|  |  |
| --- | --- |
| Федеральное агентство железнодорожного транспорта  Омский государственный университет путей сообщения  Кафедра «Автоматика и системы управления»  ЗНАКОМСТВО С PYTHON  Лабораторная работа № 8  по теме «Информационные технологии» | |
|  | Студент гр. 23з                            М. А. Медведев, О. Н. Пастушенко, А. В. Титова, М. И. Шруб  «    »                2024 г.  Руководитель –  старший преподаватель кафедры «АиСУ»                            Т.В. Васеева  « 12   » Апреля 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

## Настройка среды

Наша группа выбрала среду Yupyter Notebook–это веб-приложение с открытым исходным кодом, которое широко используется в области Data Science для визуализации данных. Оно представляет собой среду разработки, где можно сразу видеть результат выполнения кода и его отдельных фрагментов. Запускаем Yupyter Notebook и вставляем локальный репозиторий (рисунки 1-2).

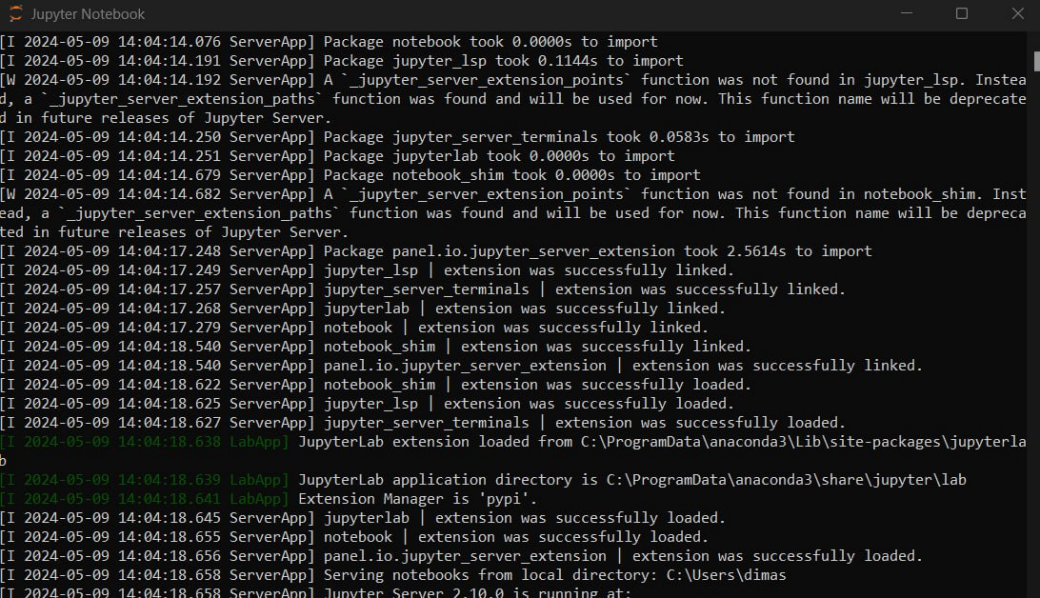


Рисунок 1 – Запуск Jupyter Notebook

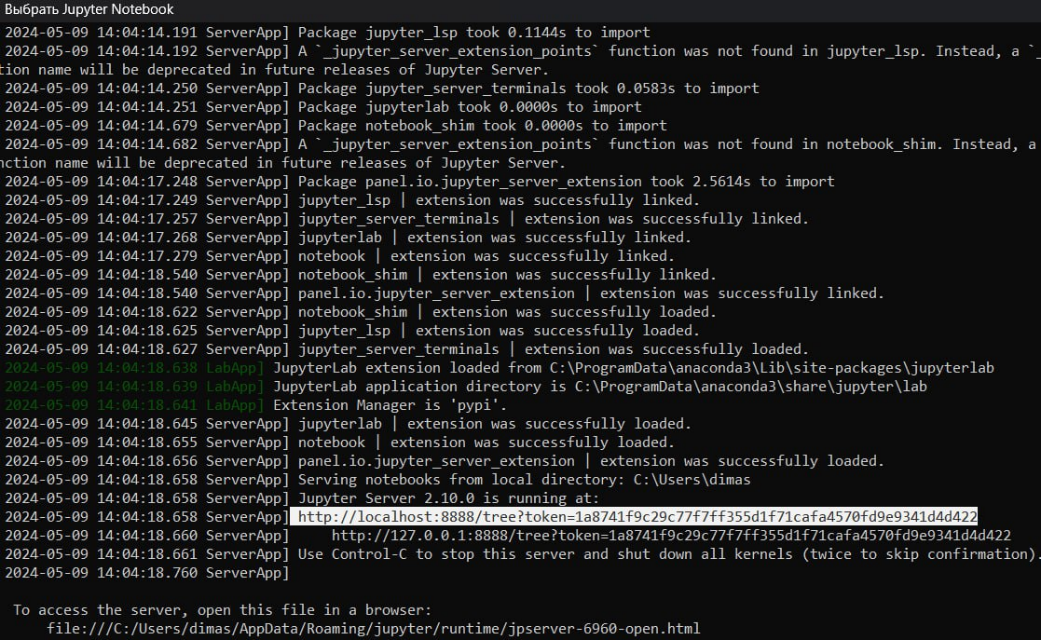


Рисунок 2 – Ссылка локального сервера

Далее запускаем программу через Jupyter, всиавояем ссылку локального репозитория и нажима на Python Anaconda. Затем устанавливаем библиотеки tensorflow (рисунки 3-6).

