|  |  |
| --- | --- |
| Федеральное агентство железнодорожного транспорта  Омский государственный университет путей сообщения  Кафедра «Автоматика и системы управления»  ЗНАКОМСТВО С PYTHON  Лабораторная работа № 8  по дисциплине «Информационные технологии» | |
|  | Студентки гр. 23М                            И.А. Шевченко  С. О. Самойленко  А.В. Цвиль  Е.О. Толмашова  «    »        05        2024 г.  Руководитель –  старший преподаватель кафедры «АиСУ»                            Т.В. Васеева  «    »                  2024 г. |
| Омск 2024 | |

# Цель работы

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

# Выполнение работы

## Выбор среды

Рассмотрены различные среды разработки, такие как Jupyter Notebook, PyCharm, IDE Python и Google Colaboratory. Рекомендовано использование веб-приложения Jupyter Notebook, входящего в состав среды Anaconda, для удобства работы с визуализацией данных (рисунок 1).

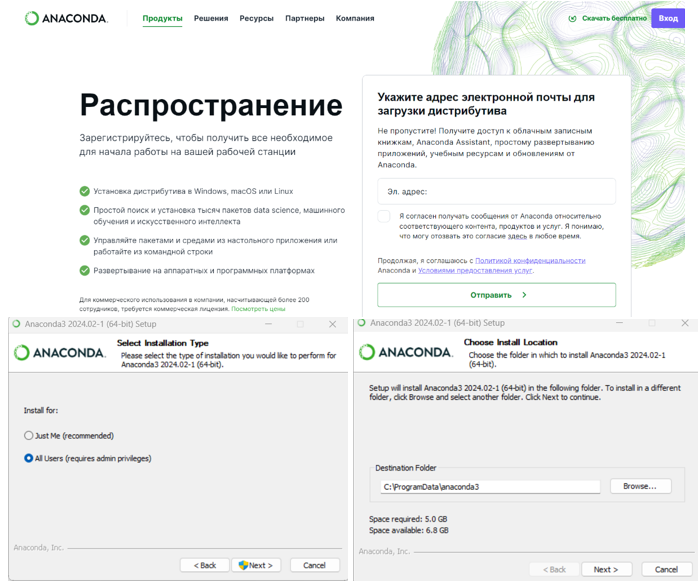


Рисунок 1 – Установка Anaconda

После устанавливаем Yupiter Notebook и копирует ссылку репозитория (рисунок 2).

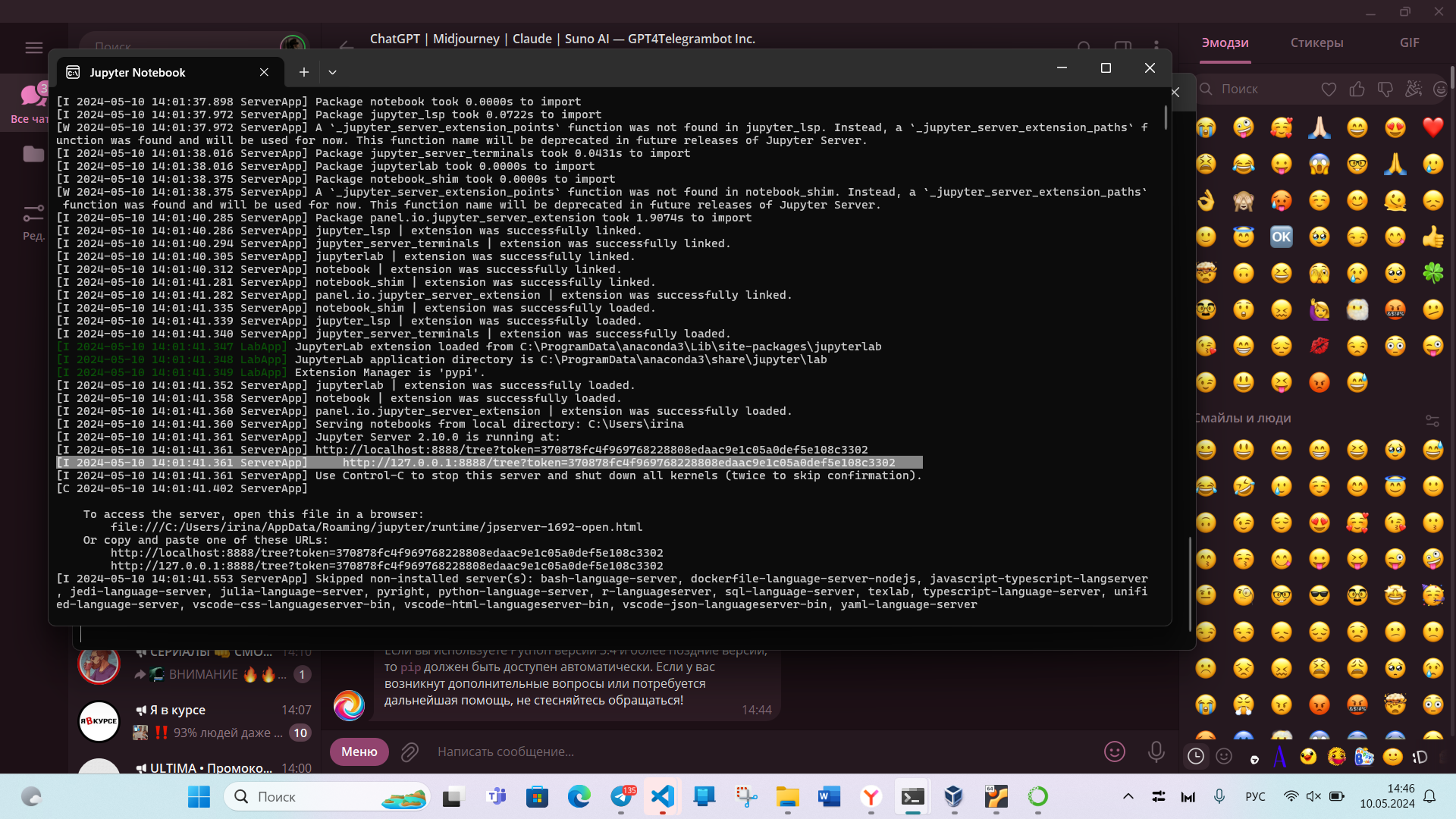


Рисунок 2 – Yupiter Notebook

Далее открываем файл в VS CODE и подгружаем нужные библиотеки (рисунок 3).

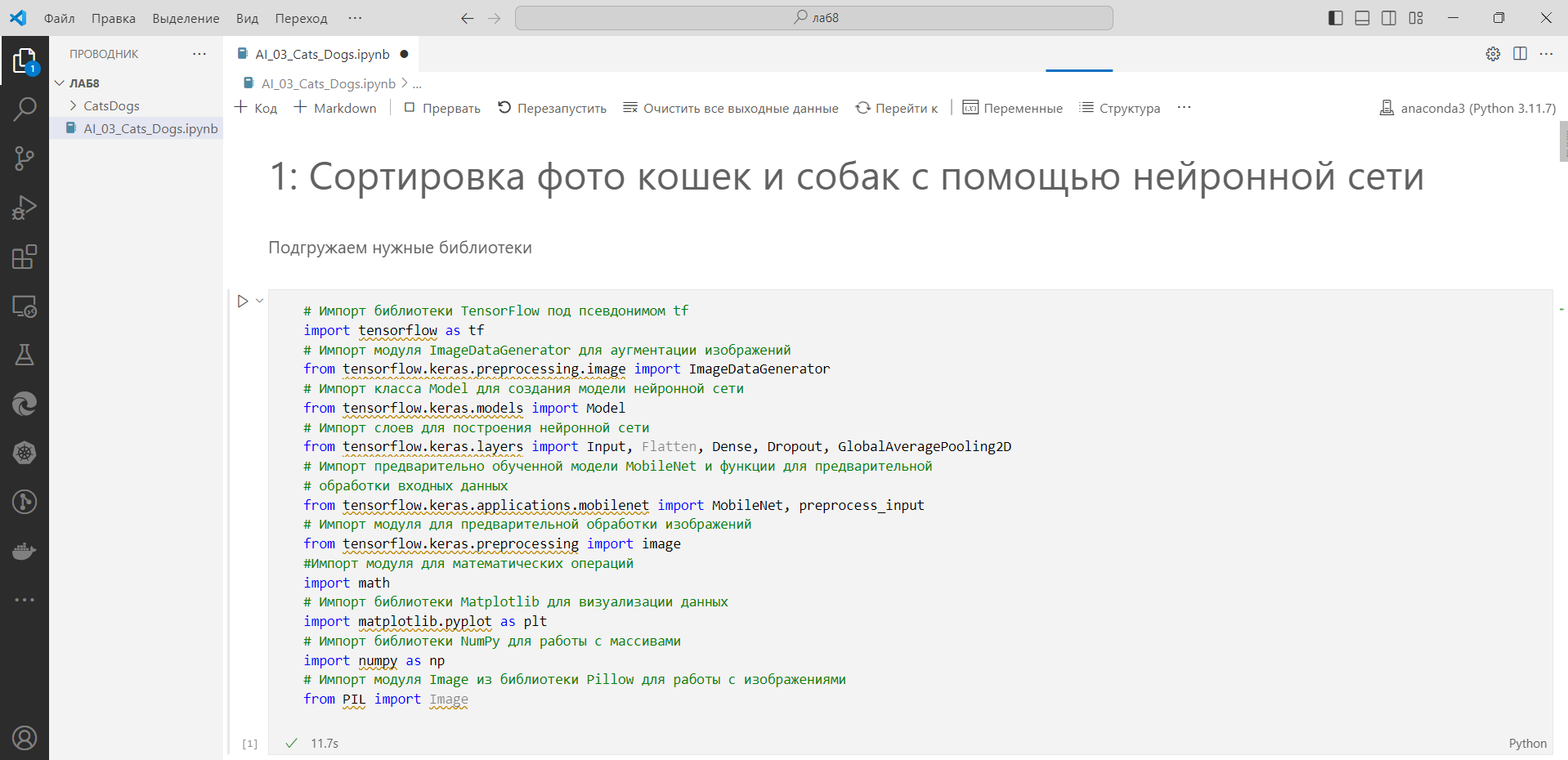


Рисунок 3 – Запуск программы

Скачиваем отсортированные по папкам файлы. В папке находятся train файлы для обучения нейросети, а в папке test файлы для проверки нейросети. Внутри этих папок есть папки cats и dogs, где собрано несколько десятков фотографий с кошками и собаками (рисунок 4).

