

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«**СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ**»

Институт космических и информационных технологий

Кафедра вычислительной техники

УТВЕРЖДАЮ

Заведующий кафедрой

_____ О.В. Непомнящий

« ___ » _____ 2023 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 – Информатика и вычислительная техника

Классификация изображений растений на основе нейронной сети

Руководитель	_____	доцент, канд. техн. наук	Н.Ю. Сиротинина
	подпись, дата	должность, ученая степень	инициалы, фамилия
Выпускник	_____		Г.Ю. Аникутин
	подпись, дата		инициалы, фамилия
Нормоконтролер	_____		Н.Ю. Сиротинина
	подпись, дата		инициалы, фамилия

Красноярск 2023

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

УТВЕРЖДАЮ

Заведующий кафедрой

_____ О.В. Непомнящий

« ____ » _____ 2022 г.

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы**

Красноярск 2022

РЕФЕРАТ

Выпускная квалификационная работа по теме «Классификация изображений растений на основе нейронной сети» содержит 39 страниц текстового документа, 23 иллюстрации, 2 таблицы, 8 использованных источников.

КЛАССИФИКАЦИЯ ИЗОБРАЖЕНИЙ, КЛАССИФИКАЦИЯ ИЗОБРАЖЕНИЙ РАСТЕНИЙ, ПАТОЛОГИИ РАСТЕНИЙ, НЕЙРОННАЯ СЕТЬ, СВЕРТОЧНАЯ НЕЙРОННАЯ СЕТЬ

Цель работы – создание системы, выполняющей по фотографии классификацию растений в соответствии с заболеванием, которым они страдают.

Задачи работы:

1. Провести анализ задания на работу, изучение предметной области, аналитический обзор существующих решений схожих задач, выбор модели распознавания и технологию ее реализации.
2. Спроектировать, реализовать и обучить модель.
3. Спроектировать, реализовать и протестировать систему диагностики растений.

По итогу выполнения выпускной работы получены следующие результаты.

1. В качестве модели распознавания выбрана сверточная нейронная сеть.
2. Найдены и подготовлены наборы данных для обучения.
3. Предложена архитектура нейронной сети и выполнен подбор гиперпараметров.
4. Обучена нейронная сеть, классифицирующая изображения растений по нескольким классам, показывающая точность (accuracy) 87-89%.
5. Разработано автономное приложение, позволяющее распознавать заболевания растений по фотографии.

СОДЕРЖАНИЕ

Введение.....	3
1 Обзор современного состояния и способов решения задачи.....	5
1.1 Описание предметной области.....	5
1.2 Требования к разрабатываемой системе.....	8
1.3 Аналитический обзор существующих решений.....	9
1.4 Выводы по главе.....	11
2 Проектирование и реализация нейронной сети.....	12
2.1 Особенности организации сверточной нейронной сети.....	12
2.2 Обучающая выборка и аугментация данных.....	14
2.3 Реализация нейронной сети.....	17
2.4 Подбор гиперпараметров.....	19
2.5 Обучение и тестирование нейронной сети.....	23
2.6 Выводы по главе.....	27
3 Проектирование и реализация системы.....	28
3.1 Функциональное проектирование системы.....	28
3.1.1 Прецедент «Классификация изображений».....	29
3.1.2 Прецедент «Просмотр данных изображения».....	30
3.1.3 Прецедент «Пометка ошибки тестирования».....	32
3.1.4 Прецедент «Загрузка весов модели».....	33
3.2 Интерфейс системы.....	36
Заключение.....	39
Список использованных источников.....	40
Приложение А Код с реализацией нейронной сети.....	41

ВВЕДЕНИЕ

Гидропонное выращивание растений является важной частью сельского хозяйства. Одним из условий высокого качества продукции и эффективности производства в условиях гидропонной фермы, которые характеризуются плотной посадкой растений, ограниченным пространством и повышенной влажностью, является своевременное выявление болезней и патологий. Многие заболевания растений требуют раннего выявления и принятия своевременных мер, иначе они быстро распространяются и приводят к снижению урожайности и ухудшению товарного вида продукции.

Еще одной особенностью выращивания растений на современной гидропонной ферме является ее ограниченная площадь и автоматизация всех процессов, связанных с выращиванием растений. Это подразумевает наличие большого числа датчиков, контролируемого освещения и состава питательной жидкости, а так же прочих современных цифровых средств контроля и регулирования. Поэтому автоматизация отслеживания заболеваний растений (а в перспективе и общего их состояния) с использованием современных интеллектуальных технологий является актуальной и реализуемой без больших дополнительных затрат.

Цель работы – создание системы, выполняющей по фотографии классификацию растений в соответствии с заболеванием, которым они страдают.

В ходе выполнения работы требуется решить следующие задачи:

1. Провести анализ задания на работу, изучение предметной области, аналитический обзор существующих решений схожих задач, выбор модели распознавания и технологию ее реализации.
2. Спроектировать, реализовать и обучить модель.
3. Спроектировать, реализовать и протестировать систему диагностики растений.

По итогу выполнения выпускной работы получены следующие результаты.

Анализ работ, связанных с заболеваниями и повреждениями растений методами искусственного интеллекта, показал, что для решения данной задачи наиболее эффективными являются сверточные нейронные сети.

Найдены и подготовлены наборы данных для обучения.

Рассмотрены принципы работы сверточных нейронных сетей, предложена архитектура нейронной сети и выполнен подбор гиперпараметров.

Обучена нейронная сеть, классифицирующая изображения растений по нескольким классам, показывающая точность (accuracy) около 87-89%.

Разработано автономное приложение, позволяющее распознавать заболевания растений по фотографии.

1 Обзор современного состояния и способов решения задачи

1.1 Описание предметной области

При выращивании растений угрозу может представлять большое количество разнообразных насекомых, грибковых, патогенных бактерий и вирусов. Степень и частота заражения болезнями и насекомыми приводят к снижению или полной потере урожая, непривлекательному внешнему виду и плохому качеству плодов, а в результате – к значительным экономическим потерям. Своевременное обнаружение заболеваний имеет ключевое значение для борьбы с его распространением. Неправильное или поздно начатое лечение так же является значимой проблемой, потому что не останавливает распространение заболеваний. Ошибочные диагнозы несут риски нерационального использования химических веществ из-за чего увеличиваются затраты на лечение растений, вредное воздействие на окружающую среду и здоровье людей, а так же риски появления более устойчивых патогенов. При промышленном гидропонном выращивании ситуацию усугубляет высокая плотность посадки, из-за которой насекомые и заболевания распространяются с большой скоростью и могут поразить весь урожай.

Из-за большого количества патогенных факторов анализ растений требует высоких трудозатрат и привлечения подготовленных сотрудников. Как правило, сотрудников сначала обучают, используя изображения симптомов болезни и повреждения насекомыми, но из-за наличия большого количества переменных в реальном саду им требуется значительное время, чтобы ознакомиться со многими классами симптомов, обусловленными либо возрастом, либо типом инфицированных тканей, либо стадией цикла болезни или развития вредителя. То есть даже обученному сотруднику требуется время и опыт, чтобы работать достаточно эффективно. Форма симптомов также могут меняться с течением времени по мере прогрессирования болезни и старения тканей листьев или плодов.

Кроме того сотрудники тратят много времени, вводя и интерпретируя результаты анализа. Обработка результатов людьми так же занимает много ресурсов.

Распространенной культурой растений, выращиваемых на гидропонных фермах, являются томаты. Выращивание томатов так же проходит на гидропонной ферме на базе Института гастрономии СФУ. Поэтому в дальнейшей работе будут рассматриваться в основном заболевания томатов.

Растения могут быть подвержены множеству грибковых заболеваний, вирусов, воздействию насекомых и другим патогенам. Самыми распространенными заболеваниями томатов являются ранняя гниль томатов (early blight) и септориоз листьев (Septoria leaf spot) [1].

Пример здорового листа томата, выращенного гидропонным способом, приведен на рисунке 1, листа, пораженного ранней гнилью, – на рисунке 2, и листа, пораженного септориозом, – на рисунке 3.



Рисунок 1 – Пример здорового листа томата (healthy)



Рисунок 2 – Пример листа томата, пораженного ранней гнилью (early blight)



Рисунок 3 – Пример листа томата, пораженного септориозом (Septoria leaf spot)

1.2 Требования к разрабатываемой системе

Исходя из задания на работу было определено, что система должна выполнять следующие функции.

1. Загрузка изображений для обработки и выявления пораженных болезнями растений. При работе с системой пользователь загружает в неё свои изображения. Загрузка изображений может выполняться поодиночке или наборами: пользователь загружает изображения в папку, после чего проводится анализ всех изображений в директории.

2. Анализ изображений с целью выявления поражений растений. Система выполняет классификацию загруженных изображений на заранее определенные классы, соответствующие здоровым растениям или одной из определяемых патологий.