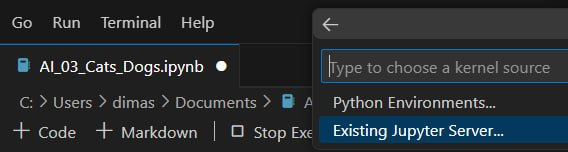


Рисунок 3 – Запуск программы через юпитер

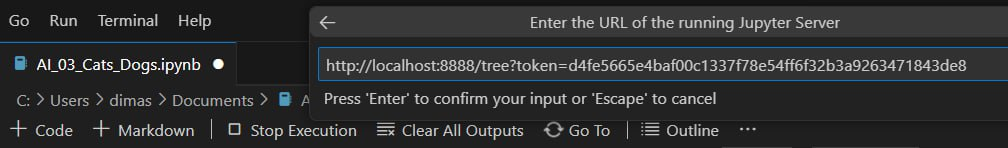


Рисунок 4 – Ссылка на локальный сервер

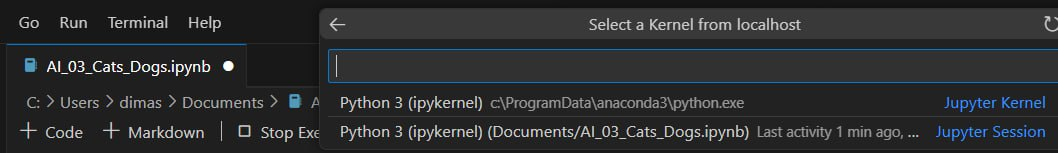


Рисунок 5 – Python Anaconda

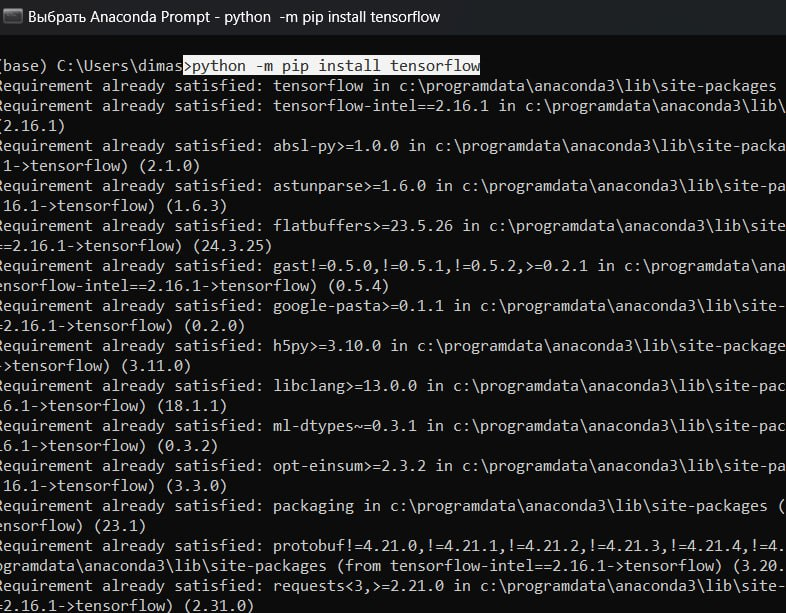


Рисунок 6 – Установка библиотеки tensorflow

## Работа с файлом

Начинаем сортировку собак и кошек с помощью нейронной сети. Для этого сначала подгружаем библиотеки (рисунок 7).

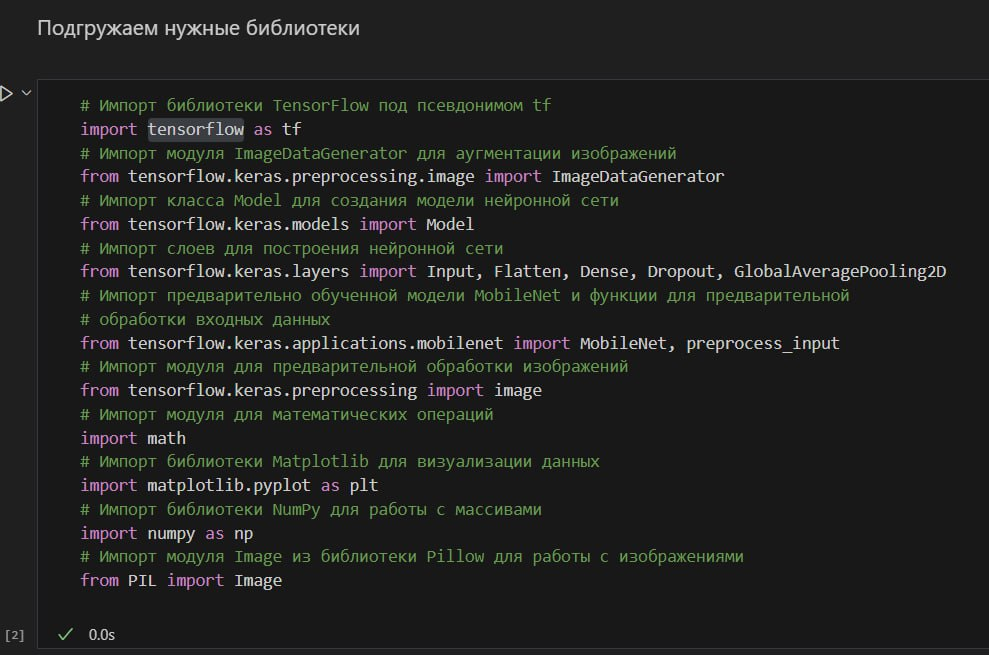


Рисунок 7 – Подгрузка библиотек

Далее скачиваем уже отсортированные файлы и работаем с ними. Для этого нам необходимо прописать константы. TRAIN\_DATA\_DIR отвечает за Путь к каталогу с обучающими данными, VALIDATION\_DATA\_DIR за путь к каталогу с данными для валидации, TRAIN\_SAMPLES за общее количество обучающих примеров, VALIDATION\_SAMPLES за общее количество примеров для валидации, NUM\_CLASSES за количество классов (в нашем случае два класса: «кошки» и «собаки»), IMG\_WIDTH, IMG\_HEIGHT за размеры изображений (ширина и высота), BATCH\_SIZE за размер пакета для обучения модели (количество образцов, обрабатываемых моделью за один шаг обучения) (рисунок 8).

