|  |  |
| --- | --- |
| 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                            Т.А Абуталипов  «  17  »        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

## Скачиваем Anaconda-navigator

## Подгрузим нужные нам библиотеки

## Чтобы началась подгрузка библиотек следует воспользоваться командой «pip install tensorflow» (рисунок 1)

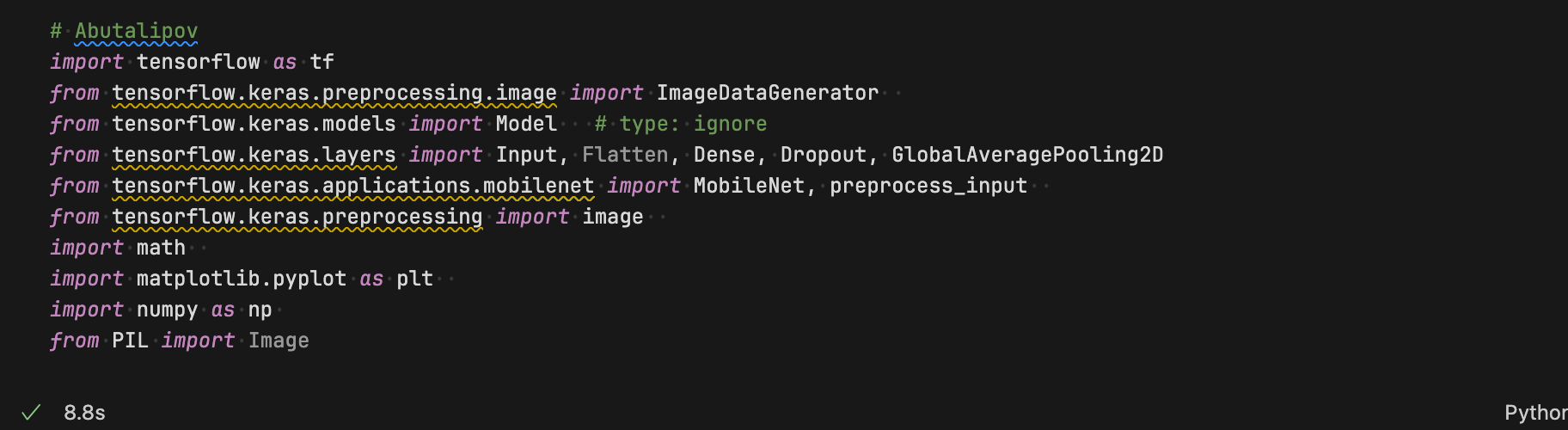


Рисунок 1 – Библиотеки

## Успешное скачивание файлов для обучение нейросети (рисунок 2)

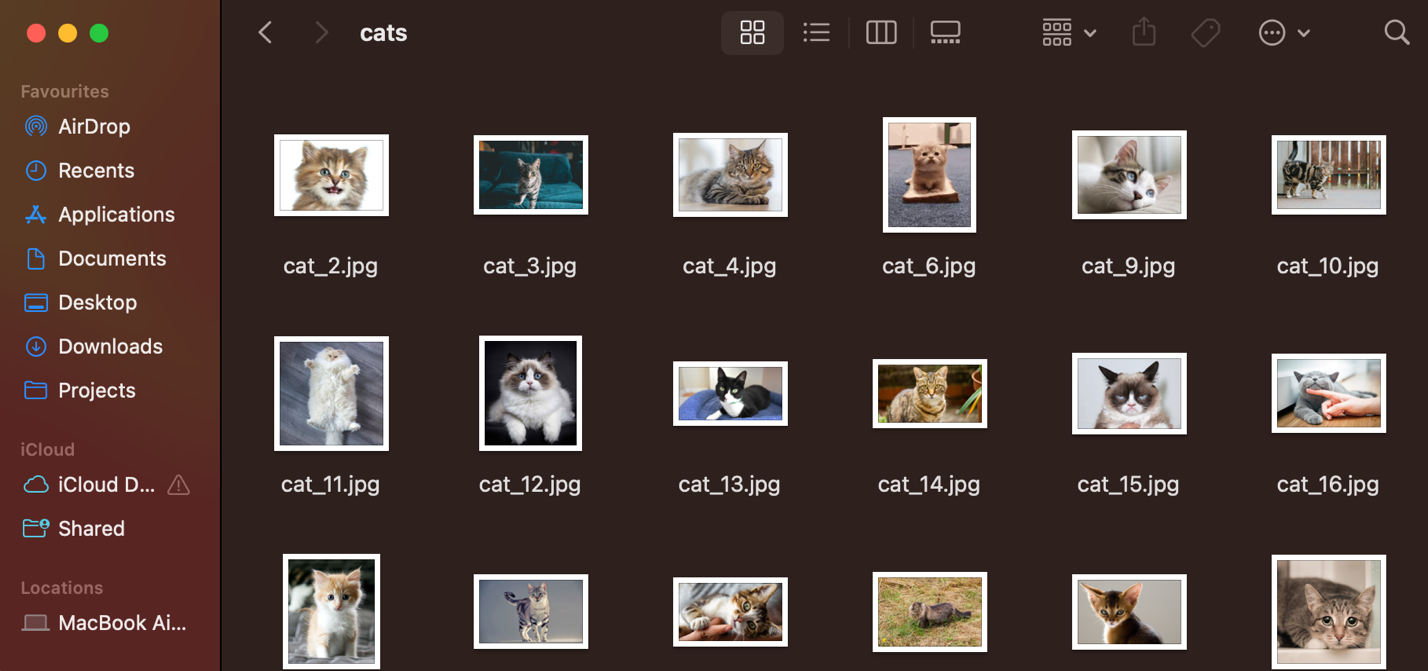


Рисунок 2 – Файлы для обучения

## Подключаемся к серверу Yupiter

В приложении Anaconda-Navigator открываем Jupiter notebook после чего копируем адрес сервера. В VSCode нажимаем выбрать ядро, выбираем python3 и вставляем скопированные данные (рисунки 3-4).