3. Обработка изображений разного размера. Поскольку фотографии, с которыми придется работать системе, могут быть потенциально разного размера, система должна приводить их к одному формату, с которым будет работать.

4. Вывод результатов обнаружения поражений, содержащий информацию о наиболее вероятном классе, к которому принадлежит предъявленное изображение или каждое изображение в группе.

5. Просмотр подробных результатов для конкретного изображения. Пользователю должна предоставляться возможность детального просмотра результатов классификации с указанием вероятности принадлежности изображения к каждому из классов.

6. Пометка ошибочных результатов обработки. В случае если пользователь видит, что изображение было классифицировано ошибочно, он должен иметь возможность отметить его, после чего система поместит его в директорию, предназначенную для спорных изображений. Эти изображения в дальнейшем предполагается использовать при дообучении модели.

Точность, с которой система должна классифицировать состояние растения по фотографии, должна быть выше 80%.

1.3 Аналитический обзор существующих решений

Для выбора метода машинного обучения и инструментальных средств разработки системы был проведен анализ актуальных способов решения подобных задач. В качестве параметров сравнения используемых решений выбраны следующие:

- используемая модель машинного обучения;
- среда разработки;
- количество классов состояний растений;
- точность классификации.

Было найдено и проанализировано несколько публикаций о классификации патологий растений методами искусственного интеллекта на русском и английском языке.

В работе [2] описывается использование сверточной нейронной сети для оценки состояния растений в среде разработки MATLAB. Для обучения сети семантической сегментации рассматривался метод получения сегментированных масок путем пороговой бинаризации цветного изображения. Для детектирования состояния растений был подготовлен набор данных, состоящий из 3 000 изображений растений, разделенных на 2 класса: больные и здоровые.

В работе [3] рассматривается разработка нейронных сетей для этой задачи с использованием языка программирования Python. Так же там приведен набор данных для обучения нейронных сетей и варианты реализации. Подробно описаны механизмы обработки изображений для обучающей выборки и работы нейронной сети. Система определяет принадлежность изображений к 4 классам с точностью 95%.

В работе [4] описывается модель сверточной нейронной сети для определения заболеваний по фотографии с мобильного телефона. Подготовлен набор данных изображений листьев яблоневых деревьев, включающий пять классов. Четыре класса с признаками определенных заболеваний листьев яблони и один класс здоровых растений. Система определяет принадлежность изображений к 5 классам с точностью 93%.

В работе [5] рассматривалась задача распознавания заболеваний растений, проанализированы и протестированы архитектуры сверточных нейронных сетей VGG16 и Inception-v3 применительно к задаче распознавания заболеваний растений. Эксперименты с архитектурами нейронных сетей и оптимизаторами (Adam и RMSprop) проводились на одном из датасетов, доступных на платформе Kaggle. Наилучшие результаты при распознавании заболеваний растений были достигнуты для сверточной нейронной сети с архитектурой Inception-v3, с точность 97% на 38 классах.

Сравнение рассмотренных систем показано в таблице 1.

Таблица 1 – Сравнительный анализ существующих решений задачи оценки состояния растений

Название работы	Модель машинного обучения	Среда разработки	Количество классов	Точность
Оценка состояния растений с использованием сверточных нейронных сетей	SqueezeNet	Matlab	2	94%
Plant Pathology 2020	DenseNet, EfficientNet	Python, библиотеки TensorFlow, Keras	4	95%
Выбор модели искусственной сверточной нейронной сети для определения заболеваний яблоневых деревьев	ResNet	Python, фреймворк PyTorch	5	93%
Решение задачи распознавания болезней растений на основе нейронных сетей	Inception-v3	Python, библиотека Keras	38	97%

Модели, описанные в рассмотренных работах, показывают достаточно высокую точность, однако специфика среды разработки и набора данных могут накладывать ограничения. Однако все рассмотренные работы реализовали модели сверточных нейронных сетей, а так же большинство работ рассмотренных при анализе использовало язык программирования Python.

1.4 Выводы по главе

В данной главе было проанализировано задание, сформулированы требования к разрабатываемой системе.

Выполнен анализ существующих аналогов, который показал, что наиболее эффективными в решении поставленных задач являются архитектуры сверточных нейронных сетей.

В рассмотренных работах наиболее распространенным средством разработки является язык Python.

На основании результатов аналитического обзора принято решение, что для выполнения поставленной задачи будет реализована сверточная нейронная сеть, написанная на языке Python, с использованием библиотек TensorFlow и Keras.

Так же будет использована среда разработки google colabatory, так как она предоставляет вычислительные мощности для обучения нейронных сетей.

2 Проектирование и реализация нейронной сети

2.1 Особенности организации сверточной нейронной сети

На основе анализа существующих решений и теоретического материала было принято, что наиболее эффективными в данной задаче являются сверточные нейронные сети.

Сверточная нейронная сеть (Convolutional Neural Network, CNN) - это архитектура нейронной сети, которая используется для обработки данных с пространственной структурой, таких, как изображения или звук.

Функционирование сверточной нейронной сети представляет собой применение фильтра (ядра) к входным данным с целью извлечения признаков. Операция свертки выполняется путем перемещения фильтра по всему входному изображению и вычисления скалярного произведения между фильтром и соответствующими пикселями изображения. Результатом свертки является новое изображение, в котором сохранены только те признаки, которые были выделены фильтром. Для выполнения операции свертки в нейронных сетях используется сверточный слой (Convolutional layer). Сверточный слой состоит из набора фильтров, каждый из которых применяется к входным данным независимо. Результаты свертки каждого фильтра объединяются в выходной тензор. Каждый фильтр имеет свои веса, которые обучаются в процессе обучения нейронной сети.

Нейроны сверточного слоя, как правило, реализуют функцию активации ReLU (Rectified Linear Unit). Это нелинейная функция активации, которая возвращает входное значение аргумента, если оно больше 0. Данная функция является вычислительно эффективной, поскольку выполняется достаточно быстро.

Между сверточными слоями используется функция пулинга, которая понижает размер входных данных по высоте и ширине. Для этого результат свертки разбивается на квадратные фрагменты, и каждому из них в

соответствие ставится определенное значение. Наиболее часто используется функция макс-пулинга, которая принимая максимальное значение из окна ввода для каждого канала ввода.

Выходные сигналы последнего сверточного слоя развертываются в одномерный массив, который поступает на вход полносвязного слоя.

Полносвязные слои идут в конце нейронной сети. Количество нейронов последнего слоя совпадает с количеством классов в нейронной сети. Сигналы на выходе этих нейронов показывают, к какому классу принадлежит изображение.

Результат классификации определяется с использованием функции `softmax`, которая принимает вектор действительных аргументов (сигналов выходного слоя) и преобразует их в вектор, компоненты которого заключены в диапазоне $(0, 1)$, а их сумма равна 1, таким образом, на выходе будут вероятности принадлежности изображения к каждому классу.

При конструировании сверточной нейронной сети необходимо определить следующие гиперпараметры:

- число сверточных слоев;
- параметры ядер для каждого сверточного слоя, включающие количество ядер, размер, смещение ядра;
- используемый вариант обрамления изображения (пэддинг);
- размер ядра и функцию пулинга для каждого слоя;
- количество и размер полносвязных слоев на выходе сверточной части сети.

Большинство гиперпараметров устанавливается экспериментальным методом, так как для разных задач они могут зависеть от разных факторов.

2.2 Обучающая выборка и аугментация данных

В начале разработки был выбран набор данных «Plant Pathology 2020»[3] на ресурсе Kaggle. Представленный на сайте в свободном доступе набор данных рассчитан на обучение нейронной сети, определяющей патологии растений по фотографии, что подходит для решения поставленной задачи и даст возможность практического применения нейронной сети.

В наборе данных содержатся изображения поврежденных и здоровых растений, принадлежащих к 4 классам:

- здоровый, или healthy;
- пораженный ржавчиной растений, или rust;
- пораженный паршой растений, или scab;
- имеющий множественные поражения, или multiple_diseases.

На основе этих изображений составится обучающая и тестовая выборка.

Данные для обучения модели загружены в облачное хранилище google drive.

Данные содержат 1820 изображений, которые были разделены на 1565 изображений в обучающей выборке и 255 изображений в валидационной выборке. Изображения хранятся в формате jpg с разрешением 2048 на 1365 пикселей.

Однако данный датасет содержит изображения яблоневых листьев, поэтому результаты обучения на этом наборе могут не подходить для анализа изображений растений на гидропонных фермах, предполагающих другие культуры растений.

Так же на ресурсе Kaggle были найдены наборы данных с изображениями листьев томата.

В наборе данных «Tomato Disease Multiple Sources» [6] содержатся изображения поврежденных и здоровых растений, принадлежащих к 10 классам:

- бактериальная пятнистость (bacterial spot);
- ранняя гниль (early blight);
- фитофтороз (late blight);
- кладоспориоз (leaf mold);
- септориоз (septoria leaf spot);
- паутинный клещ (spider mites);
- мишенеобразная пятнистость (target spot);
- вирус желтой курчавости листьев (yellow leaf curl virus);
- вирус мозаики (mosaic virus);
- здоровый (healthy).