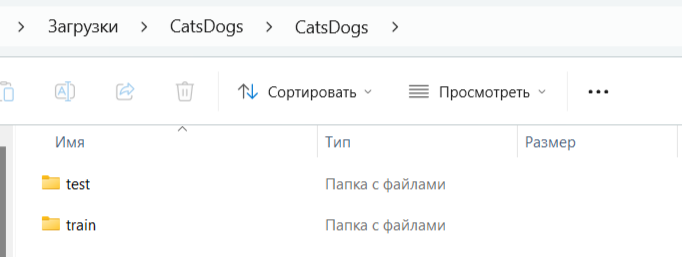


Рисунок 4 – Скачивание файла

Добавляем константы (рисунок 5).

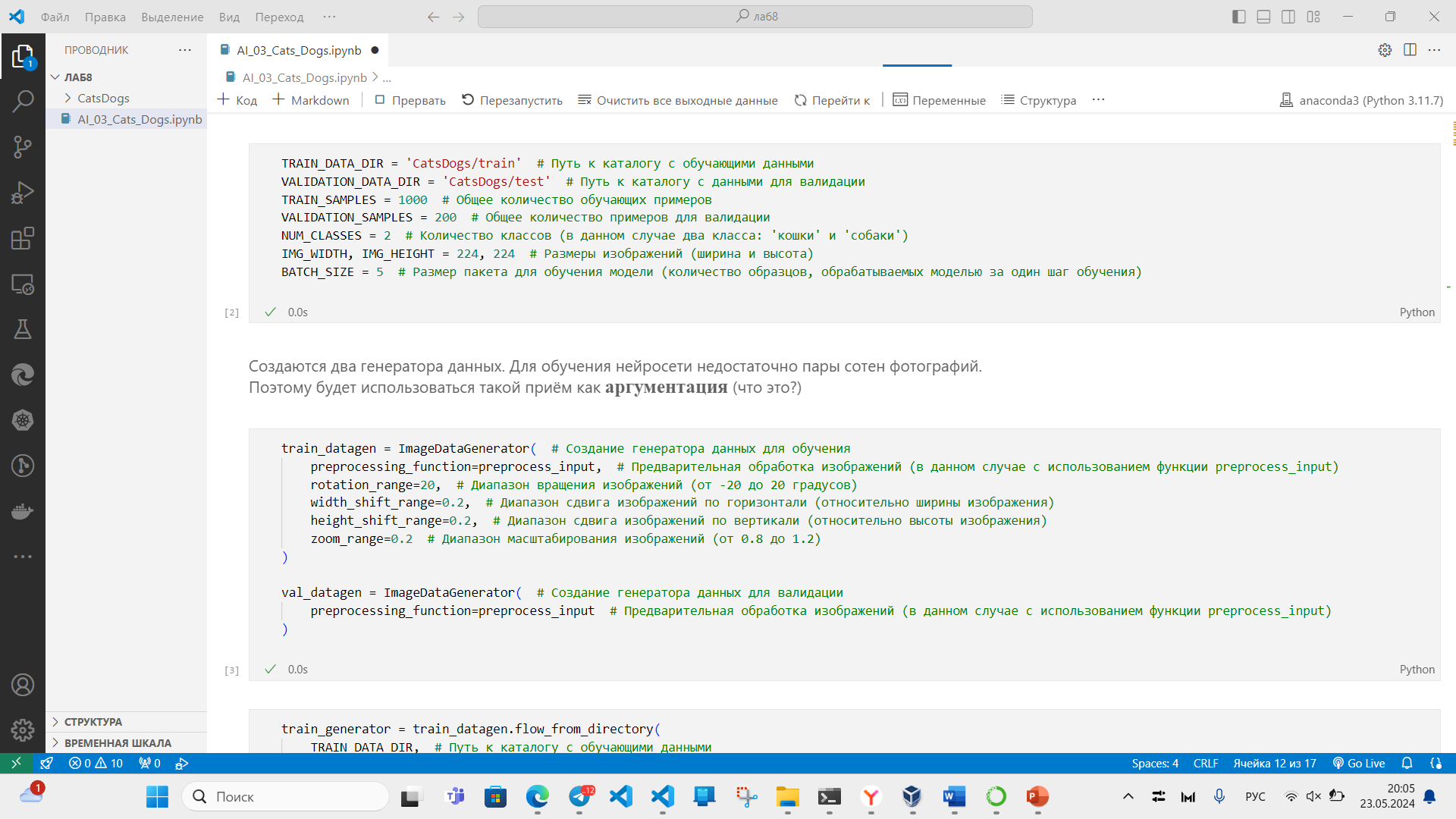


Рисунок 5 – Добавление констант

Эти константы используются для задания путей к каталогам с обучающими данными и данными для валидации, указания общего количества обучающих и вариационных примеров, определения количества классов указания размеров изображений (ширина и высота), а также определения размера пакета (batch size) для обучения модели. Константы помогают упростить настройку и использование модели машинного обучения, делая код более читаемым и понятным.

После создаются два генератора данных. Для обучения нейросети недостаточно пары сотен фотографий (рисунок 6-7).