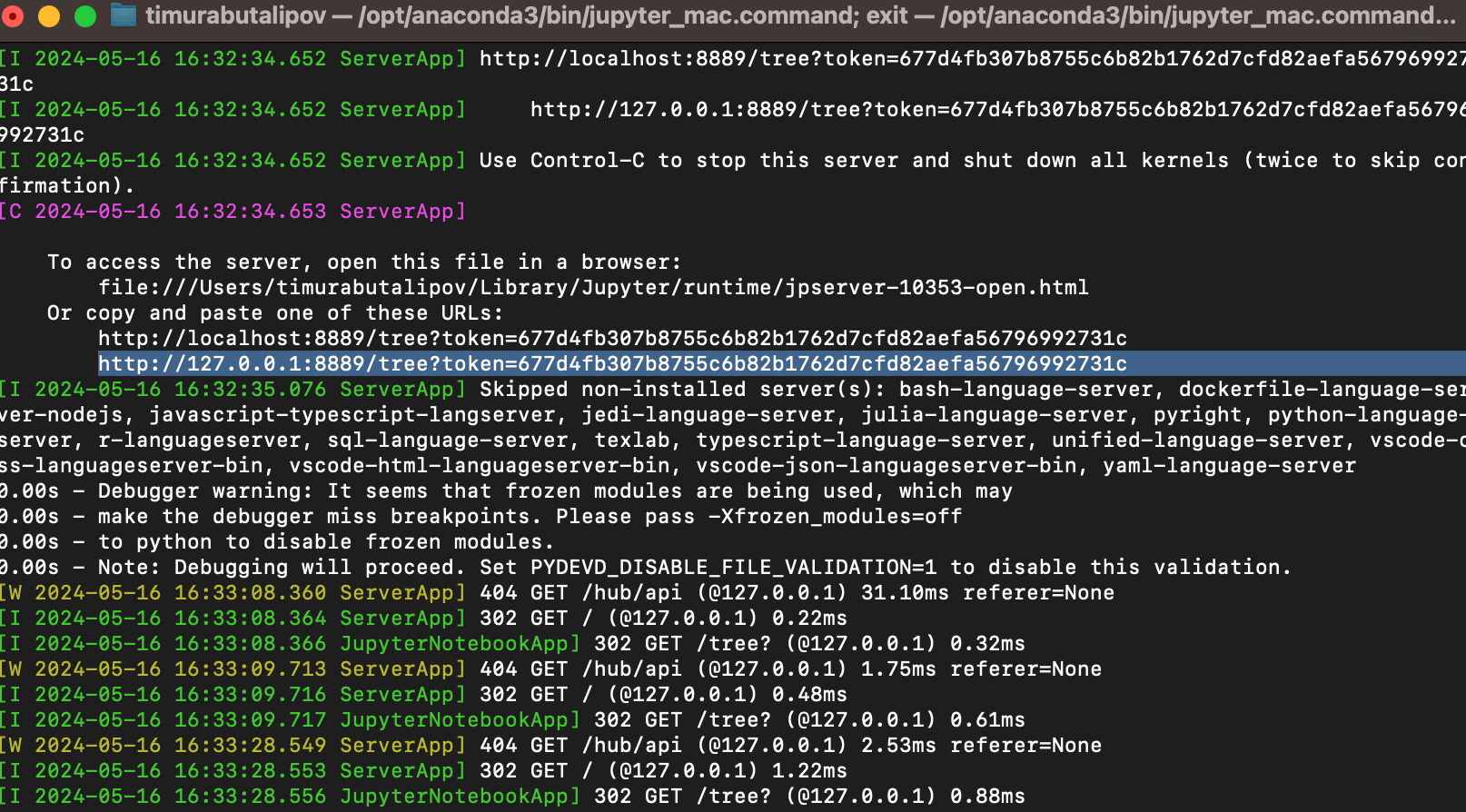


Рисунок 3 – Адрес сервера

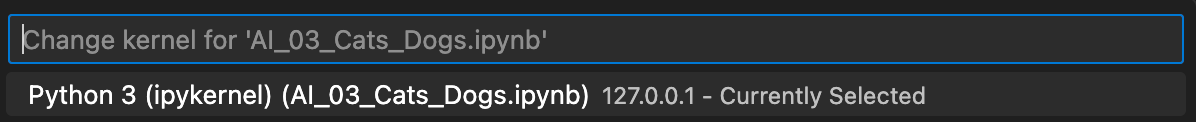


Рисунок 4 – Python3

## Эти константы служат для установления параметров и структуры обучения нейронной сети (рисунок 5)

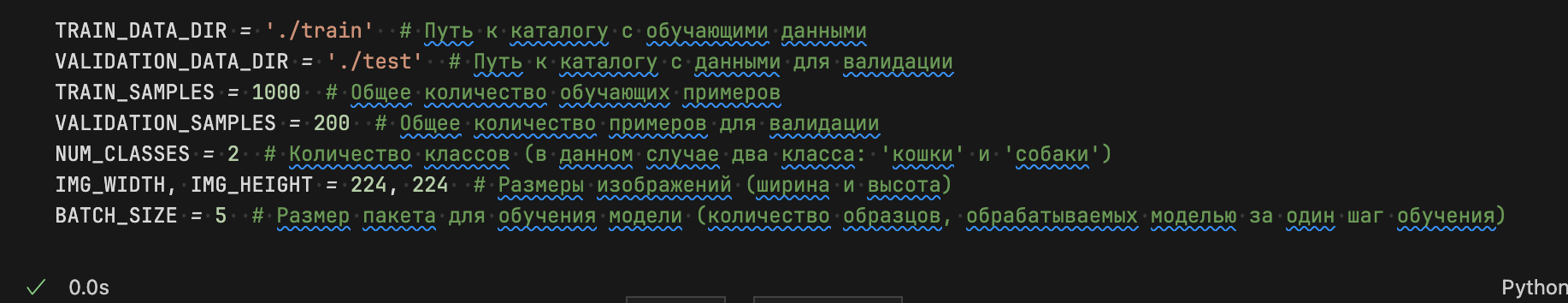


Рисунок 5 – Константы

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

Поэтому будет использоваться такой прием как аргументация – это техника, которая используется в машинном обучении, особенно при работе с изображениями, для искусственного увеличения объема обучающих данных путем создания модифицированных версий уже существующих   
данных (рисунок 6).