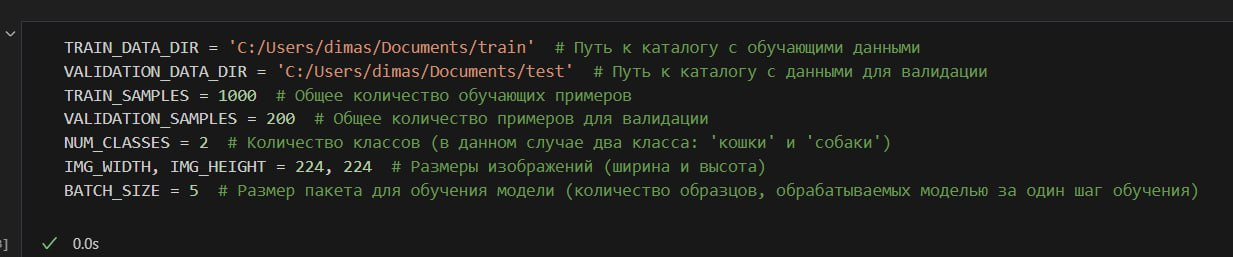


Рисунок 8 – Константы

Затем создаются два генератора данных. Для обучения нейросети недостаточно пары сотен фотографий. Поэтому будет использоваться такой приём как аргументация - это процесс искусственного создания новых данных на основе существующих. Она применяется для увеличения объёма обучающей выборки, повышения разнообразия данных и улучшения качества моделей машинного обучения. Далее 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 9).



Рисунок 9 – 16 слоев нейронами

Следующим шагом компилируем и обучаем модель. Мы использовали 10 эпох (рисунок 10).

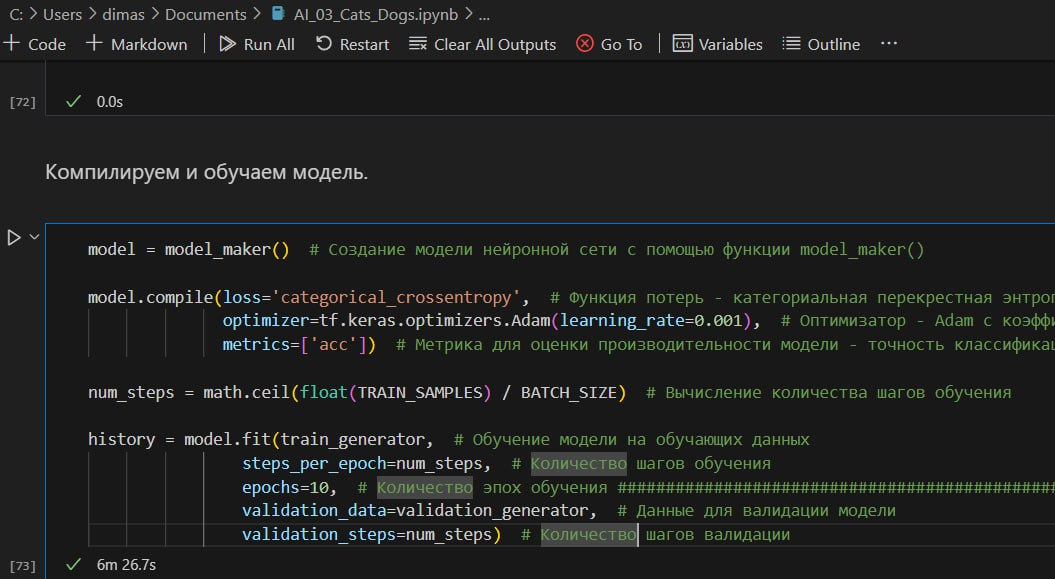


Рисунок 10 – 10 эпох обучения

Как итог получаем график (рисунок 11).