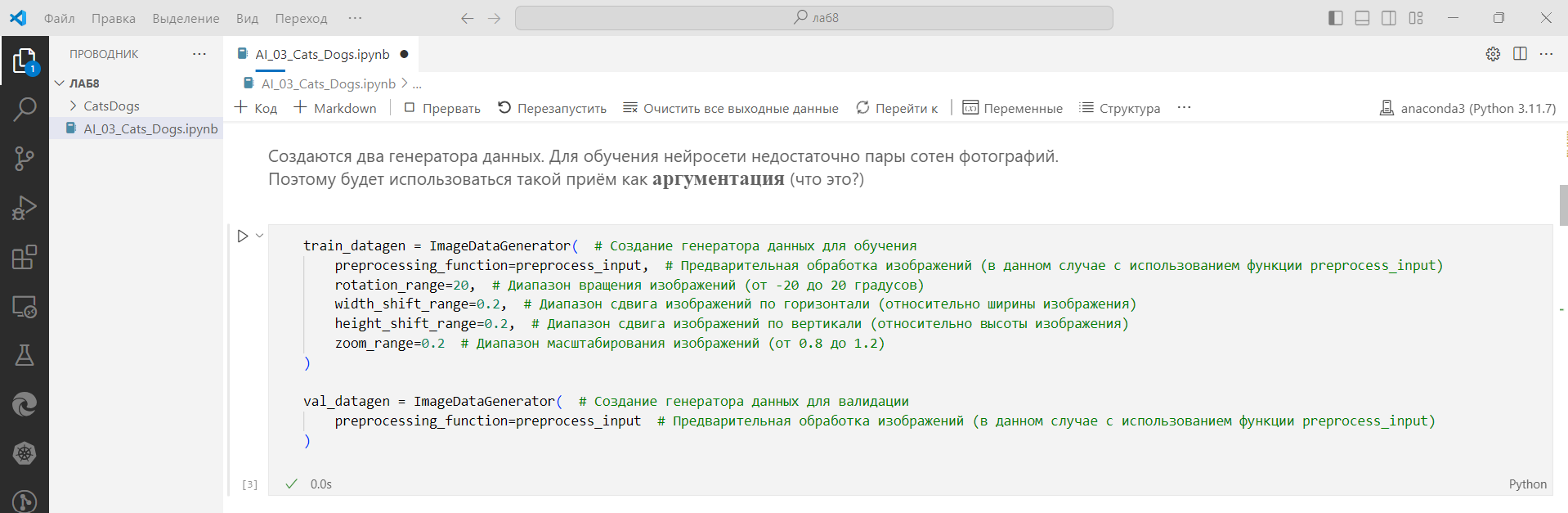


Рисунок 6 – Создание первого генератора данных

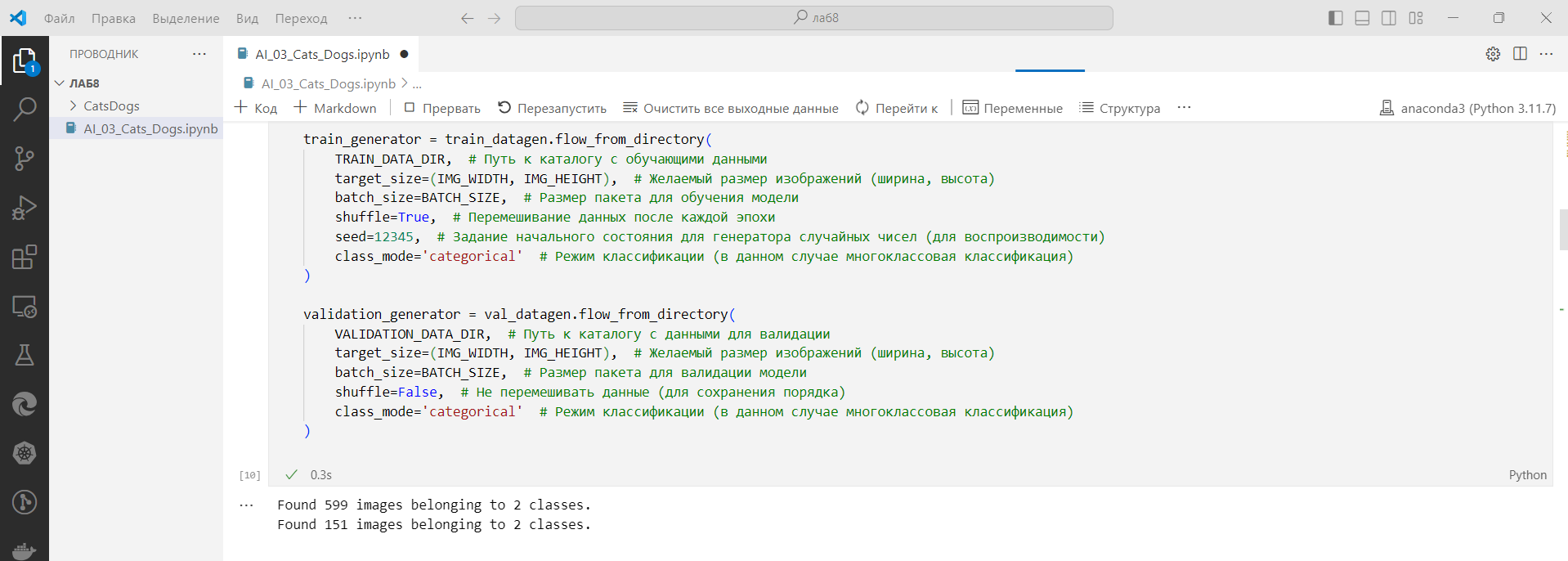


Рисунок 7 – Создание второго генератора данных

В данном приеме используется аргументация. Этот приём позволит нам увеличить количество изображений путём различных трансформаций. Позволяет обучить более мощную модель, которая будет лучше обобщать на новые данные. Предотвращает «запоминание» моделью только данных обучения, делая ее более устойчивой к вариациям в реальных данных. Добавляет новые вариации к исходным данным, делая модель более устойчивой к шуму, поворотам, изменению освещения и другим изменениям.

## Создание модели на основе нейронной сети, которая основана на готовой нейронной сети MobileNet

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 такие как 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 8).

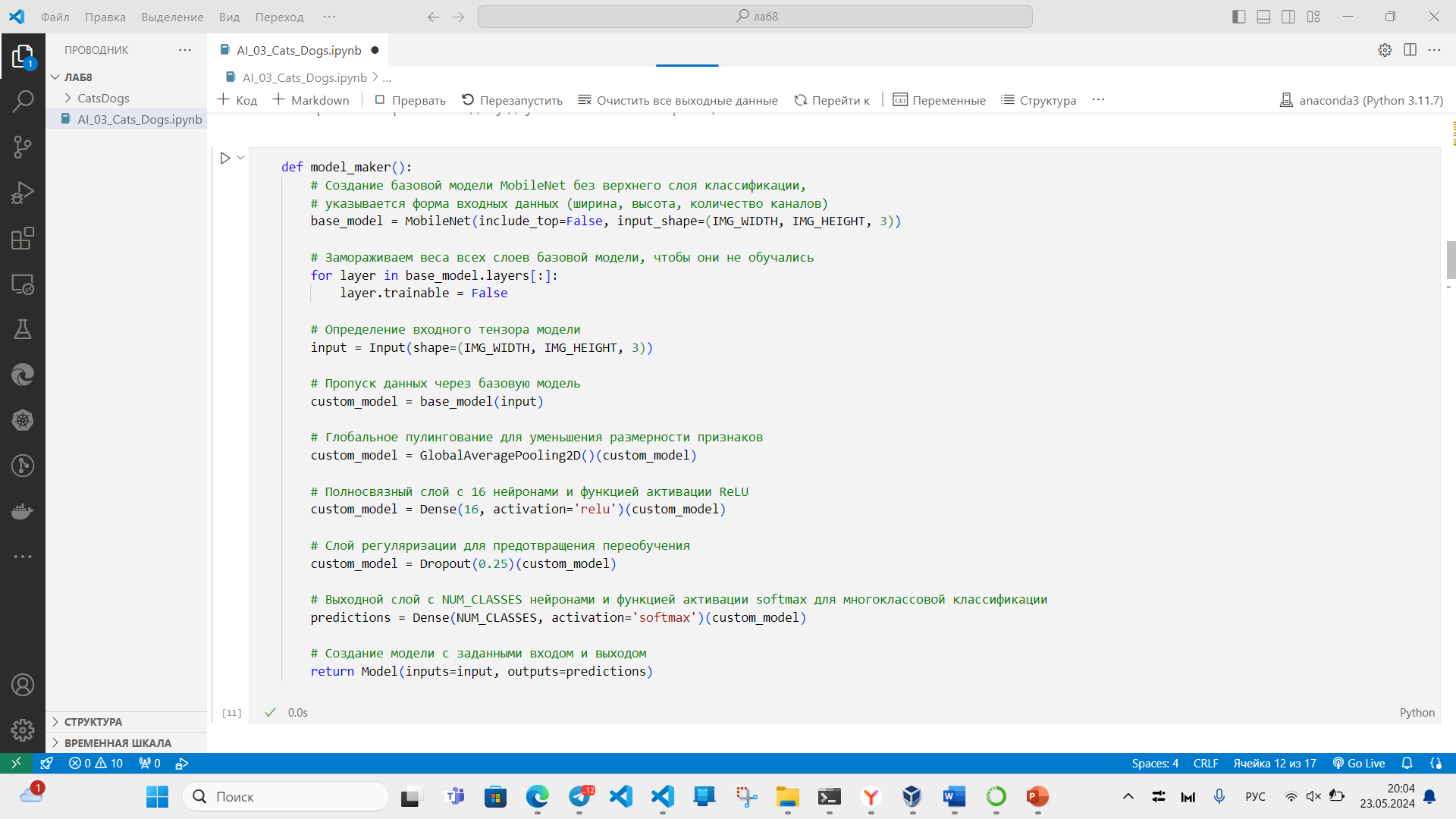


Рисунок 8 – Создание модели

Компилируем и обучаем модель (рисунок 9).

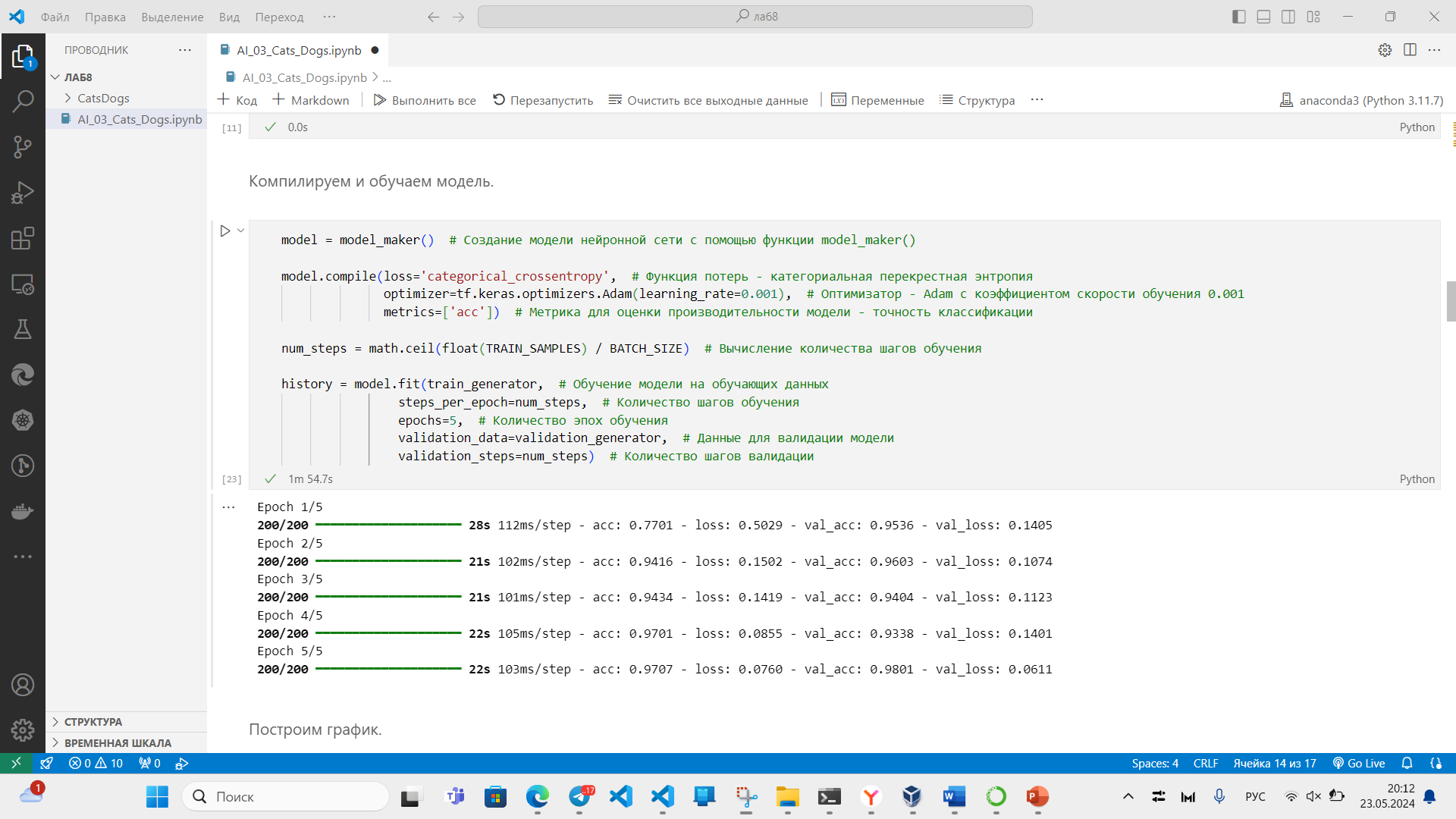


Рисунок 9 – Обучение модели

Построим график (рисунок 10).