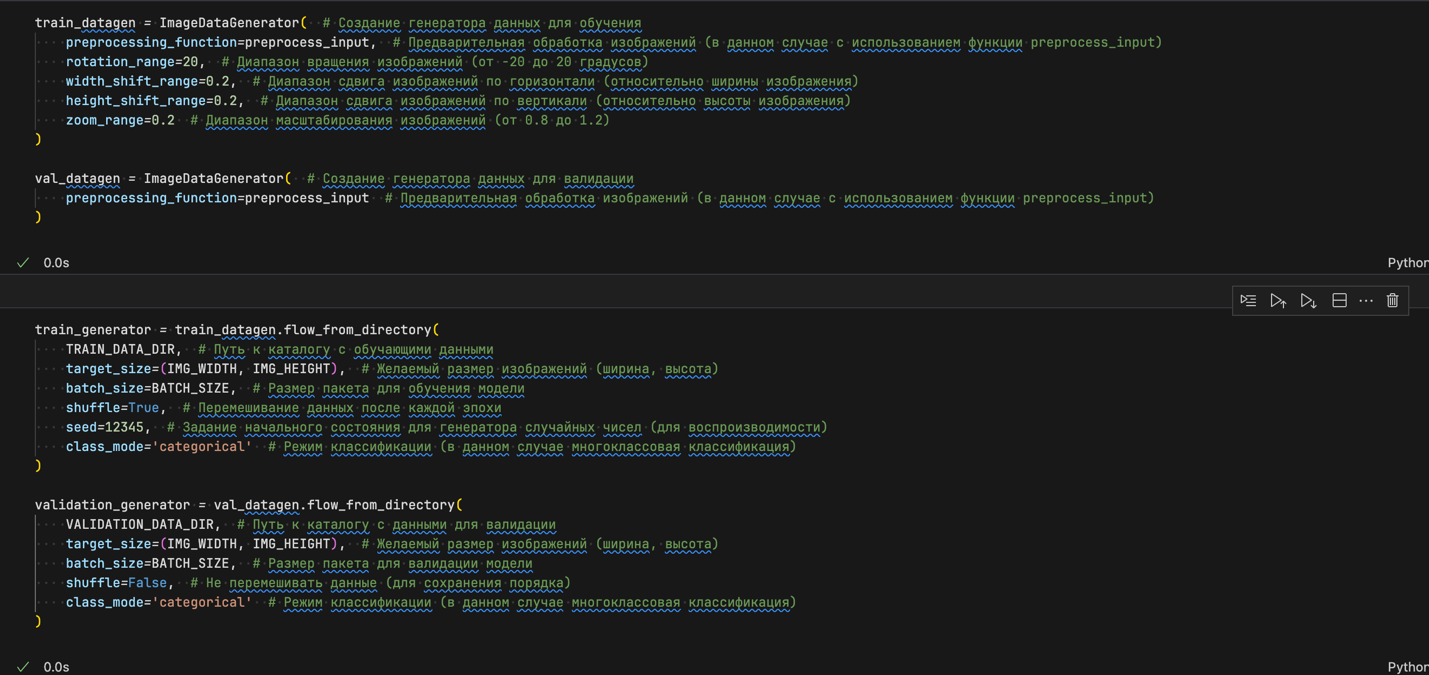


Рисунок 6 – Аргументация

## 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 24 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).