Большинство из этих заболеваний развиваются в теплой и влажной среде поэтому могут представлять серьезную угрозу для растений на гидропонных фермах. Данные содержат 10000 изображений в обучающей выборке и 1000 изображений в валидационной выборке. Изображения хранятся в формате jpg с разрешением 296 на 296 пикселей.

В наборе данных «Tomato Leaves Dataset» [7] содержатся изображения поврежденных и здоровых растений, принадлежащих к 11 классам:

- бактериальная пятнистость (bacterial spot);
- ранняя гниль (early blight);
- фитофтороз (late blight);
- кладоспориоз (leaf mold);
- септориоз (septoria leaf spot);
- паутинный клещ (spider mites);
- мишенеобразная пятнистость (target spot);
- вирус желтой курчавости листьев (yellow leaf curl virus);
- вирус мозаики (mosaic virus);
- мучнистая роса (powdery mildew);
- здоровый (healthy).

Данные содержат около 25858 изображений в обучающей выборке и 6683 изображений в валидационной выборке. Однако изображения в наборе

взяты из разных источников, имеют разный фон и разный формат, так же данные в этом наборе уже аугментированы.

Несмотря на то, что последний рассмотренный набор содержит самое большое количество данных, точность обучения на нем может быть сомнительной из-за утечки данных. Причиной этого явления может быть то, что изображения разных классов приведены на различном фоне. Поэтому на результат классификации системы может влиять фон загружаемых изображений. Так же обучение на наборе данных такого размера занимает значительно больше времени, что осложняет тестирование и улучшение модели.

Подготовка данных предполагает преобразование размера изображений. Обучение модели на изображениях оригинального разрешения занимало бы слишком много времени и ресурсов, а так же содержало бы избыточные данные, поэтому изображения преобразуются к удобному для работы формату. Это повышает скорость обучения модели и не сказывается на точности, так как информация необходимая системе для классификации при таком сжатии не теряется.

Аугментация (увеличение) данных – метод, позволяющий сделать модель устойчивой к изменениям входных данных, таких как освещение, положение в кадре и ориентация. Так же это способ расширения обучающей выборки. Например, после зеркального преобразования изображения по горизонтали или вертикали, система будет воспринимать его как новое изображение, несмотря на то, что фотографию этого же листа уже обработала. Это позволит значительно расширить обучающую выборку и сделать систему устойчивой к данным снятым с разных сторон.

- Предлагается использовать такие параметры аугментации данных, как
- `shear_range` – производит сдвиг изображения;
 - `zoom_range` – масштабирует изображение;
 - `horizontal_flip` – отражает изображение по горизонтали;
 - `vertical_flip` – отражает изображение по вертикали.

Они дополняют набор данных в соответствии с тем, как могли бы выглядеть фотографии растений при разных условиях. Таким образом, они не смещают обучающую выборку и повышают надежность модели.

2.3 Реализация нейронной сети

Была разработана сверточная нейронная сеть, для которой была использована модель Sequential, которая позволяет последовательно добавлять слои в модель нейронной сети, что дает возможность достаточно легко дополнять и тестировать модель. Схема нейронной сети показана на рисунке 4.

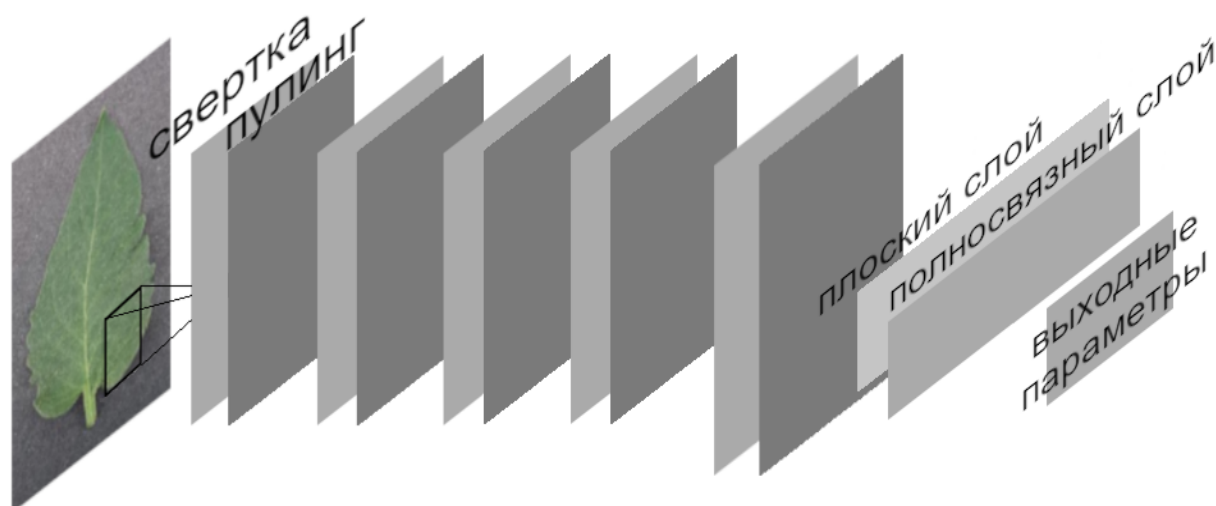


Рисунок 4 – Схема разработанной нейронной сети

На вход сети подаются приведенные к определенному размеру и предобработанные изображения. Размер изображений уменьшается с помощью функции `resize()`, изменяющей разрешение изображения до значений параметров, поданных в функцию. В модели используется размер 200 на 200 пикселей.

Изначальная модель нейронной сети содержит пять слоев двумерной свертки. Для создания сверточного слоя используется функция `Conv2D`. Функция `Conv2D` имеет следующие параметры: количество фильтров, размер

ядра свертки, входные данные и функцию активации, которая определяет выходное значение для каждого нейрона.

Для входного слоя исходной модели нейронной сети были выбраны следующие значения параметров: 32 ядра свертки размером 3*3.

В качестве функции активации для сверточного слоя используется функция ReLU.

Данные со сверточного слоя поступают на слой пулинга, реализуемый функцией MaxPooling2D().

Соответствующий входному слою нейросети код приведен ниже.

```
pattern.add(Conv2D(32, (3, 3), input_shape=input_shape))
pattern.add(Activation('relu'))
pattern.add(MaxPooling2D(pool_size=(2, 2)))
```

Три следующих сверточных слоя имеют 32 фильтра, последний слой имеет 64 фильтра, размер ядер свертки взят 3*3, функции пулинга на них идентичные, параметр pool_size равен (2, 2).

Выходные сигналы с последнего сверточного слоя преобразуются в одномерный массив размером 64 элемента, с использованием функции Flatten().

Эти сигналы поступают на полносвязный слой Dense, число нейронов в котором равно количеству выходных параметров последнего сверточного слоя.

Для уменьшения переобучения сети используется функция Dropout(), которая исключает определенный процент случайных нейронов во время обучения. Выбрано значение параметра этой функции равное 0.3, что означает, что процент исключаемых нейронов равен 30.

После чего идет выходной полносвязный слой нейронной сети, количество его выходных параметров равно числу классов и показывает результат классификации. После чего в свою очередь применяется функция Activation('softmax'). Соответствующий код приведен ниже.

```
pattern.add(Flatten())
pattern.add(Dense(64))
```

```
pattern.add(Activation('relu'))
pattern.add(Dropout(0.5))
pattern.add(Dense(10))# число классов
pattern.add(Activation('softmax'))
```

2.4 Подбор гиперпараметров

Были проведены тесты модели с разным количеством сверточных слоев. Для этого последовательно производилось обучение модели с двумя, тремя, четырьмя и пятью сверточными слоями.

Количество фильтров для первого слоя было установлено равным 32, для последнего равным 64, для слоев, добавляемых между ними, этот параметр так же устанавливался равным 32. Размер ядер свертки для всех слоев равен (3, 3).

Графики динамики процесса обучения продемонстрированы на рисунках 5 и 6. Первый график показывает изменения значения точности в зависимости от эпох обучения на тренировочной и валидационной выборке. Вторым графиком показывается значение функции потерь на тренировочной и валидационной выборке.