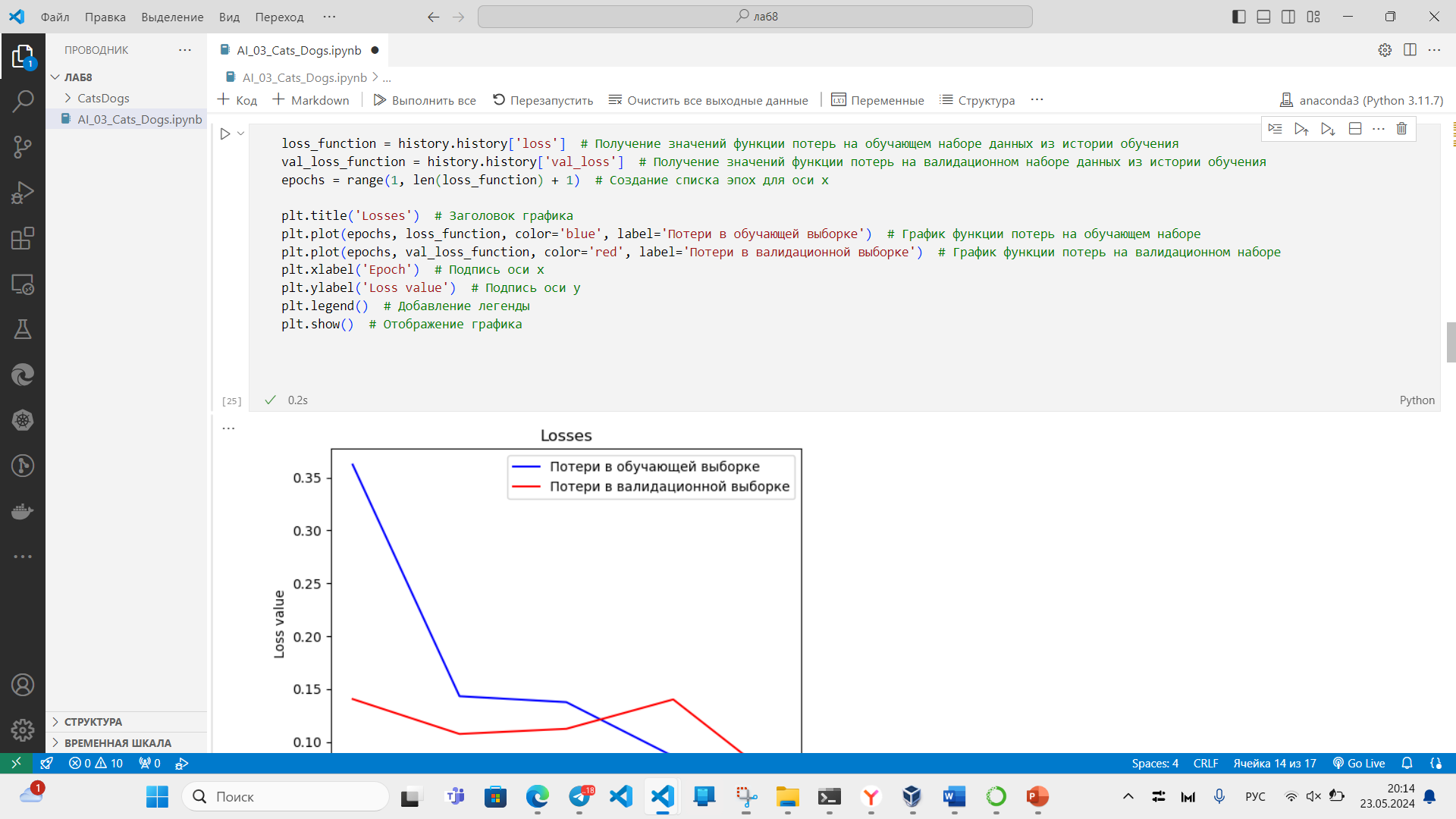


Рисунок 10 – Построение графика

Результат построения графика (рисунок 11).

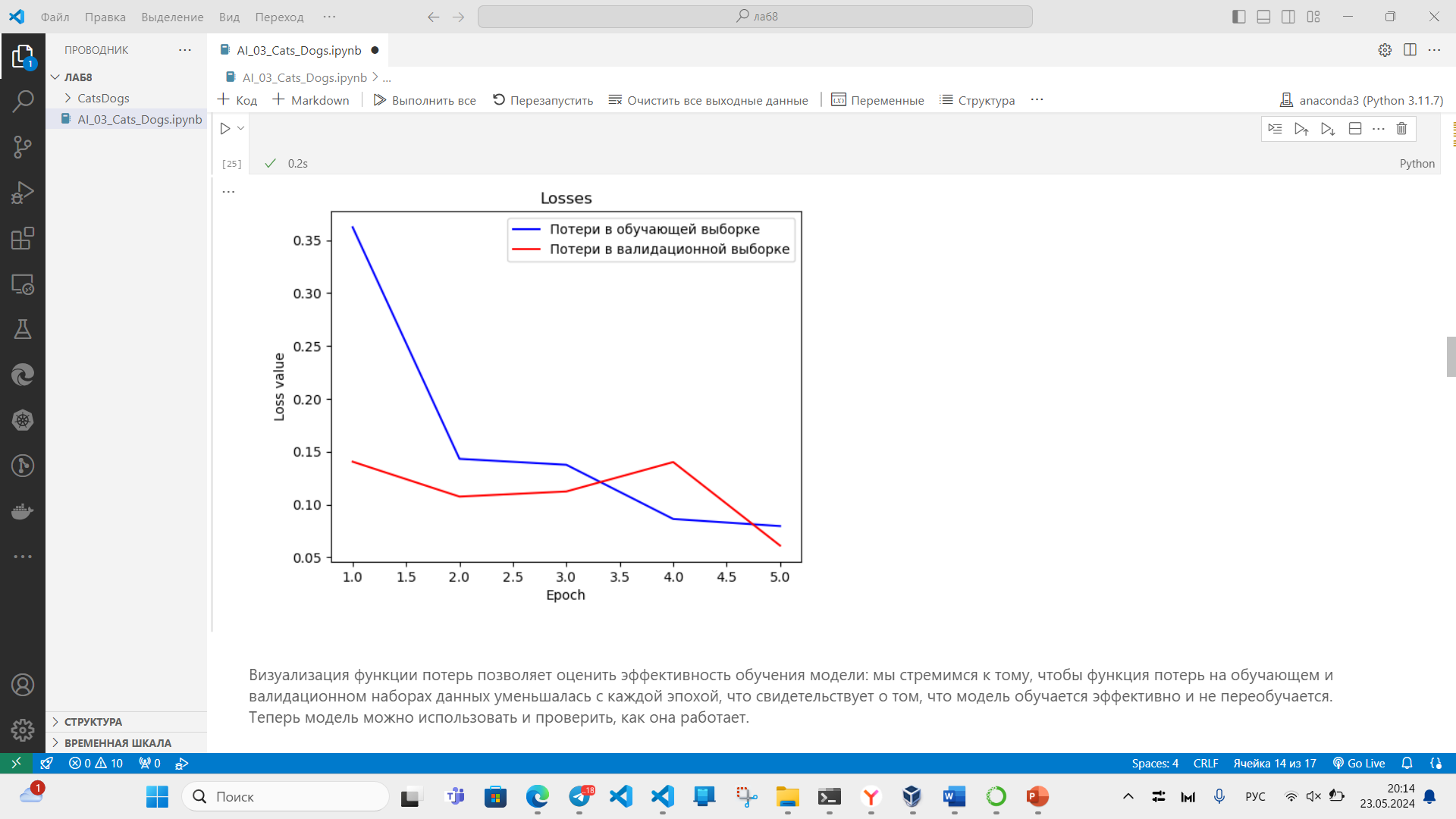


Рисунок 11 –График

Визуализация функции потерь позволяет оценить эффективность обучения модели: мы стремимся к тому, чтобы функция потерь на обучающем и валидационном наборах данных уменьшалась с каждой эпохой, что свидетельствует о том, что модель обучается эффективно и не переобучается. Теперь модель можно использовать и проверить, как она работает   
(рисунок 12).

