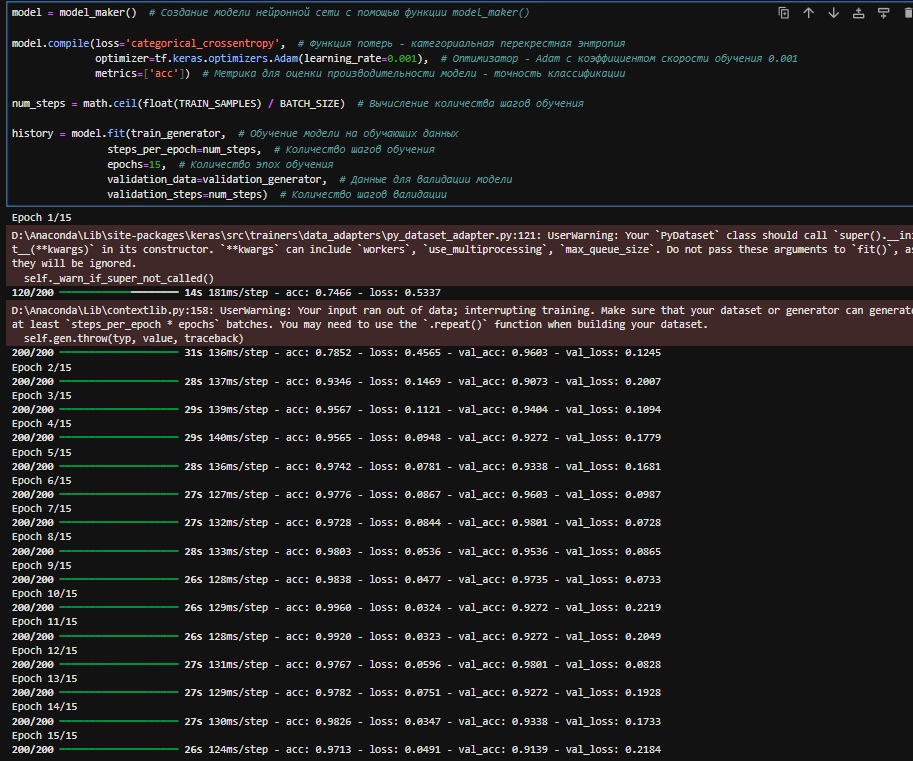
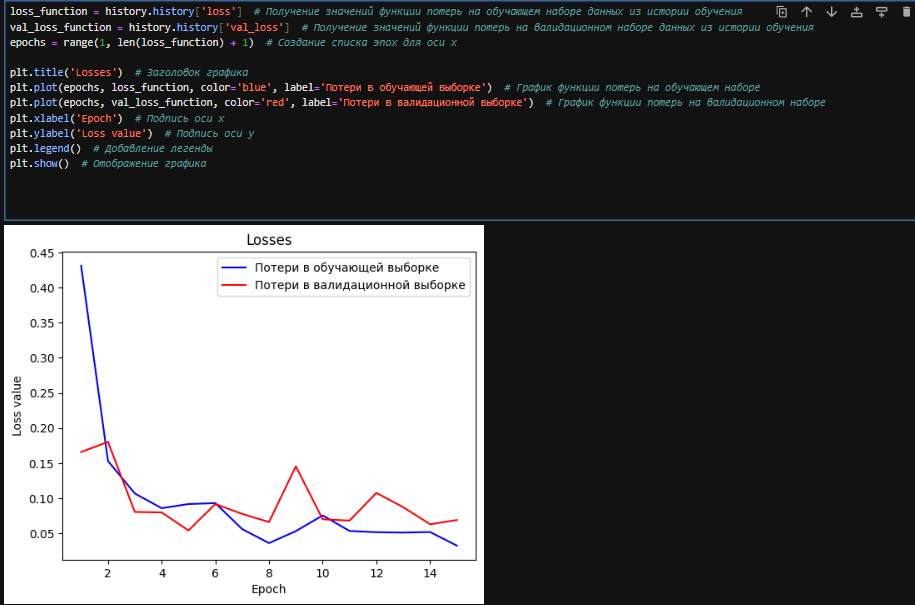