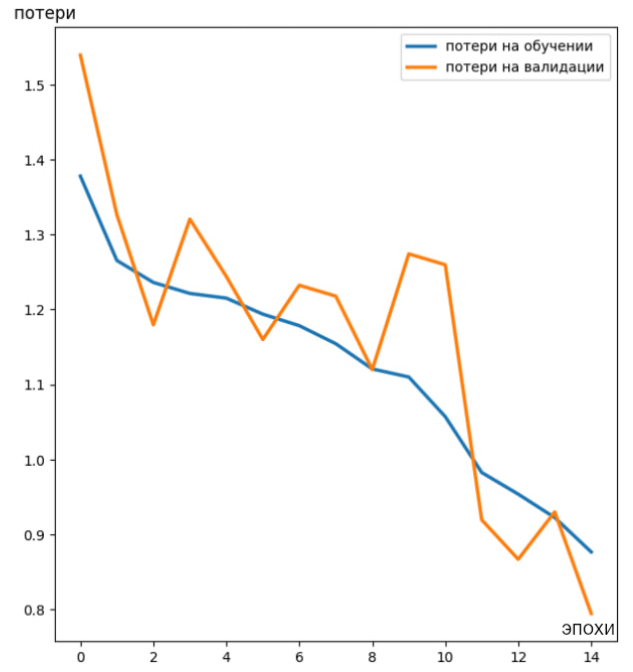
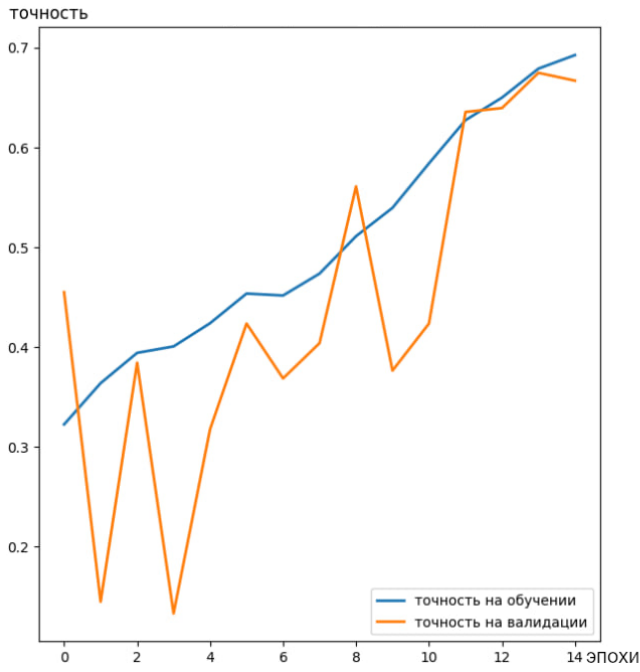


Рисунок 5 – Графики точности и потерь на 15 эпохах обучения модели с двумя сверточными слоями

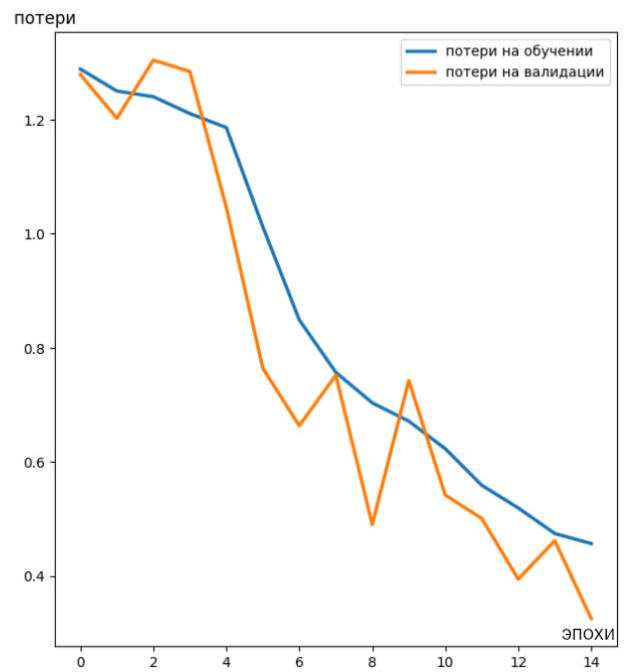
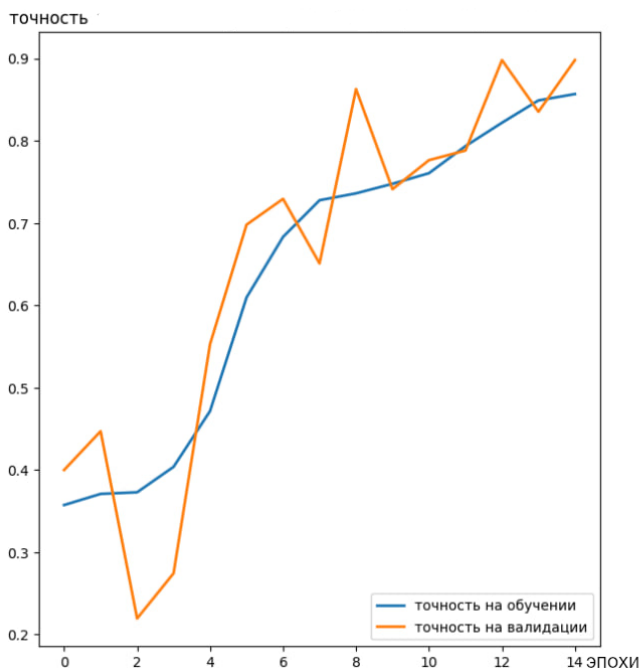


Рисунок 6 – Графики точности и потерь на 15 эпохах обучения модели с пятью сверточными слоями

Таблица 2 – точность модели при тестировании с разным количеством слоев.

Количество слоев	Точность на обучающей выборке	Точность на валидационной выборке
2	0.6923	0.6667
3	0.7832	0.7843
4	0.8135	0.7569
5	0.8568	0.8980

После чего были проведены тесты с изменением количества фильтров для сверточных слоев модели состоящей из пяти слоев (графики с результатами тестирования предоставлены на рисунке 7).

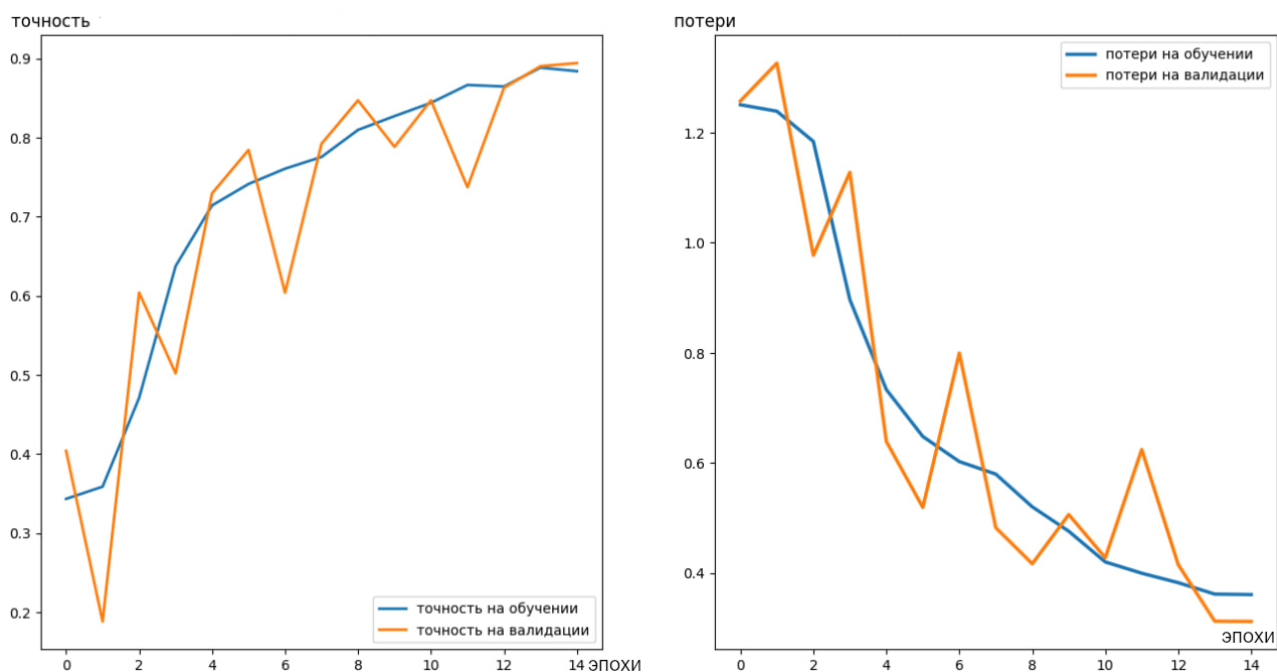


Рисунок 7 – Графики точности и потерь на 15 эпохах обучения модели с увеличенным последним слоем

Изменение количества фильтров последнего слоя до 128, привело к незначительному изменению точности (0.8839 на обучающей выборке и 0.8941 на валидационной выборке), при этом график её изменения стал более ровным. Последующее увеличение количества фильтров предпоследнего слоя до 64 привело к уменьшению точности до 0.8142 обучающей выборке и 0.8745 на

валидационной выборке (графики с результатами тестирования предоставлены на рисунке 8).