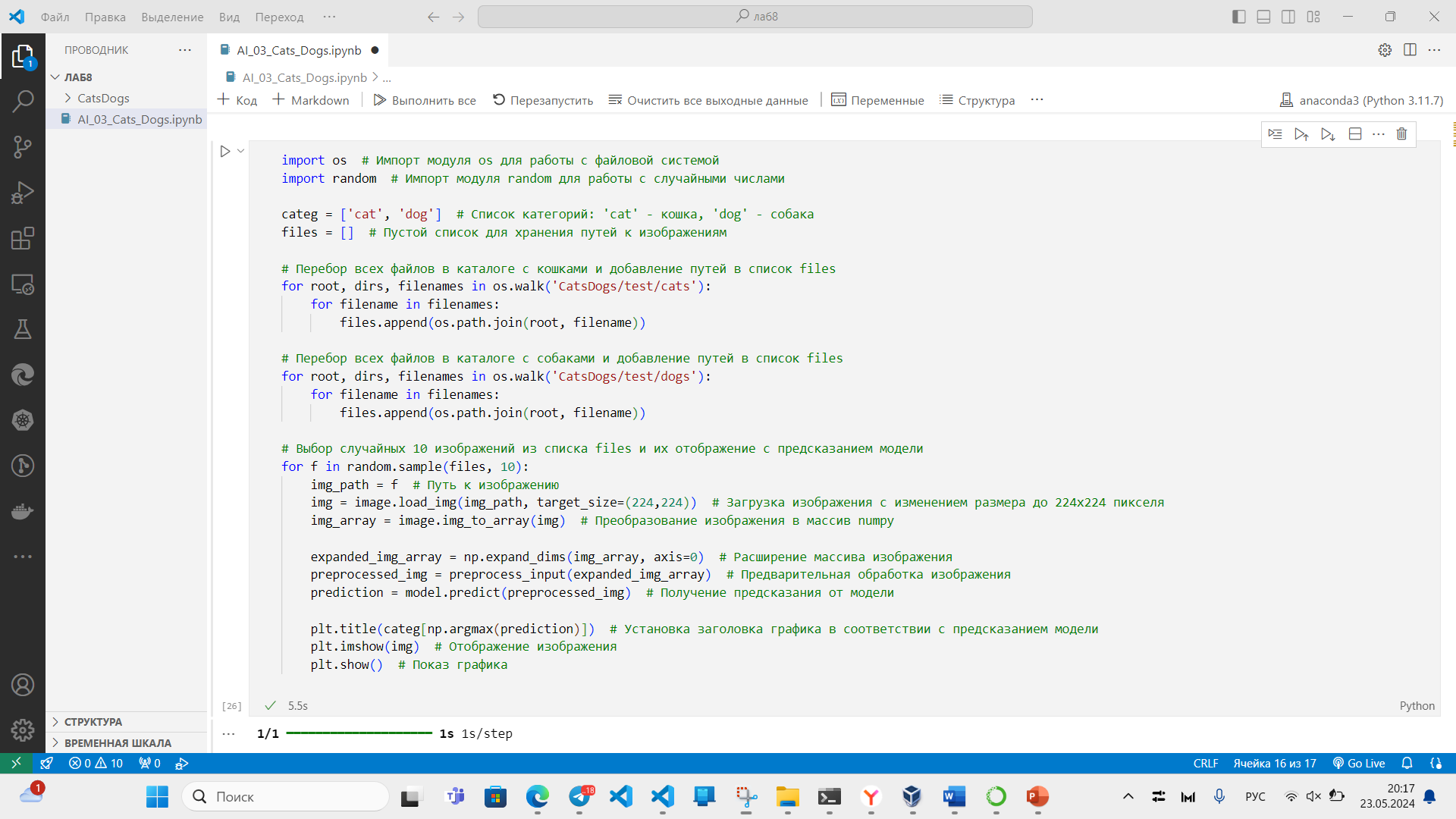


Рисунок 12 – Проверка программы

В данных таблицы 1 видно, как нейронная сеть отсортировала фотографии и подписала к каждому легенду.

Таблица 1 – Фотографии

|  |  |
| --- | --- |
|  |  |
|  |  |

Продолжение таблицы 1

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

# Задания для защиты

Скачиваем файлы, отсортированные по папкам. В папке test находятся файлы для проверки нейросети. Внутри этих папок есть папки, где собраны несколько десятков фотографий с машинами, байками, тракторами и автобусами (рисунок 13).

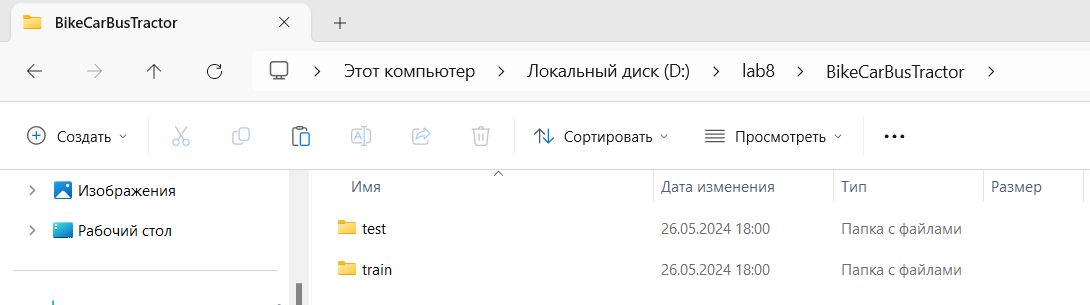


Рисунок 13 – Скачивание файлов

После этого проделываем аналогично, что и с предыдущими файлом, но изменяем путь к файлам. Подключаем библиотеки (рисунок 14).

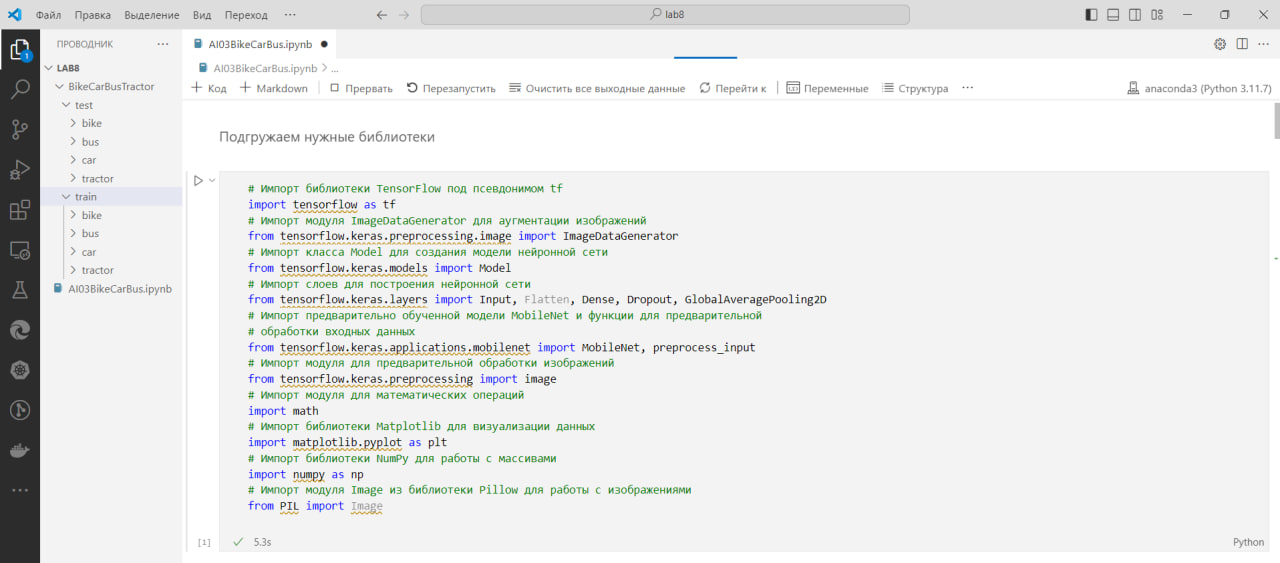


Рисунок 14 – Подключение библиотек

Изменяем значения констант на те, которые нам необходимы, меняем путь к каталогу (рисунок 15).

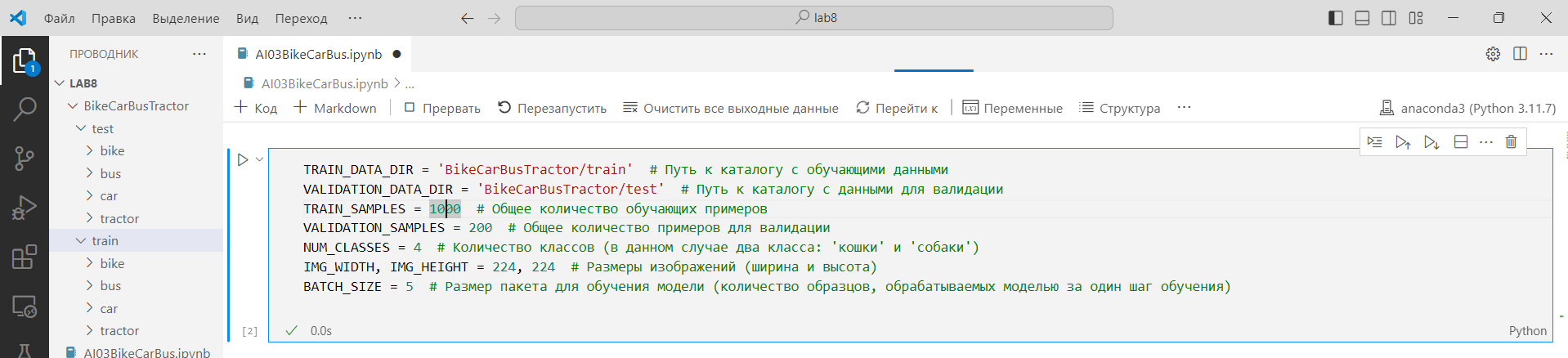


Рисунок 15 – Добавление констант

Создаем первый генератор данных (рисунок 16).