Рисунок 8 – Компиляция и обучение модели

## Построим график для визуализации обучения нейросети (рисунок 9)

  
Рисунок 9 – График

## Визуализированная проверка изображений (рисунок 10)

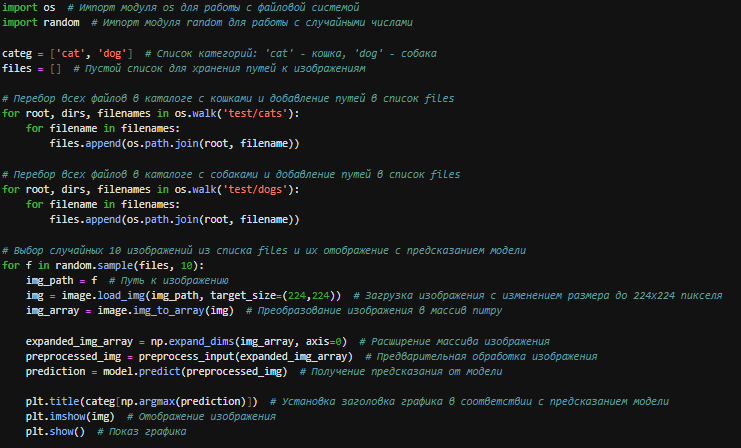


Рисунок 10 – Проверка изображений

## Проверка работы нейросети (рисунок 11-12)

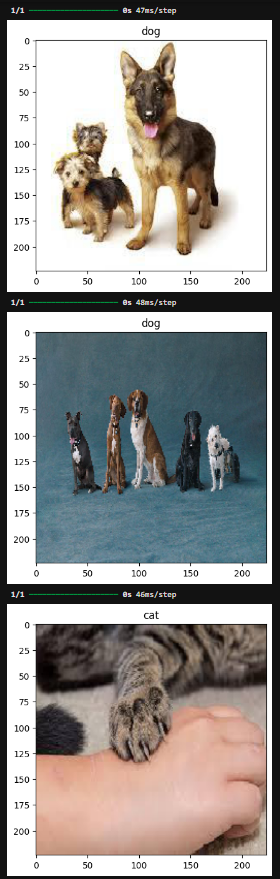


Рисунок 11 – Проверка изображений

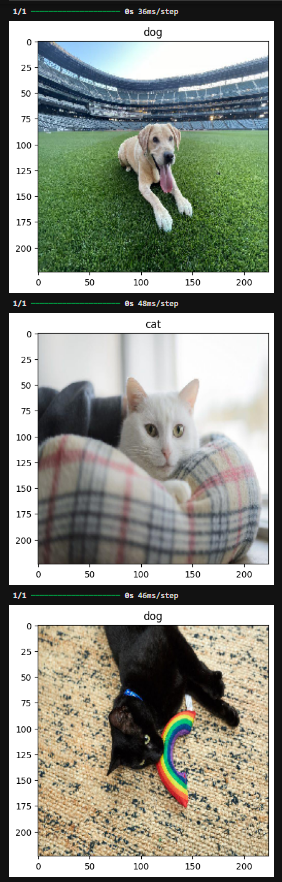


Рисунок 12 – Проверка изображений

## Попробуем поменять количество эпох обучения (рисунок 13) и полносвязный слой с 16 нейронами (рисунок 14)

После посмотрим, как изменился наш график (рисунок 15). Наша модель обучилась за 8 минут.