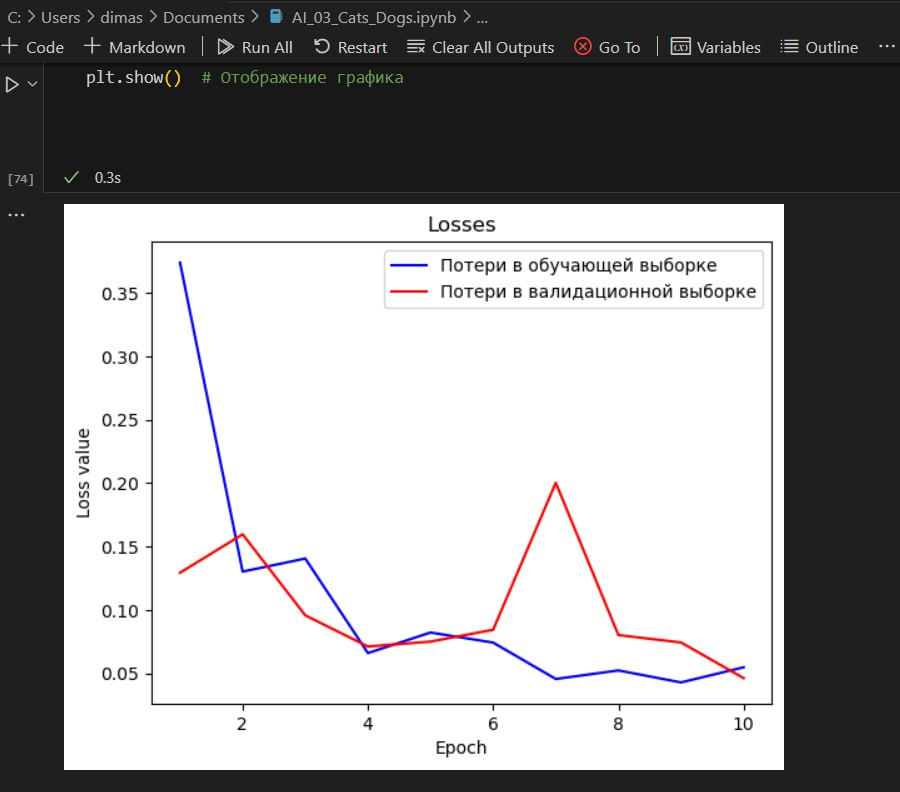


Рисунок 11 – График

# Контрольные вопросы

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

YOLO (You Only Look Once) — архитектура нейронных сетей, предназначенная для детекции объектов на изображении. Отличительной особенностью YOLO является подход к решению задачи детекции.

Один из способов решения задачи детекции заключается в разбиении изображения на квадратные области, затем классификация этих областей на наличие объекта и классификация самого объекта.

SSD (Single Shot MultiBox Detector) – используются наиболее удачные «хаки» архитектуры YOLO (например, non-maximum suppression) и добавляются новые, чтобы нейросеть быстрее и точнее работала. Отличительная особенность: различение объектов за один прогон с помощью заданной сетки окон (default box) на пирамиде изображений.

SqueezeNet – очень маленькая, но точная нейросеть. Сама по себе не решает задачу обнаружения объектов. Однако может применяться при комбинации различных архитектур. И использоваться в мобильных устройствах. Отличительной особенностью является то, что сначала данные сжимаются до четырех 1×1 сверточных фильтров, а затем расширяются до четырех 1×1 и четырех 3×3 сверточных фильтров. Одна такая итерация сжатия-расширения данных называется «Fire Module».

DeepLab (Semantic Image Segmentation with Deep Convolutional Nets) – сегментация объектов на изображении. Отличительной особенностью архитектуры является разряженная (dilated convolution) свертка, которая сохраняет пространственное разрешение. Затем следует стадия постобработки результатов с использованием графической вероятностной модели (conditional random field), что позволяет убрать небольшие шумы в сегментации и улучшить качество отсегментированного изображения.

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

В мире искусственных нейронных сетей эпоха (англ. epoch) – это один цикл обучения на массиве данных. Обучение нейронной сети обычно занимает много эпох. Проще говоря, если мы снабжаем нейронную сеть обучающими данными в различных паттернах на протяжении более чем одной эпохи, мы ожидаем улучшения генерализации, когда даем ей свежий ненаблюдаемый вход (тестовые данные).

## Сортировка фото (выбор зависит от собственных предпочтений) с помощью нейронной сети

Прописываем константы, только с другим путем каталога (рисунок 12).

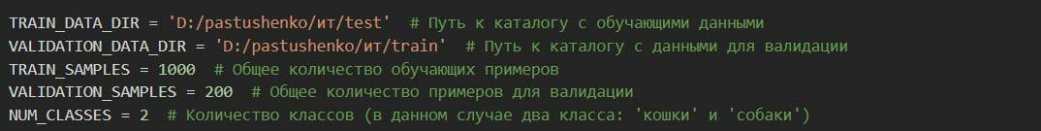


Рисунок 12 – Константы

Далее создаем базовую модель нейронной сети, обучаем ее, используя 4 эпохи и строим график (рисунки 13-15).