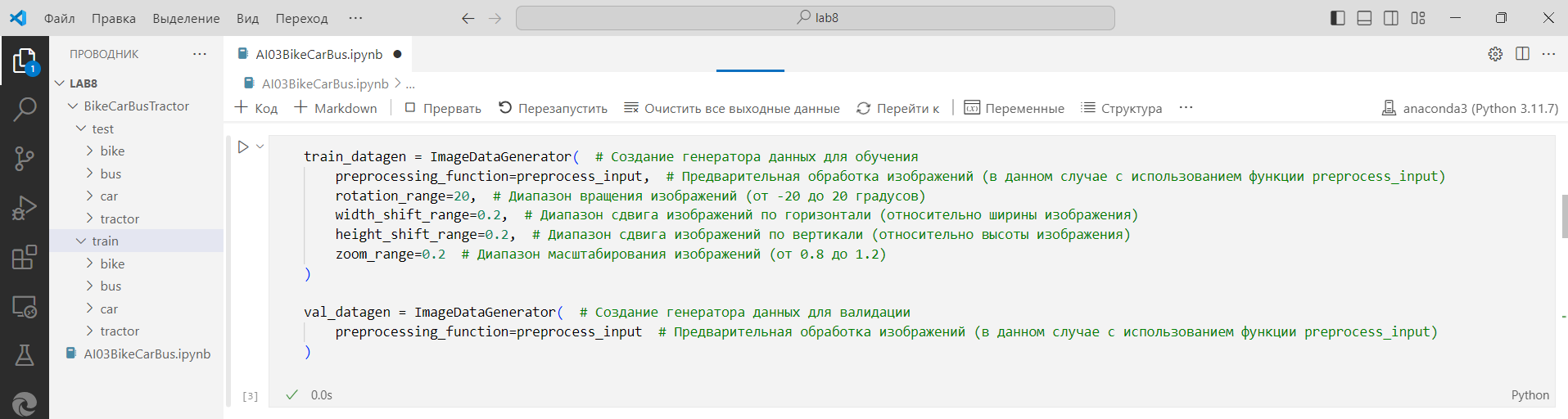
****

Рисунок 16 – Создание первого генератора данных

Затем мы создаем второй генератор данных (рисунок 17).

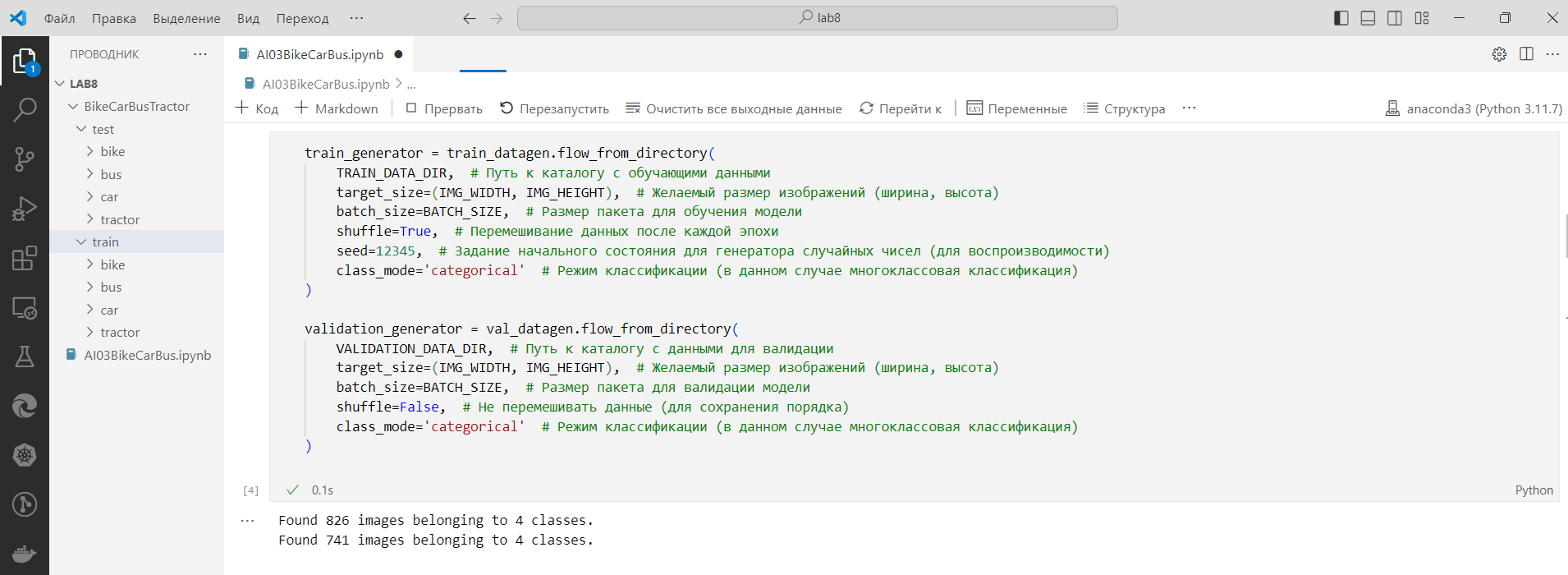


Рисунок 17 – Создание второго генератора данных

В базовой модели отключаем обучение. Входное изображение поступает на вход базовой модели. После базовой модели добавляется несколько слоев: слой пулинга GlobalAveragePooling2D, полносвязный слой из 16 нейронов Dense, механизм Дропаут. Таким образом решается задача двухклассовой классификации (рисунок 18).

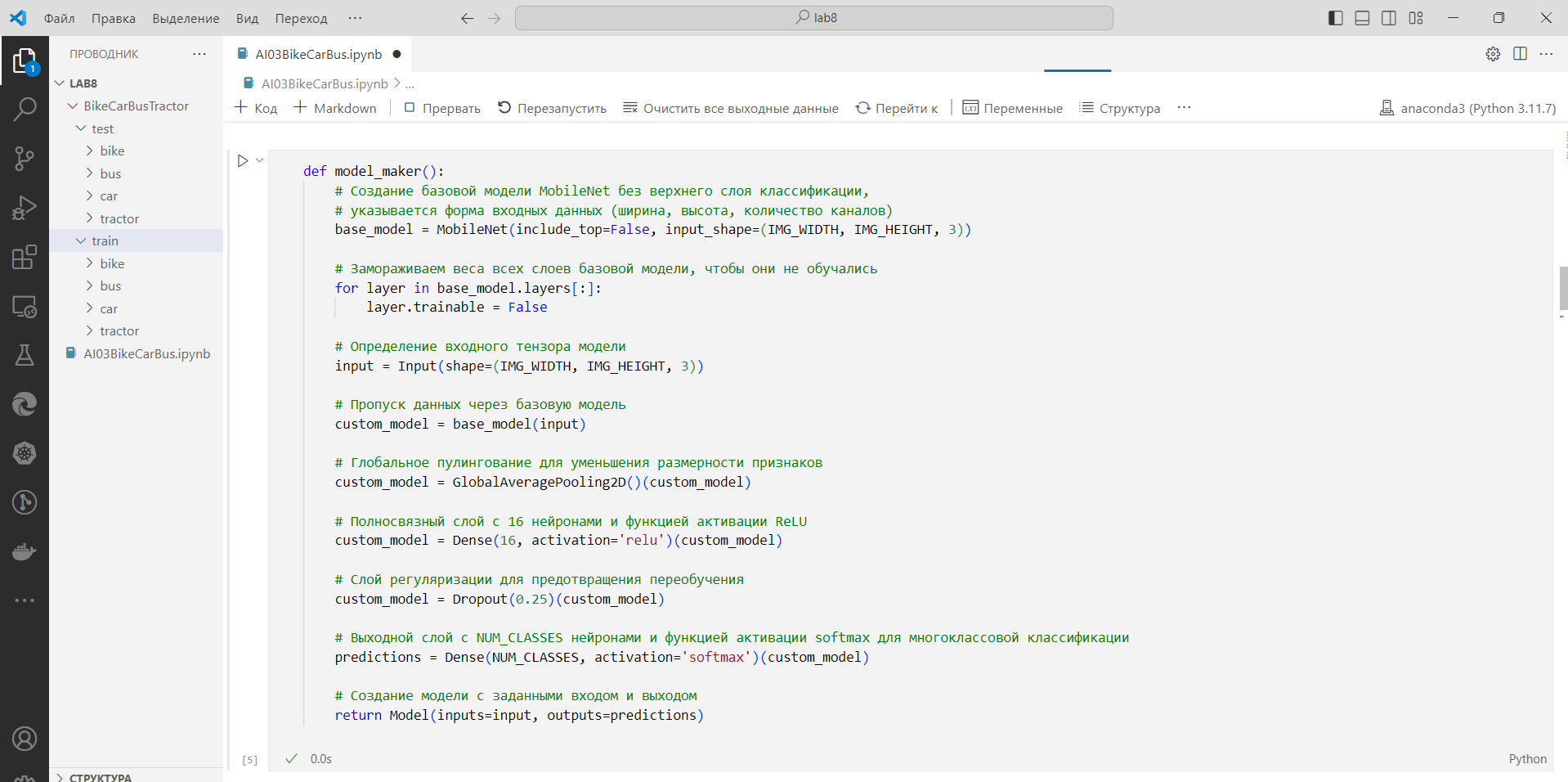


Рисунок 18 – Создание модели

После чего происходит обучение модели (рисунок 19).