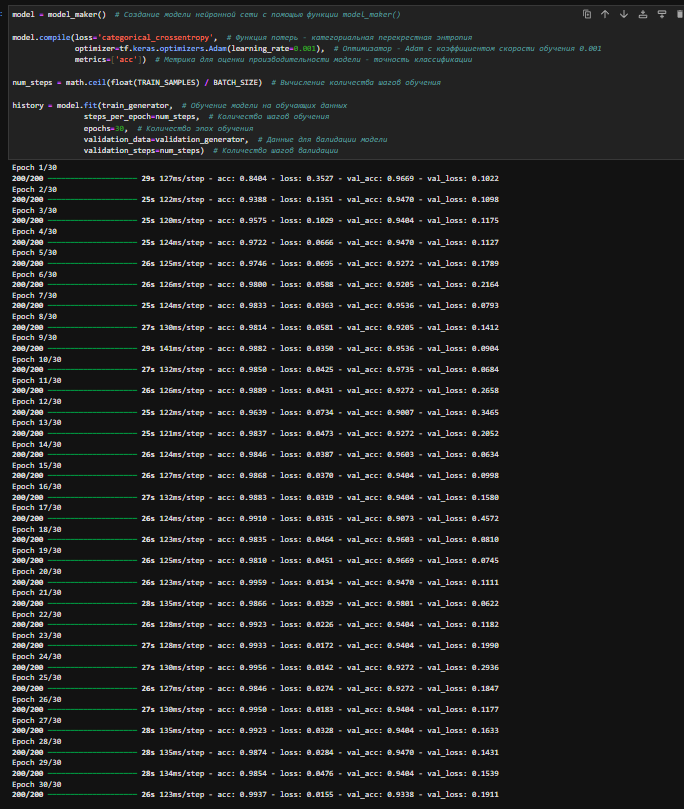


Рисунок 13 – Изменили количество эпох

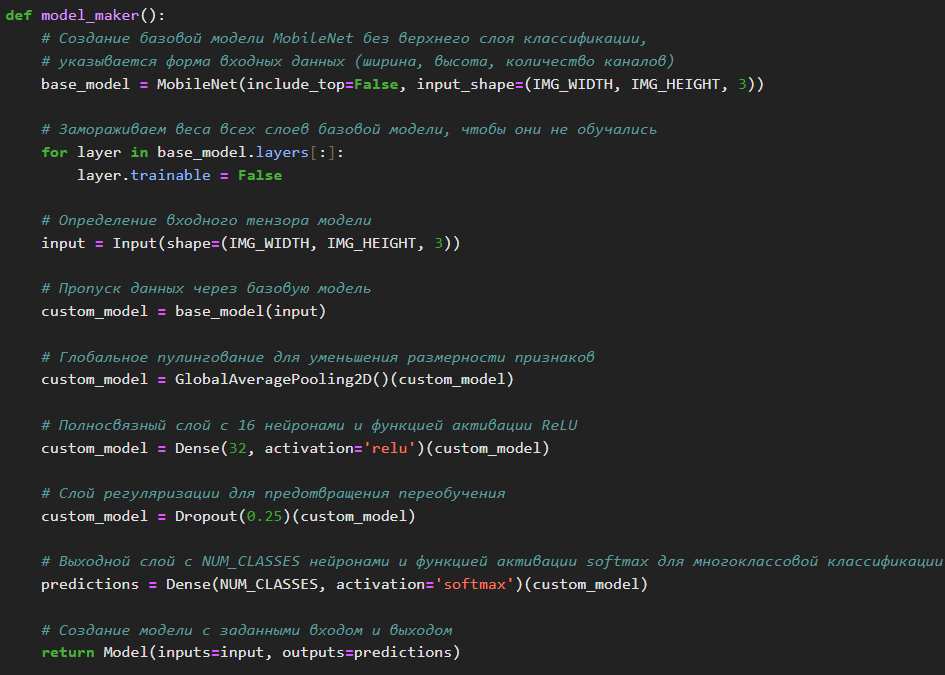


Рисунок 14 – Увеличение количества нейронов

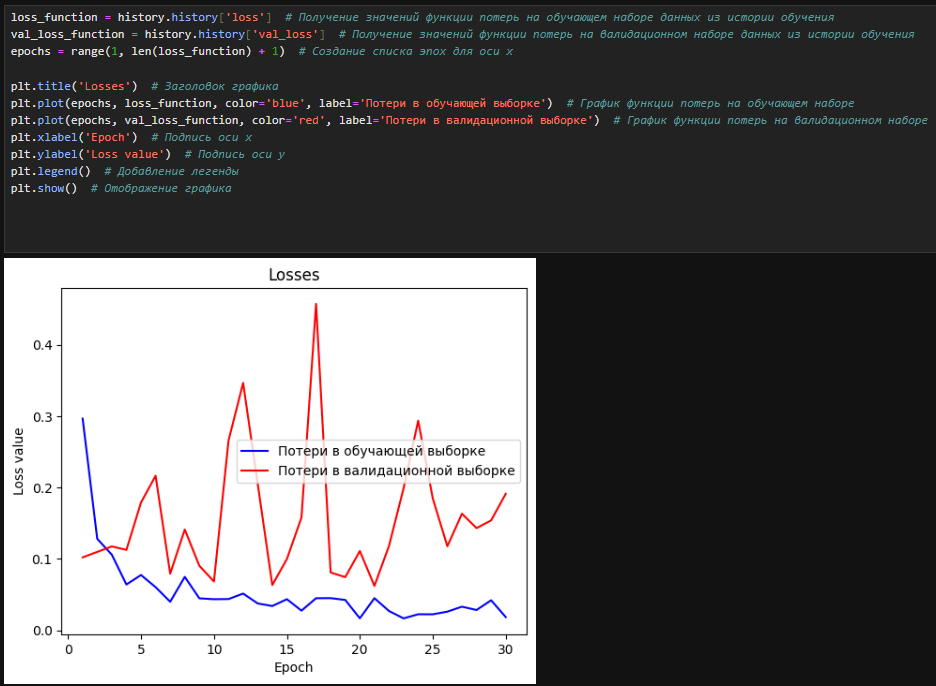


Рисунок 15 – Изменили dence

Несмотря на увеличенное количество эпох и нейронов, нейросеть всеравно допускает ошибки



Рисунок 16 – Неверный вывод

# Задания для защиты

## Какие есть еще готовые нейронные сети, которые умеют выделять осмысленно информацию, кроме MobileNet?

Помимо MobileNet существует нейронная сеть ResNet. Она позволяет создавать очень глубокие нейронные сети, с помощью использования остаточных блоков. В сравнении с MobileNet, ResNet точнее, за счет захвата более глубоких признаков. Однако MobileNet потребляет значительно меньше ресурсов и может быть эффективно использована на более слабых устройствах. Также MobileNet в основном используется в задачах реального времени, в то время как ResNet чаще используется в академических проектах, которые требуют более глубоких моделей.

## Как влияет количество эпох обучения на результат нейронной сети MobileNet?

Увеличение количества эпох может привести к улучшению качества модели. Большее количество эпох позволяет модели лучше адаптироваться к обучающим данным и выявлять более сложные закономерности в данных.

Однако слишком большое количество эпох может привести к переобучению модели. Переобучение происходит, когда модель начинает запоминать обучающие данные вместо того, чтобы обобщать общие закономерности. Это может привести к плохой обобщающей способности модели на новых данных.

## Отсортируем собственные фото с помощью нейронной сети

Для этого мы нашли по 20 фотографий людей разных расс. Поставим нашей модели обучение на 30 эпох. Посмотрим на результат (рисунки 17-21).