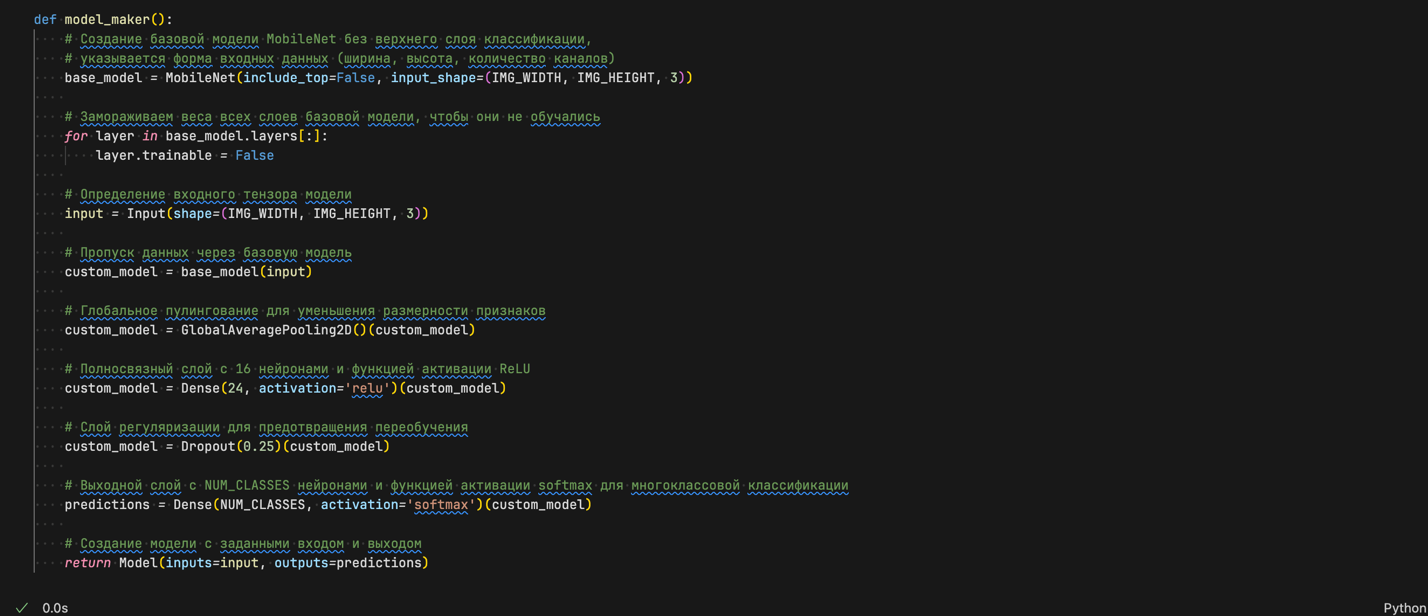


Рисунок 7 – Создание модели

## Компилируем и обучаем модель (рисунок 8).

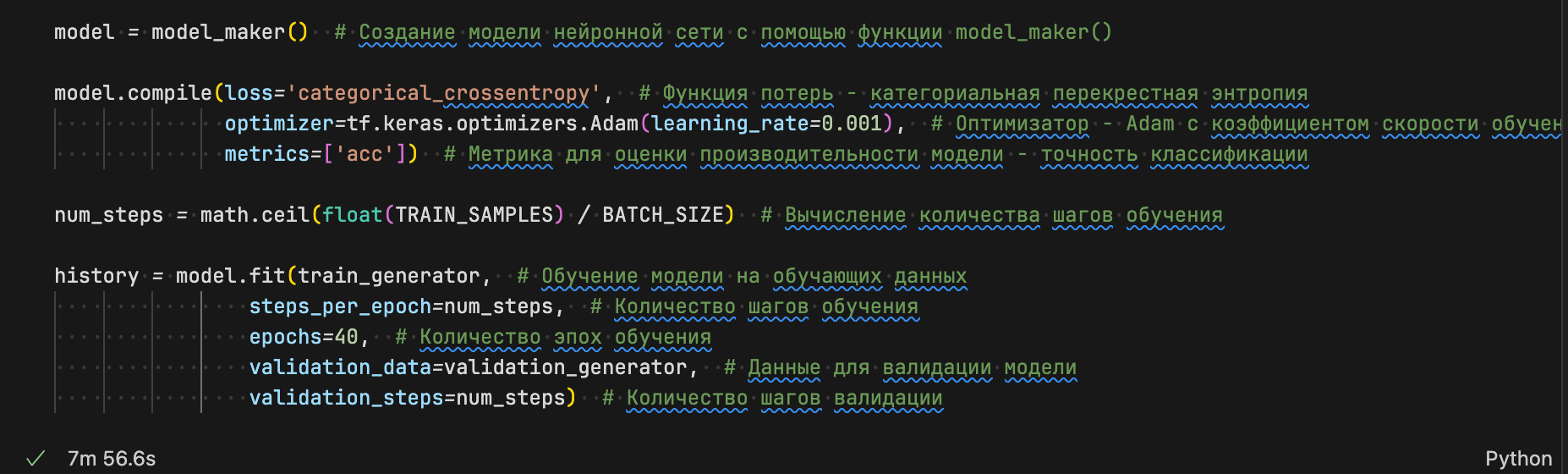
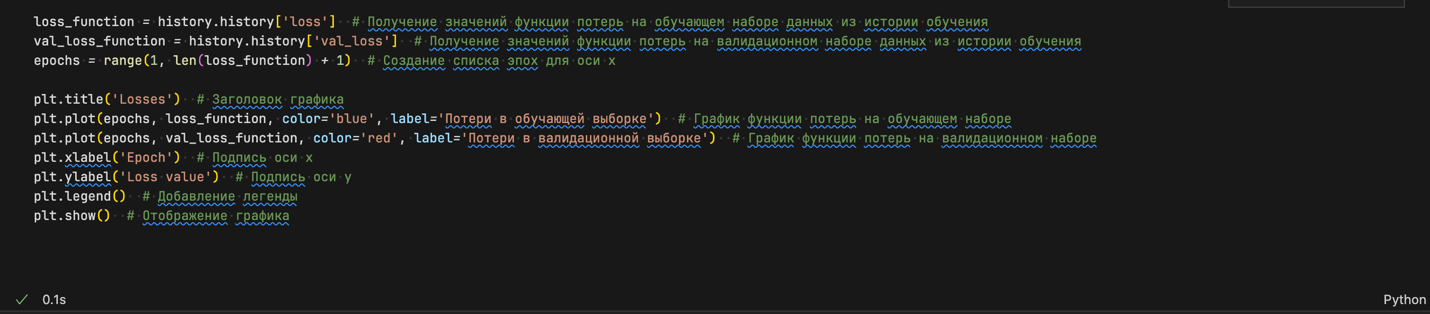