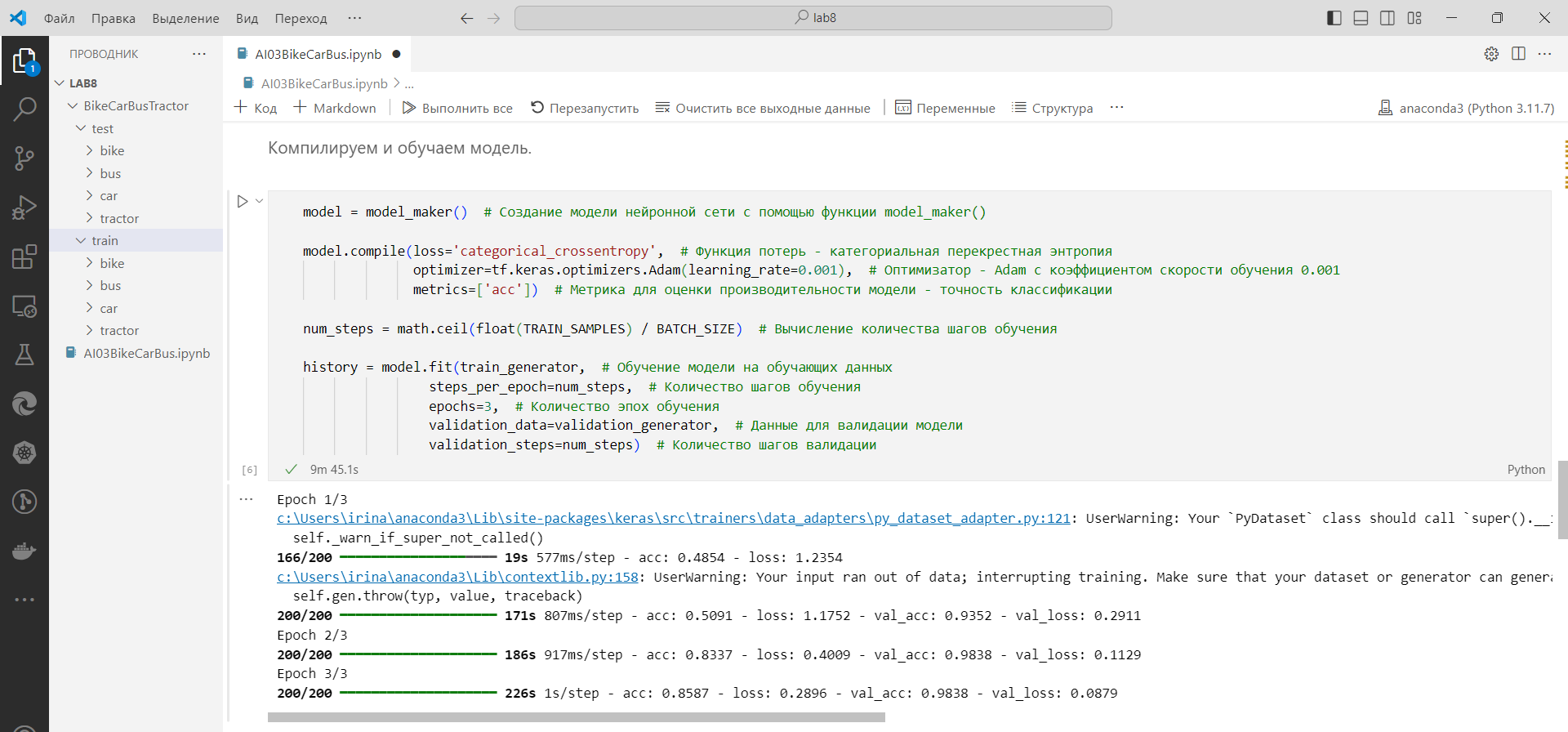


Рисунок 19 – Обучение модели

Затем мы пишем код для построения графика (рисунок 20).

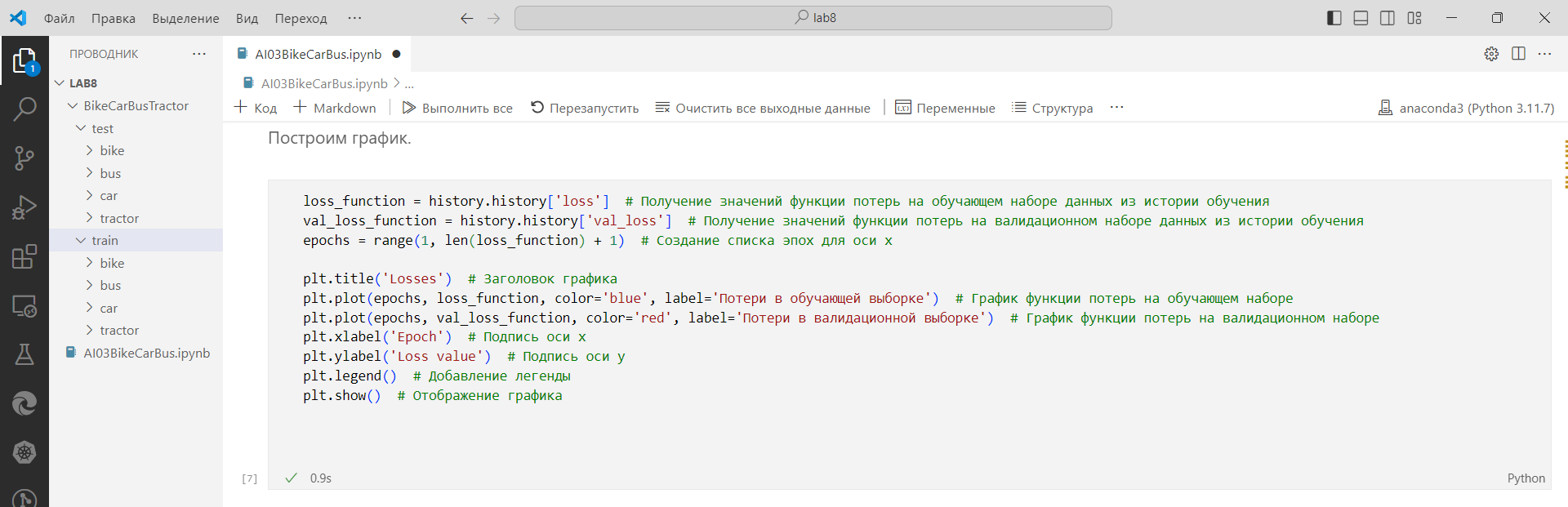


Рисунок 20 – Построение графика

График идет на спад, что говорит о том, что потерь в обучающей и валидационной выборке нет (рисунок 21).

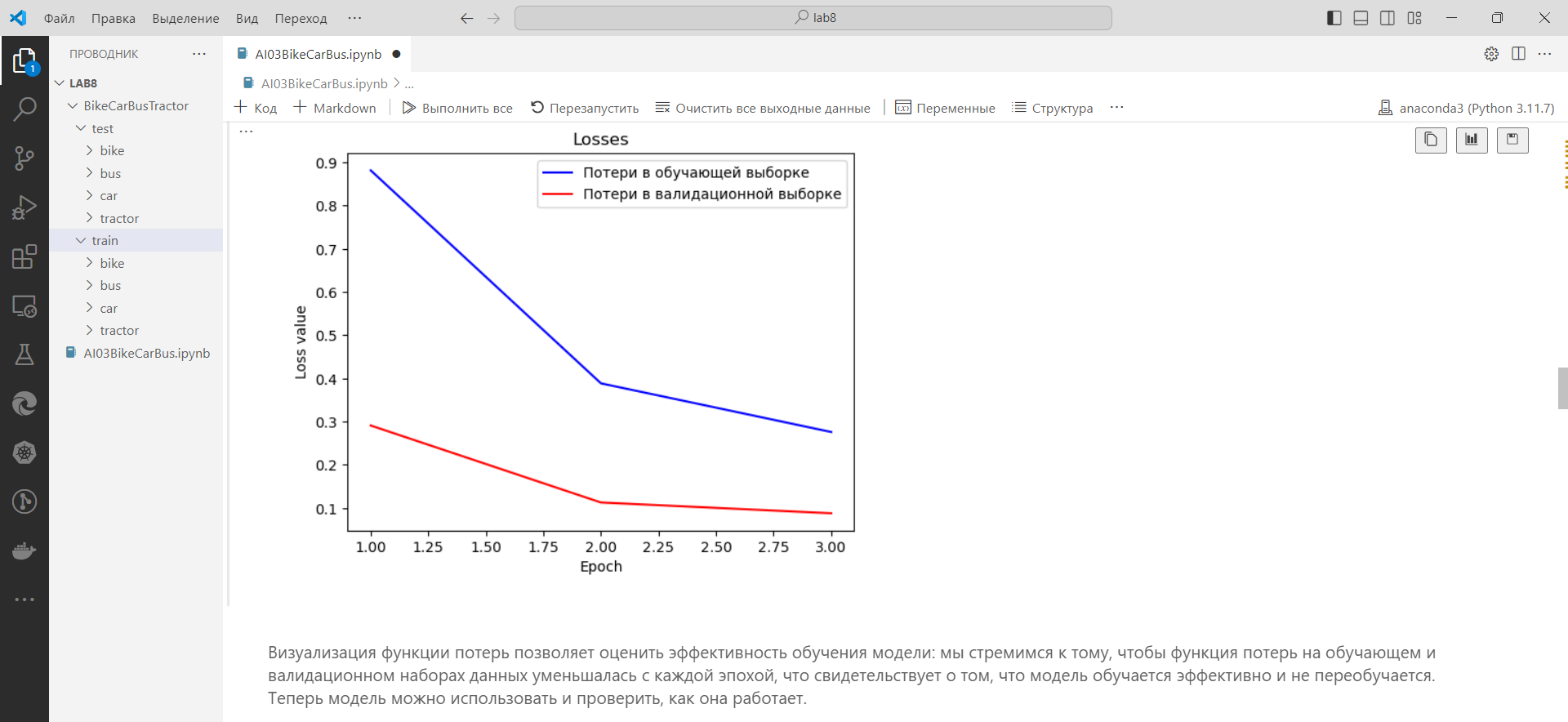


Рисунок 21 – Результат построения графика

Благодаря визуализация функции потерь видно, что модель обучается эффективно и не переобучается. Теперь эту модель можно использовать и проверить, как она работает (рисунок 22).