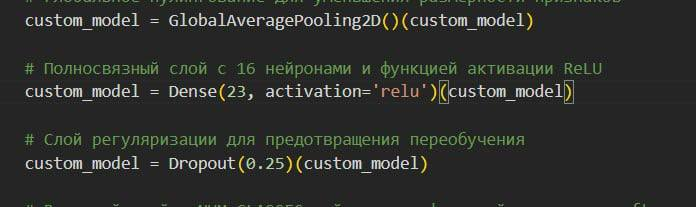


Рисунок 13 – Модель нейронной сети

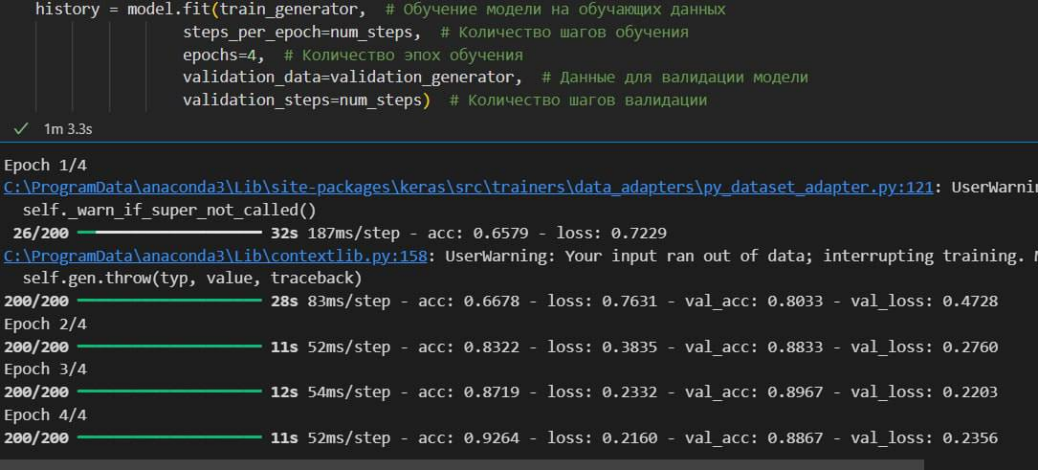


Рисунок 14 – Обучение нейронной сети

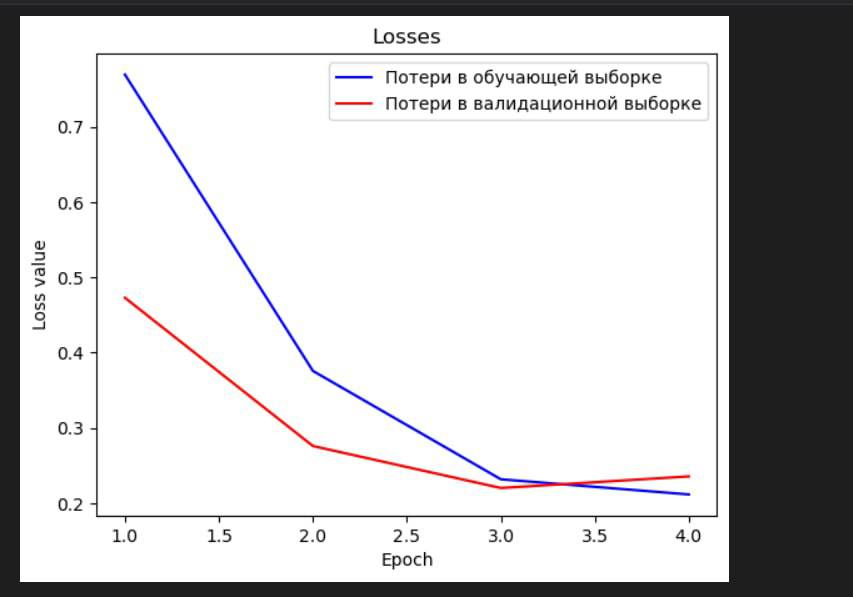


Рисунок 15 – График

Вывод

В ходе лабораторной работы мы познакомились с языком программирования Python Мы научились сортировать фото с помощью нейронной сети, которую сами обучили.