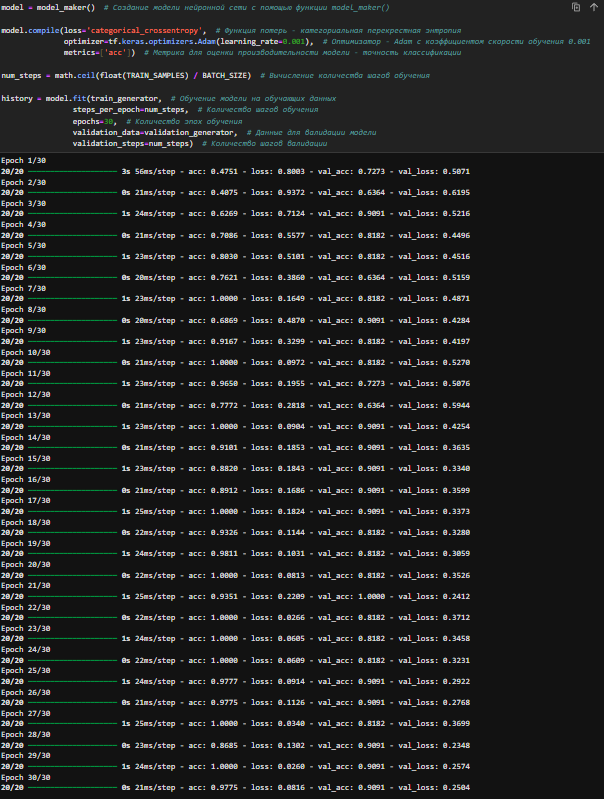


Рисунок 17 – Обучение

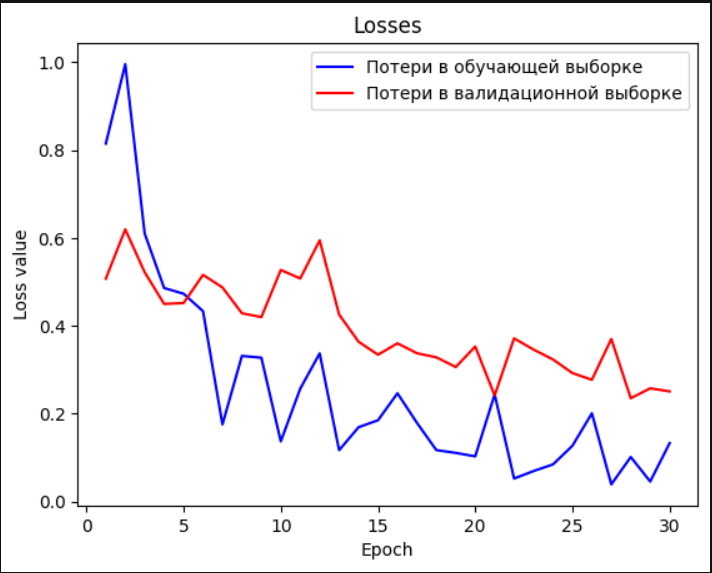


Рисунок 18 – График

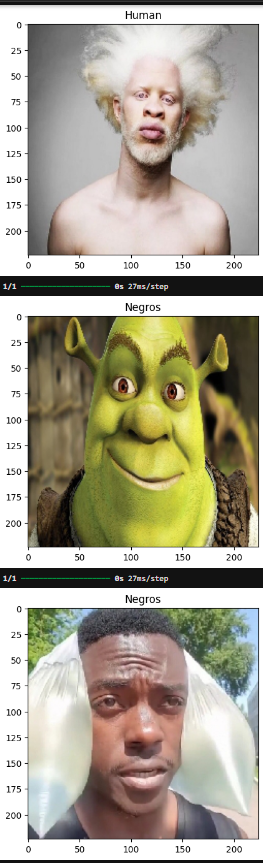


Рисунок 19 – Проверка вывода

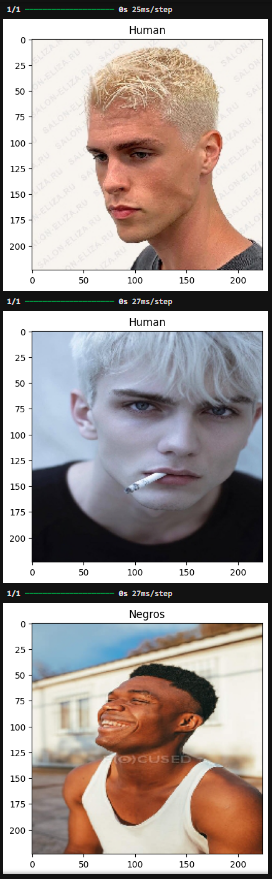


Рисунок 20 – Проверка вывода