Рисунок 8 – Компиляция и обучение модели

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

  
Рисунок 9 – График

## Визуализированная проверка изображений (рисунок 10)

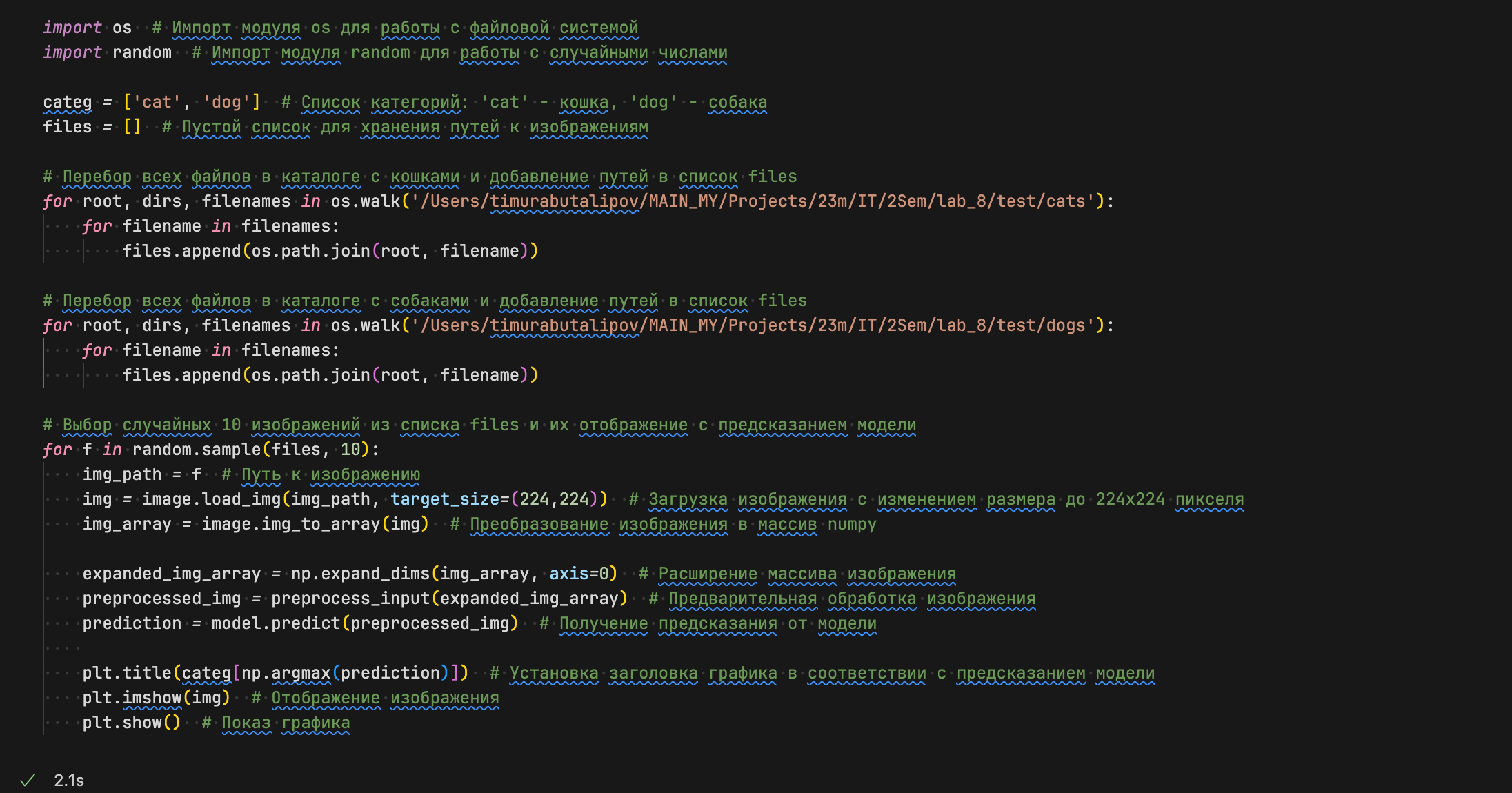


Рисунок 10 – Проверка изображений

## Запуск модели и тестирование функций (рисунки 11-13)

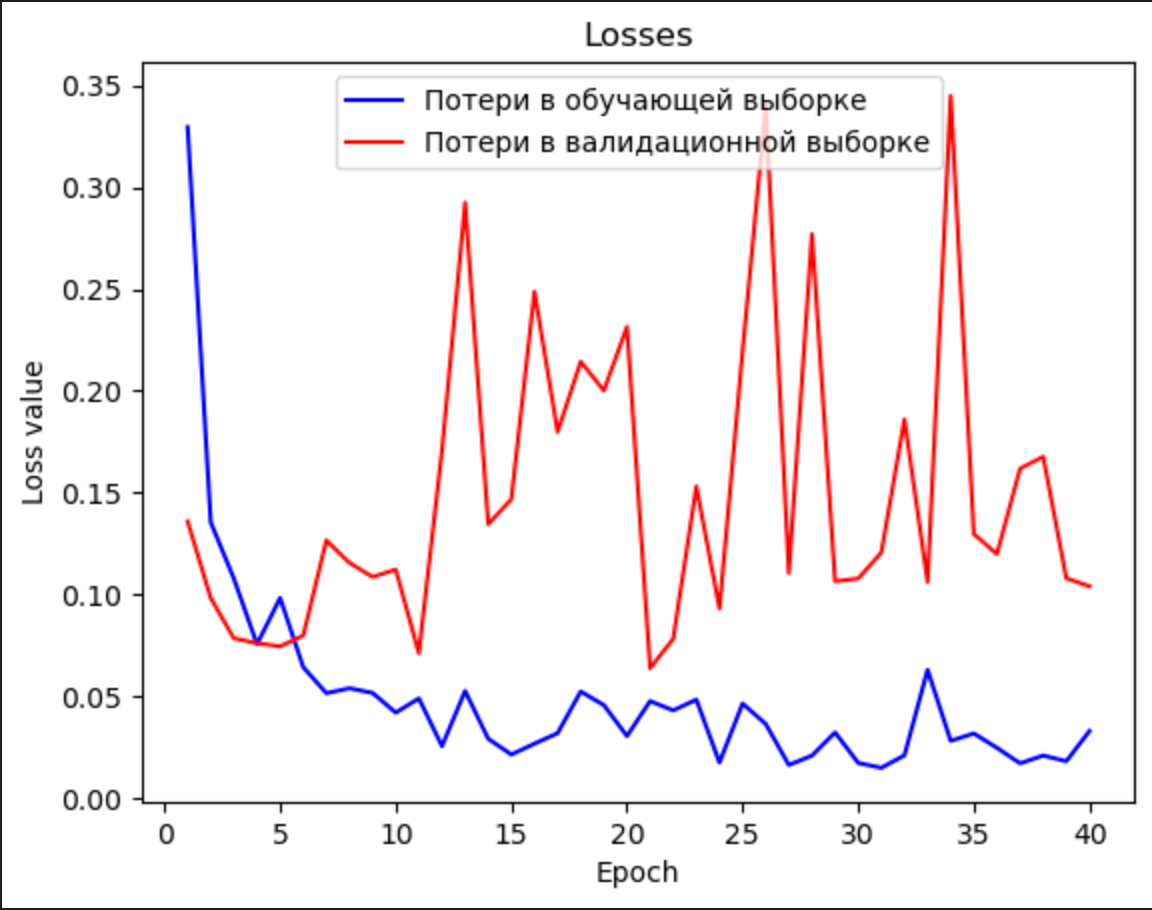


Рисунок 11 – График

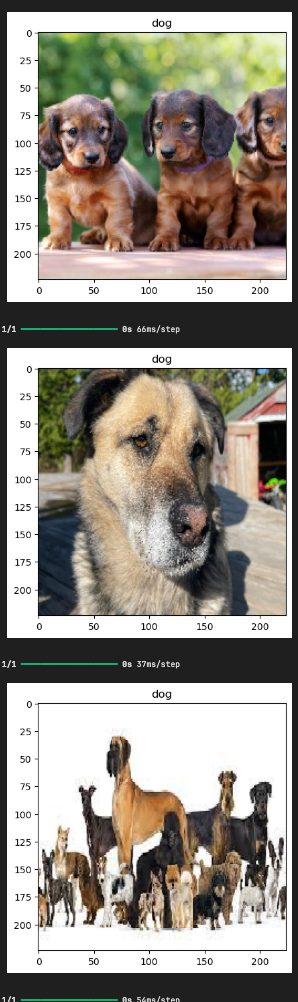


Рисунок 12 – Соответствие изображений



Рисунок 13 – Соответствие изображений

## Изменяем количество эпох и полносвязный слой с 24 нейронами и функцией активации ReLU (рисунок 15)

После посмотрим, как изменился наш график (рисунки 14-16).