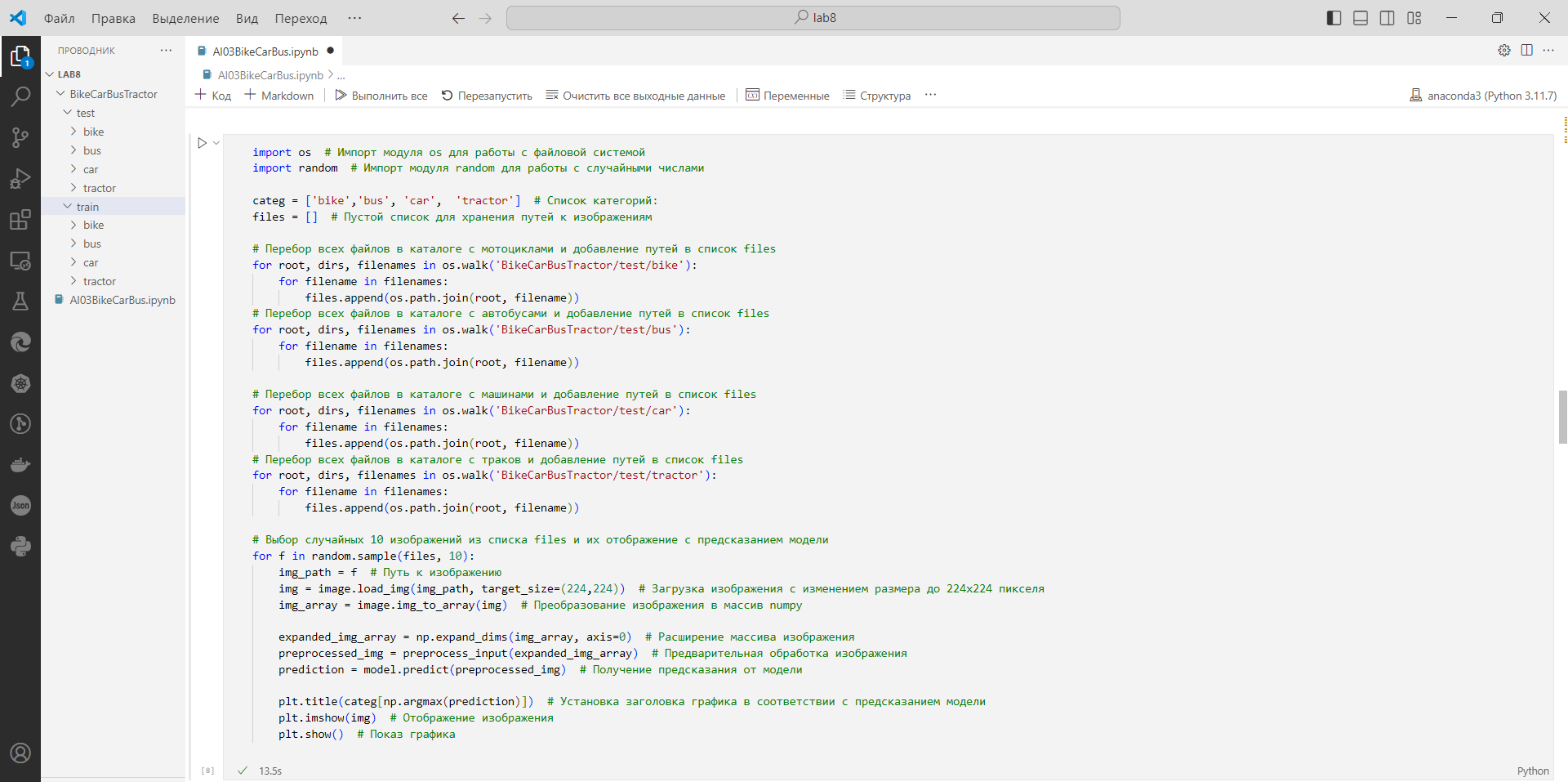


Рисунок 22 – Проверка программы

В данных таблицы 2 мы видим, как нейронная сеть отсортировала фотографии и подписала к каждому легенду.

Таблица 2 – Фотографии

|  |  |
| --- | --- |
|  |  |
|  |  |

Продолжение таблицы 1

|  |  |
| --- | --- |
|  |  |
|  |  |

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

На данный момент есть множество готовых нейронных сетей, которые умеют выделять осмысленно информацию, кроме MobileNet. Вот некоторые из них:

• EfficientNet разработана Google в 2019 году, предобучена на ImageNet, доступна в популярных библиотеках машинного обучения, поддерживает перенос обучения;

• ShuffleNet разработана Megvii в 2018 году, предобучена на ImageNet, доступна в популярных библиотеках машинного обучения, поддерживает перенос обучения;

• SqueezeNet разработана DeepMind в 2016 году, предобучена на ImageNet, доступна в популярных библиотеках машинного обучения, поддерживает перенос обучения;

• ResNet разработана Microsoft в 2015 году, предобучена на ImageNet, доступна в популярных библиотеках машинного обучения, поддерживает перенос обучения;

• VGGNet разработана Oxford Visual Geometry Group в 2014 году, предобучена на ImageNet, доступна в популярных библиотеках машинного обучения, поддерживает перенос обучения.

Каждая нейросеть имеет свои особенности, преимущества и недостатки (рисунок 23).

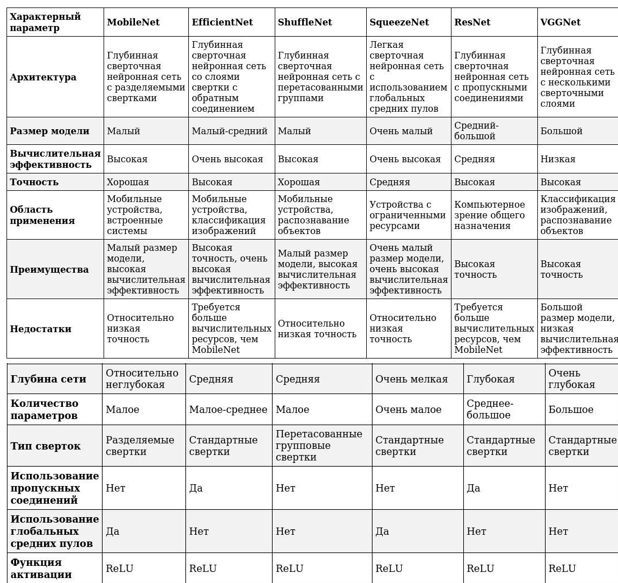


Рисунок 23 – Сравнение нейросетей с MobileNet

На следующем рисунке представлены некоторые характеристики нейронных сетей (рисунок 24).

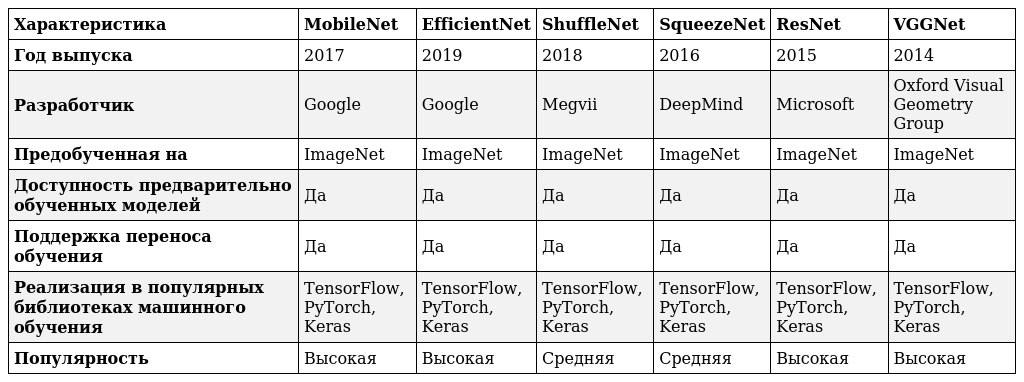


Рисунок 24 – Характеристики нейронных сетей

2 ) Как влияет количество эпох обучения на результат нейронной сети MobileNet?

Количество эпох обучения оказывает значительное влияние на результат нейронной сети MobileNet. Эпоха обучения представляет собой полный проход через весь набор данных. Во время каждой эпохи нейронная сеть обновляет свои веса и смещения, чтобы минимизировать функцию потерь.

Влияние количества эпох обучения на результат MobileNet:

• если нейронная сеть MobileNet обучается на недостаточном количестве эпох, она может не сойтись или сойтись к локальному минимуму. В результате модель может плохо обобщать новые данные и иметь низкую точность;

• при оптимальном количестве эпох обучения нейронная сеть MobileNet достигает хорошей сходимости и обобщения. Модель эффективно изучает шаблоны в данных и достигает высокой точности;

• обучение нейронной сети MobileNet на слишком большом количестве эпох может привести к переобучению. Модель становится слишком приспособленной к обучающим данным и начинает запоминать отдельные примеры, а не общие закономерности. Это приводит к снижению точности на новых данных.

Оптимальное количество эпох обучения для нейронной сети MobileNet зависит от размера и сложности набора данных, архитектуры сети, начального веса, гиперпараметров обучения (скорость обучения, размер пакета).

Обычно оптимальное количество эпох определяется путем мониторинга точности модели на наборе данных проверки во время обучения. Обучение следует прекратить, когда точность на проверочном наборе данных перестает улучшаться или начинает снижаться.

В общем, для небольших и простых наборов данных и архитектур сетей меньшее количество эпох может быть достаточным. Для больших и сложных наборов данных и архитектур сетей может потребоваться большее количество эпох.

Рекомендовано начинать с небольшого количества эпох (например, 20-50) и постепенно увеличивайте их, пока точность на проверочном наборе данных не перестанет улучшаться. Так же можно использовать методы ранней остановки, чтобы предотвратить переобучение и экспериментировать с разными значениями гиперпараметров обучения, чтобы найти лучшую комбинацию для конкретной задачи.

Вывод

В ходе данной лабораторной работе мы познакомились с языком программирования Python. Узнали, что такое нейронные сети, какими они бывают и как работают. Познакомились с нейронной сетью MobileNet сортировки фото. Полученные знания применили на практике для сортировки фото.