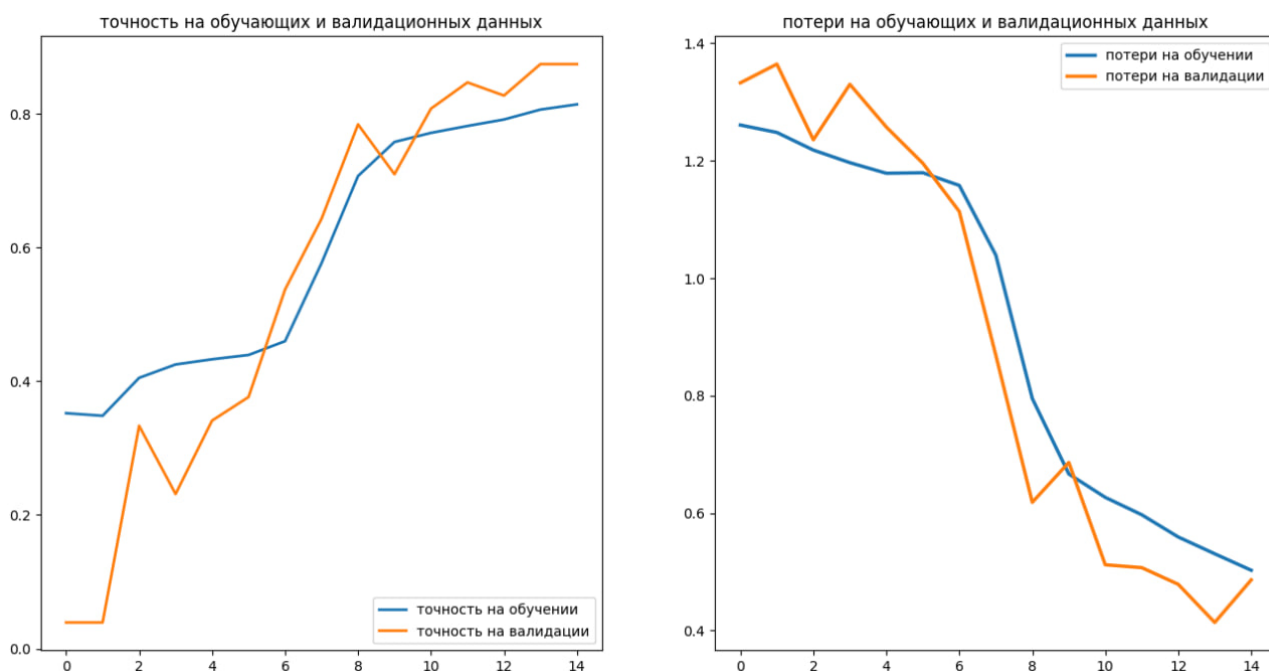


Рисунок 8 – Графики точности и потерь на 15 эпохах обучения модели с увеличенными двумя последними слоями

Увеличение размера ядра свертки первого сверточного слоя до (5, 5) привело к уменьшению времени обучения, но так же к уменьшению точности.

Изменение параметров разрешения предобработанных изображений влияет на скорость обучения модели, при установке этого параметра меньше чем 200 на 200 пикселей уменьшается точность модели.

Изменение параметра `batch_size` влияет на скорость обучения модели, при установке этого параметра больше чем 15 может уменьшаться точность модели на небольших наборах данных.

По результатам тестирования были установлены следующие параметры: изображения уменьшаются до размера 200 на 200 пикселей, `batch_size = 15`.

2.5 Обучение и тестирование нейронной сети

Разработка и тестирование системы проводилось в среде разработки Google Colaboratory, данные хранятся в облачном хранилище Google Drive. Google Colaboratory предоставляет больше ресурсов для вычислений чем имеется на доступном персональном компьютере, а так же при необходимости позволяет воспользоваться графическими ускорителями. После обучения нейронной сети с установленными финальными значениями гиперпараметров модель была скачана и запущена на локальном устройстве.

Обучение проводилось методом стохастического градиентного спуска (по батчам), в качестве набора данных для итоговой версии модели был выбран набор «Tomato Disease Multiple Sources». В качестве метрики качества выбраны такие параметры как доля верных ответов (accuracy), среднеквадратичная ошибка (mean squared error ,mse), Precision и recall, которые показывают долю истинно положительных результатов среди всех положительных результатов, и долю истинно положительных результатов среди всех объектов положительного класса.

Динамика процесса обучения приведена на рисунке 9. Точность доходит до 0.8699 на обучающей и 0.8450 на валидационной выборке, после чего держится примерно на одном уровне после 20 эпох обучения.

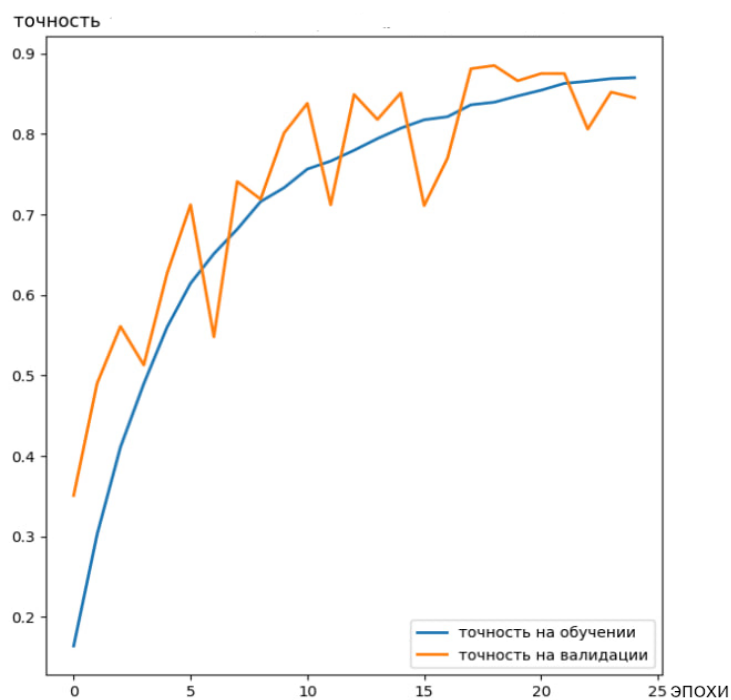


Рисунок 9 – График точности при обучении на 25 эпохах обучения

На рисунке 10 показаны графики потерь на валидационной и обучающей выборках. Если их графики начинают расходиться, возможен риск переобучения модели: модель запоминает все тестовые изображения, а не их признаки, после чего потери на тестовой выборке стремятся к 0, но на валидационной начинают расти. Как видно из графиков, вероятность переобучения возникает, но только на 25 эпохе.

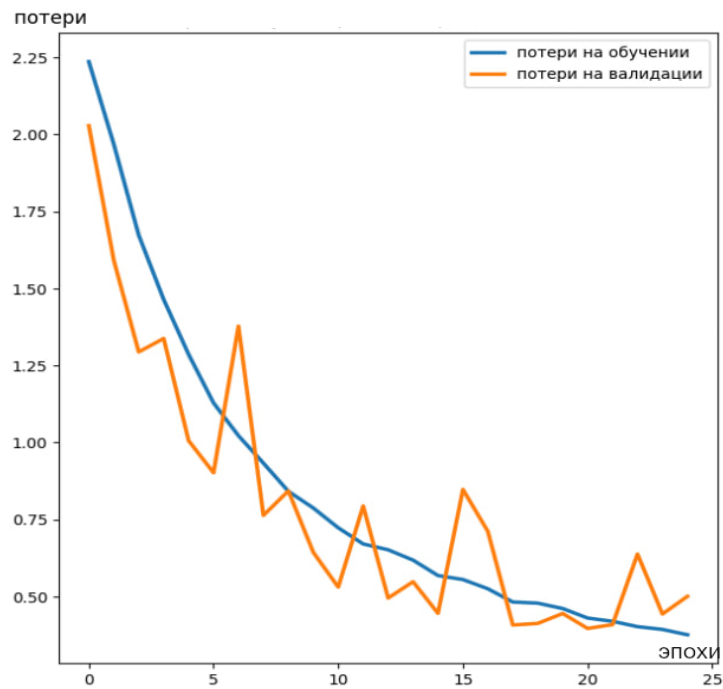


Рисунок 10 – График потерь при обучении на 25 эпохах обучения

На рисунке 11 показаны графики precision и recall, их поведение не противоречит полученному значению точности, из чего можно сделать вывод, что проблем с несбалансированностью набора данных не возникло.