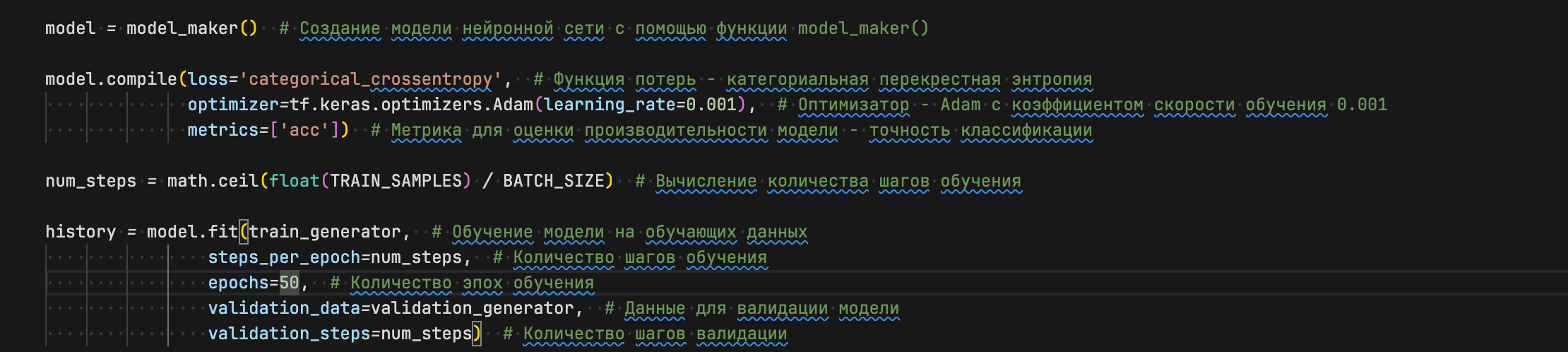


Рисунок 14 – Изменили количество эпох

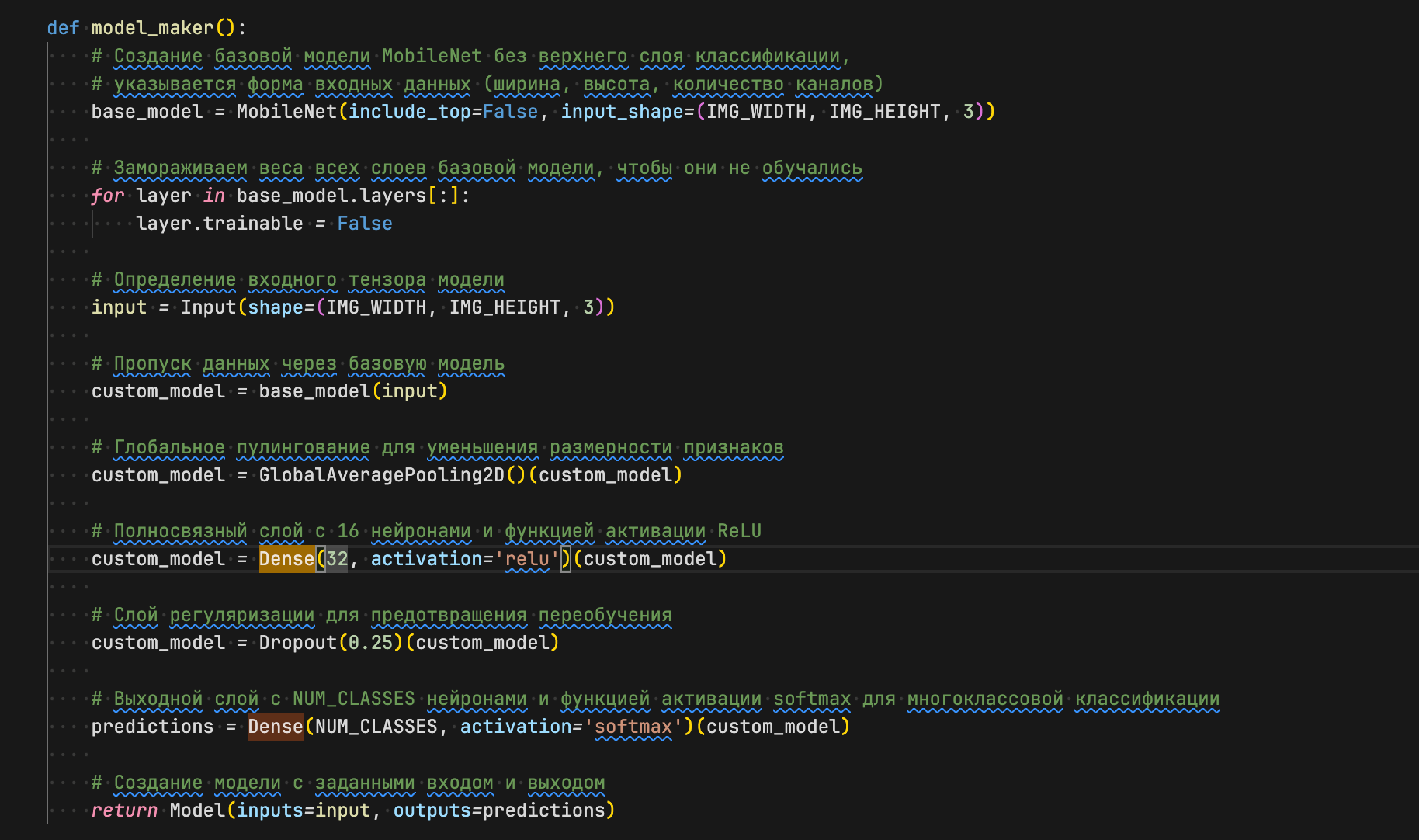


Рисунок 15 – изменили dence

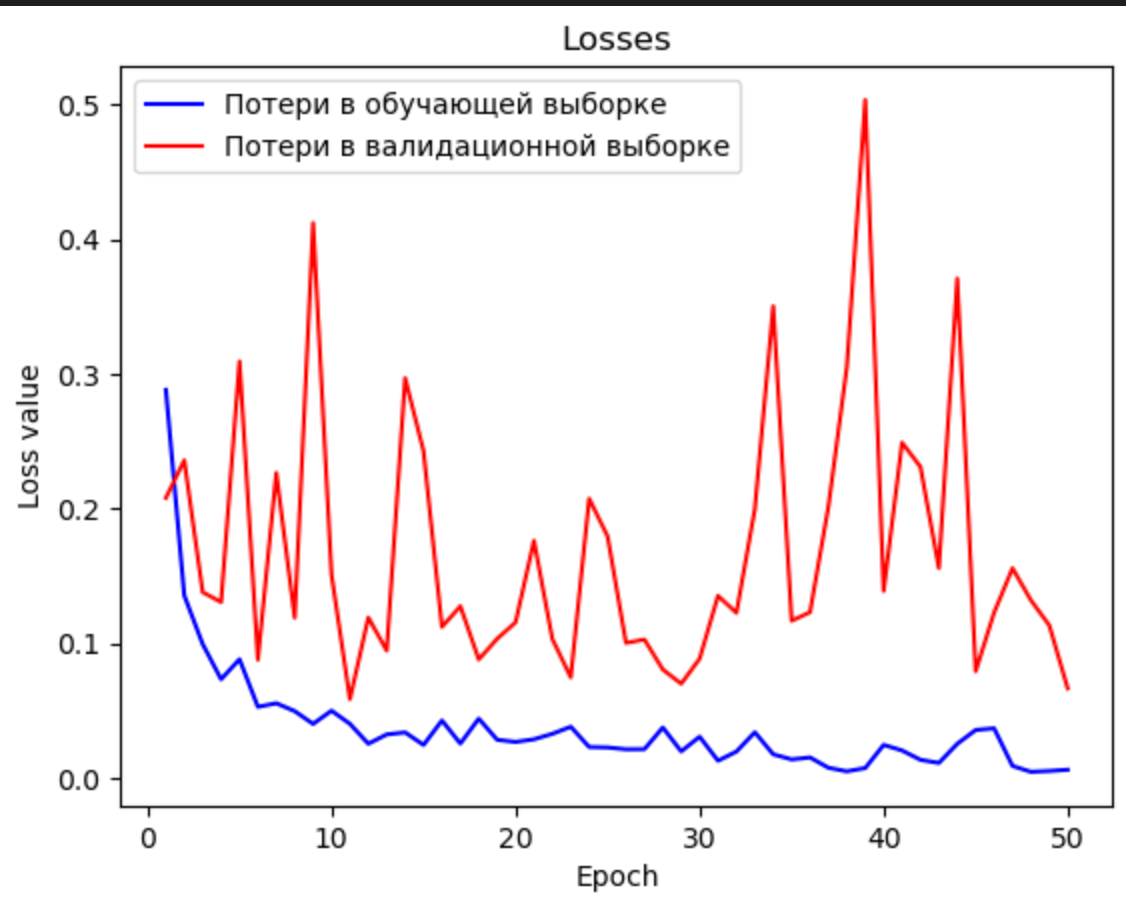


Рисунок 16 – Измененный график

# Задания для защиты

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

Нейронная сеть InceptionV3, нейронная сеть ResNet, нейронная сеть EfficientNet и нейронная сеть VGG

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

Количество эпох обучения влияет на результат нейронной сети MobileNet следующим образом:

- При недостаточном количестве эпох сеть может недообучиться и не сможет выделить все особенности изображений.

- При избыточном количестве эпох сеть может переобучиться на тренировочных данных, что приведет к плохому обобщению на новые данные.

## Отсортируем собственные фото с помощью нейронной сети

Для этого нам потребуется найти примерно по 100 фотографий разных машин, например VAZ и Ferrari. Посмотрим на результат (рисунок 17-19)