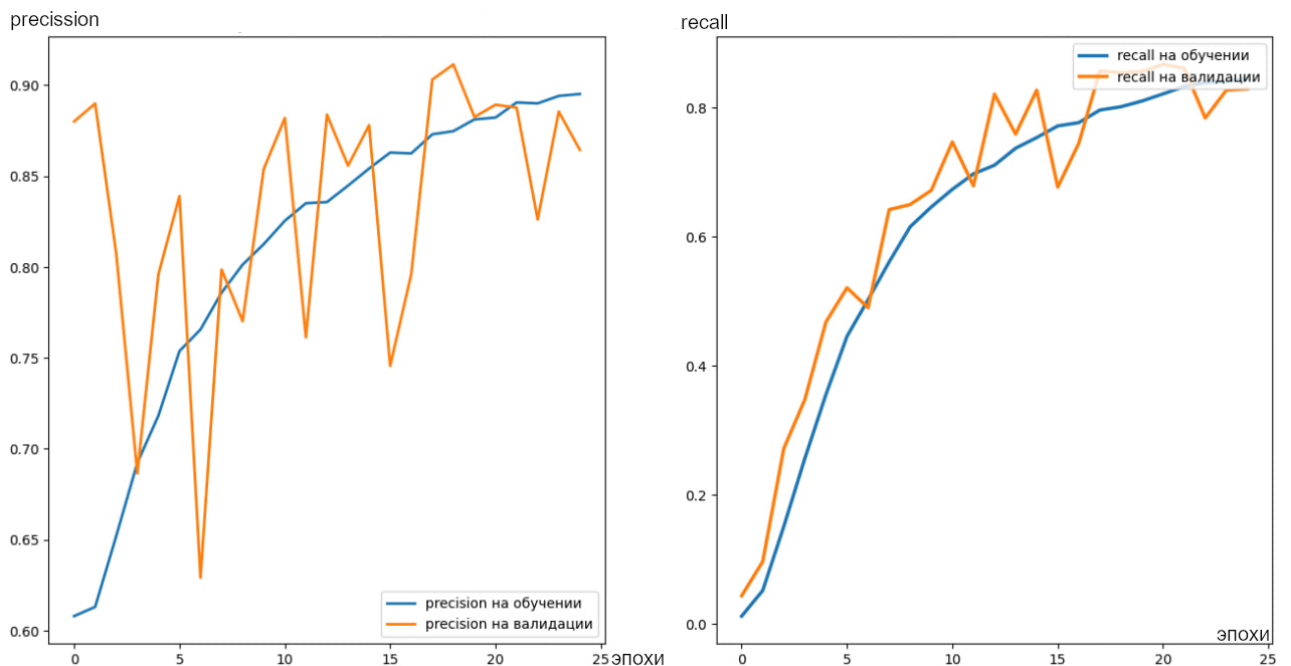


Рисунок 11 – Precision и recall на обучении и валидации

Так же было проведено тестирование на наборах данных «Tomato Leaves Dataset» и «plant-pathology 2020». Точность обучения на наборе данных «plant-pathology 2020» с теми же параметрами нейронной сети, достигала 89% после 15 эпох обучения, что говорит о возможности использования модели на разных наборах данных с разными классами и минимальной доработкой. Точность обучения на наборе данных Tomato Leaves Dataset достигала 81% что может быть связано с несбалансированностью набора данных.

2.6 Выводы по главе

Раздел посвящен вопросам проектирования и реализации модели нейронной сети.

Рассмотрены рассмотрен принцип работы сверточных нейронных сетей особенности их обучения.

Выполнен поиск обучающей выборки. Определены методы дополнительной обработки данных для повышения размера и надежности обучающей выборки которые можно использовать в обучении.

Реализована сверточная нейронная сеть, выполнен экспериментальных подбор гиперпараметров нейронной сети

Была разработана, обучена и протестирована модель сверточной нейронной сети, результаты тестирования на разных наборах данных показывают точность в районе 87-89%.

3 Проектирование и реализация системы

Для работы с пользователями предполагается автономное приложение, которое в ходе дальнейшей работы над проектом можно доработать до клиент-серверного приложения. Пользователь сможет загружать изображения, после чего нейронная сеть будет производить классификацию, затем пользователь будет получать результаты классификации.

Если кому-либо понадобится использовать систему для помощи в гидропонном выращивании, специфика работы будет зависеть от имеющихся у него инструментов, серверов, вида растений и других непредсказуемых факторов, делающих такой вариант более сложным и индивидуальным.

3.1 Функциональное проектирование системы

Актером будет являться пользователь: тот, кто в свободном режиме загружает изображения в модель. Прецеденты пользователя показаны в диаграмме на рисунке 12.



Рисунок 12 – Диаграмма прецедентов.

3.1.1 Прецедент «Классификация изображений»

Название прецедента: Классификация изображений.

Цель сценария: загрузить пакет изображений от пользователя.

Предусловия: открыто окно приложения, на устройстве пользователя находятся изображения в формате jpg.

Основной сценарий:

1. Пользователь нажимает кнопку «загрузить изображение».
2. Пользователь выбирает, какие изображения загрузить со своего устройства.
3. Открывается окно с результатами тестирования, содержащую таблицу с изображениями и их наиболее вероятными классами.

Постусловия: Пользователь получил информацию о своих изображениях

На рисунках 13 и 14 продемонстрированы диаграммы пригодности и последовательности для прецедента.

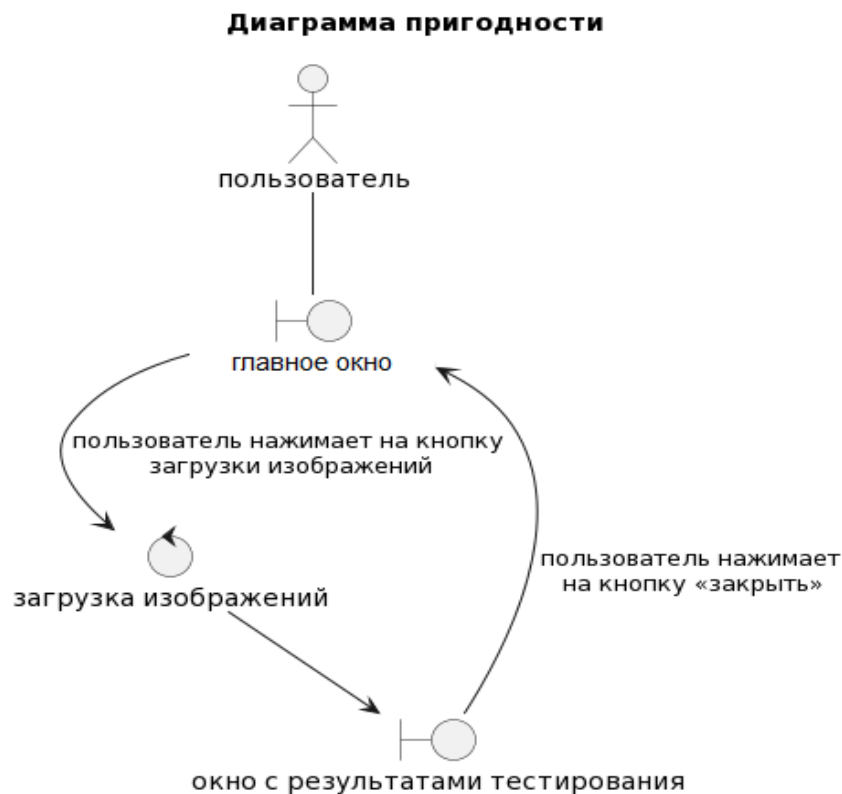


Рисунок 13 – Диаграмма пригодности прецедента «Загрузка изображений»

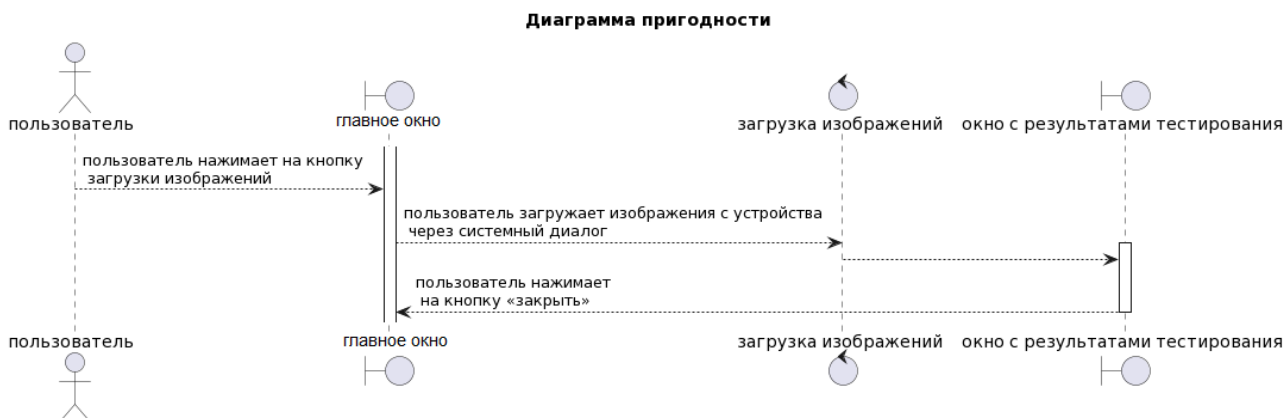


Рисунок 14 – Диаграмма последовательности прецедента «Загрузка изображений»

3.1.2 Прецедент «Просмотр данных изображения»

Название прецедента: Просмотр данных изображения.