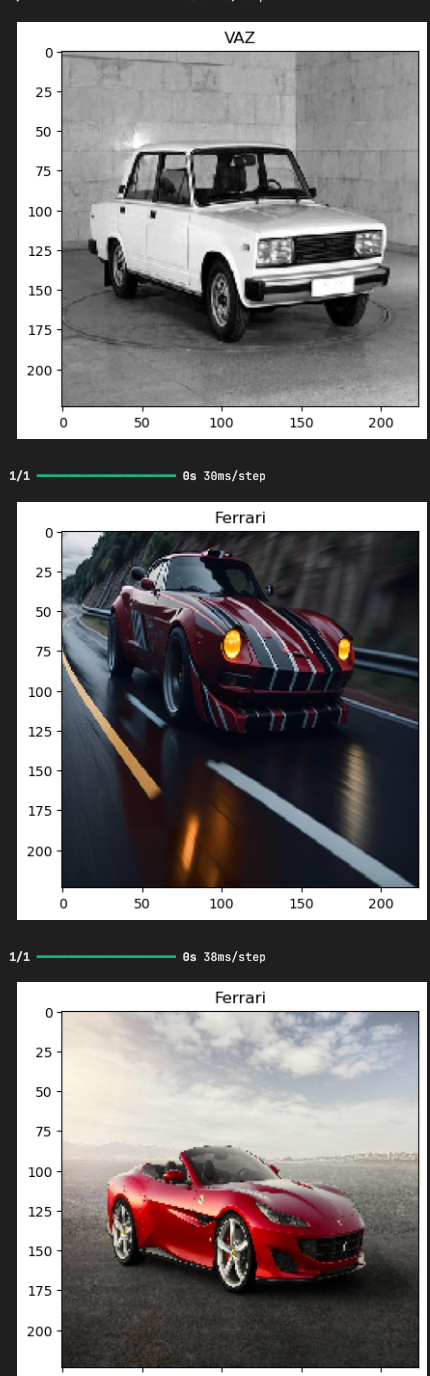


Рисунок 17 – Соответствие изображений

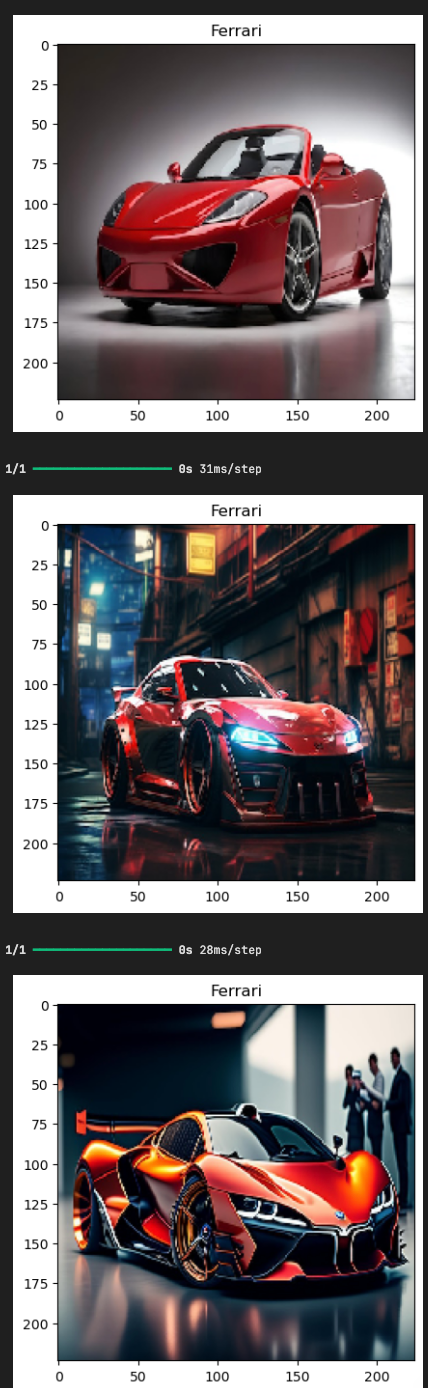


Рисунок 18 – Соответствие изображений

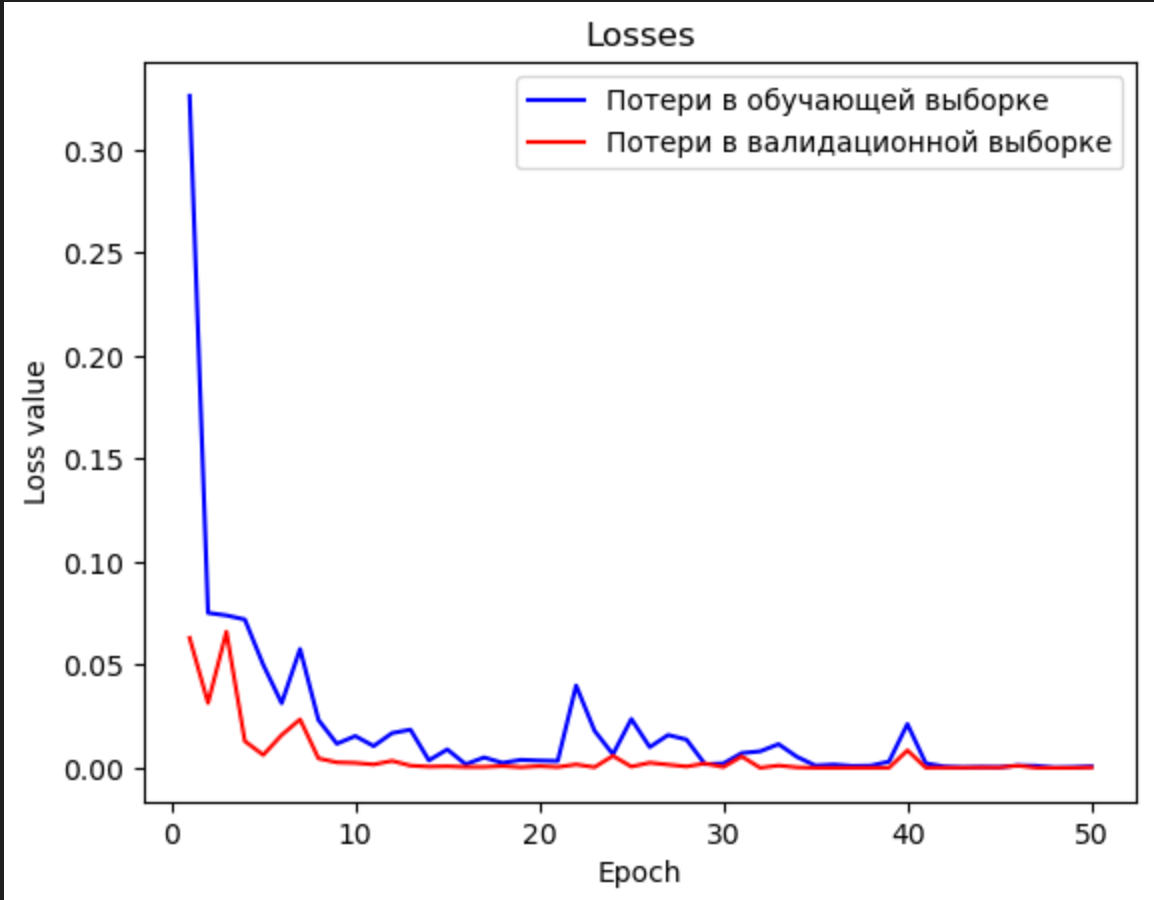


Рисунок 19 – График

Вывод

В ходе данной лабораторной работы мы познакомились с языком программирования Python, запускали нейросеть и работали в команде.