Цель сценария: Узнать подробную информацию о конкретном изображении.

Предусловия: Открыто окно с результатами тестирования.

Основной сценарий:

1. Пользователь нажимает на название нужного изображения.
2. Открывается окно с результатами тестирования изображения.

Постусловия: Пользователь получил подробную информацию о тестировании изображения.

На рисунках 15 и 16 продемонстрированы диаграммы пригодности и последовательности для прецедента.



Рисунок 15 – Диаграмма пригодности прецедента «Просмотр данных изображения»

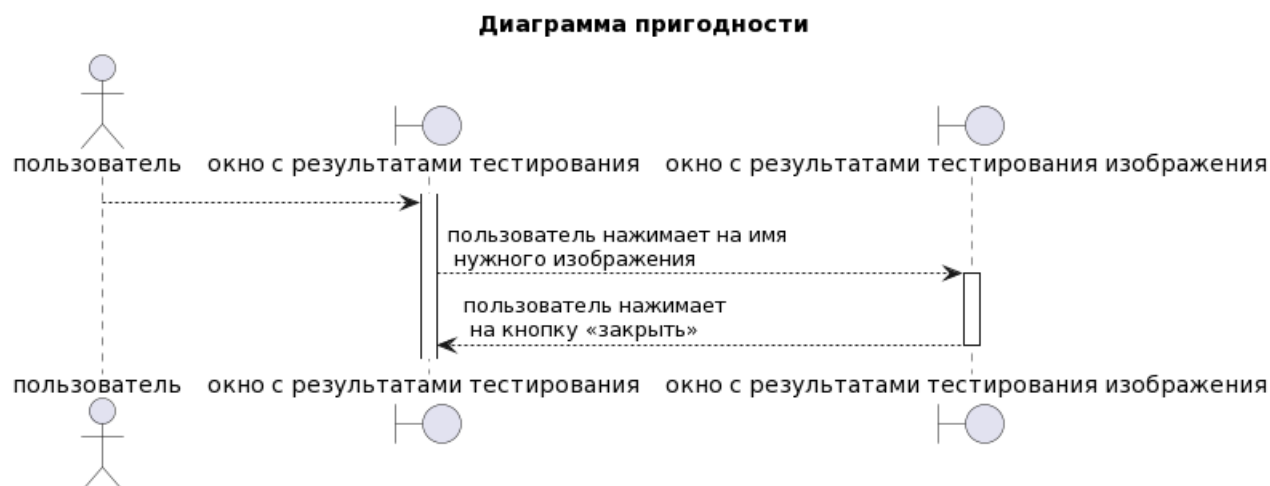


Рисунок 16 – Диаграмма последовательности прецедента «Просмотр данных изображения»

3.1.3 Прецедент «Пометка ошибки тестирования»

Название прецедента: Пометка ошибки тестирования.

Цель сценария: Отметить неверно классифицированное изображение.

Предусловия: Открыто окно с результатами тестирования изображения.

Основной сценарий:

1. Пользователь нажимает на кнопку «Пометка ошибки тестирования».
2. Изображение помещается в специальную папку.

На рисунках 17 и 18 продемонстрированы диаграммы пригодности и последовательности для прецедента

Диаграмма пригодности

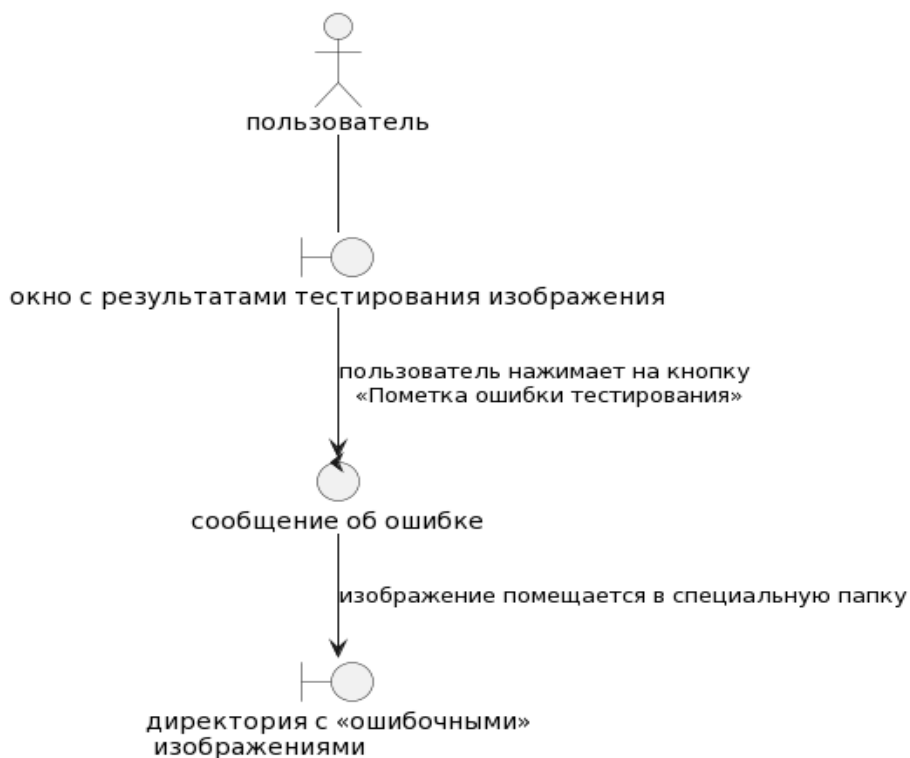


Рисунок 17 – Диаграмма пригодности прецедента «Отправка сообщения об ошибке»

Диаграмма пригодности

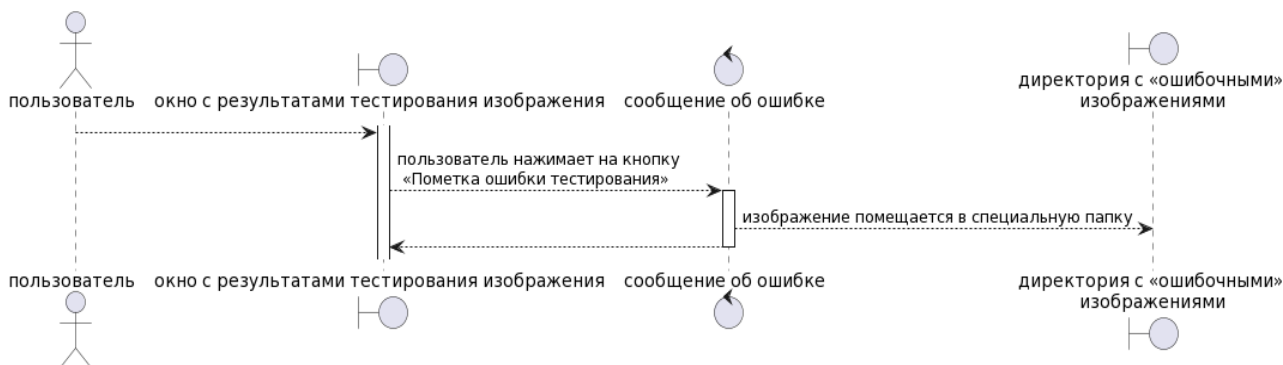


Рисунок 18 – Диаграмма последовательности прецедента «Отправка сообщения об ошибке»

3.1.4 Прецедент «Загрузка весов модели»

Приложение изначально будет иметь модель с загруженными весами, однако если пользователю понадобится воспользоваться другим набором весов,

например с результатами обучения на расширенном наборе данных, он может загрузить файл с весами обученной модели.

Название прецедента: Загрузка весов модели.

Цель сценария: загрузить готовый набор весов обученной нейронной сети.

Предусловия: открыто окно приложения, на устройстве пользователя находится файл с весами нейронной сети в формате «.h5».

Основной сценарий:

1. Пользователь нажимает кнопку «загрузка параметров нейронной сети».
2. Пользователь выбирает файл для загрузки со своего устройства.

Постусловия: В модель был загружен новый набор весов.

На рисунках 19 и 20 продемонстрированы диаграммы пригодности и последовательности для прецедента

Диаграмма пригодности

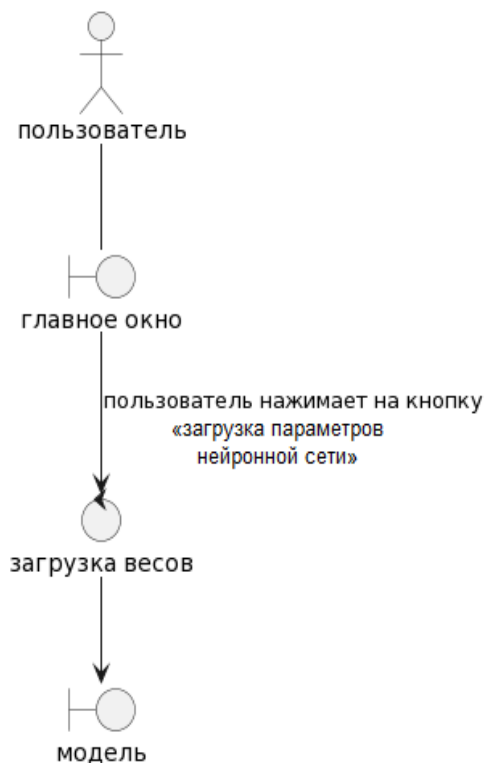


Рисунок 19 – Диаграмма пригодности прецедента «Загрузка весов модели»

Диаграмма пригодности

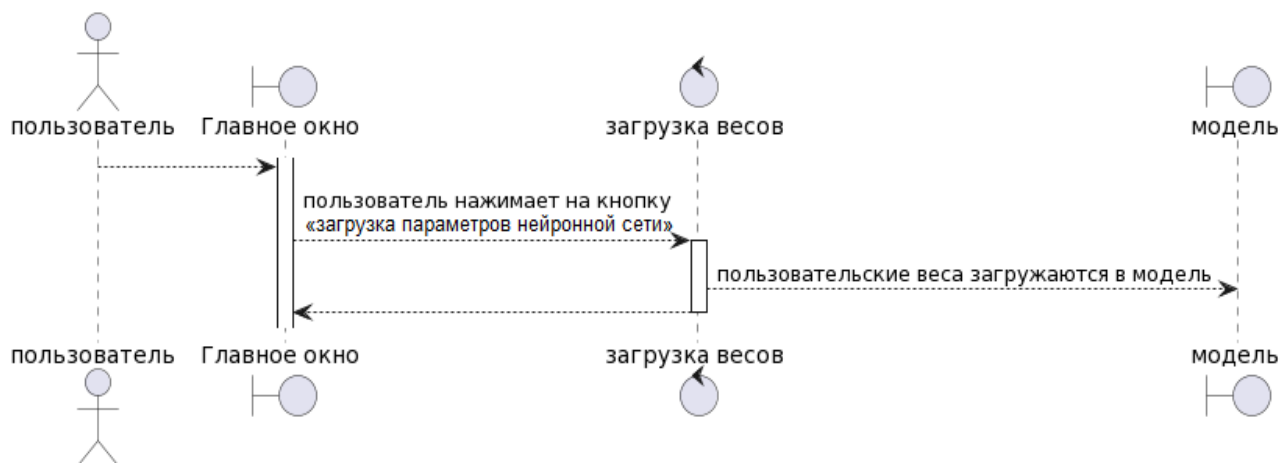


Рисунок 20 – Диаграмма последовательности прецедента «Отправка сообщения об ошибке»

3.2 Интерфейс системы

В основном окне приложения пользователь может загрузить изображение (Рисунок 21).

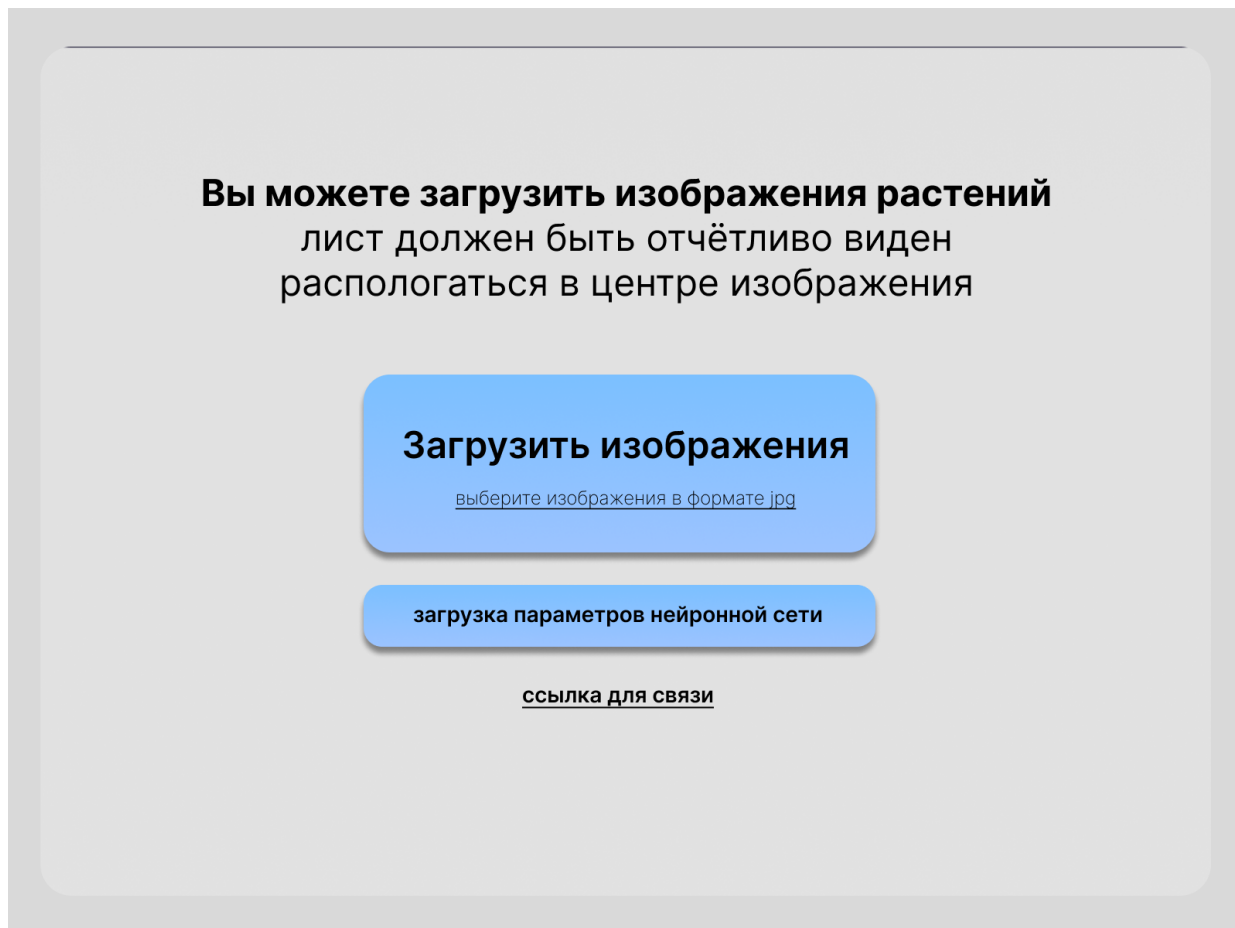


Рисунок 21 – Главное окно

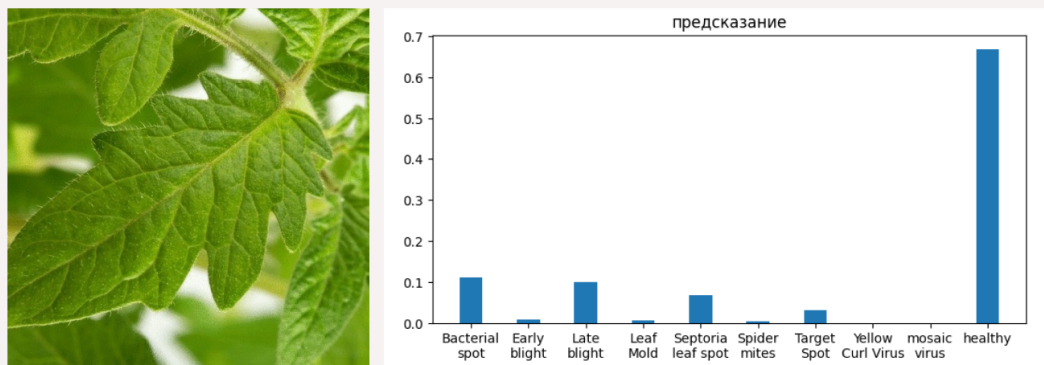
После обработки изображений открывается окно с таблицей, в которой для каждого изображения указан его наиболее вероятный класс (рисунок 22).

img1.jpg	healthy	●
img2.jpg	late blight	●
img3.jpg	leaf mold	●
img4.jpg	healthy	●
img5.jpg	target spot	●
img6.jpg	target spot	●
img7.jpg	healthy	●

заккрыть

Рисунок 22 – Результаты тестирования нескольких изображений

При нажатии на название изображения пользователь может открыть окно с подробной информацией для конкретного изображения с вероятностями принадлежности к каждому классу (рисунок 23). Если пользователь видит, что классификация произошла ошибочно, он может отправить сообщение об этом, нажав на соответствующую кнопку.



закреть

отправить сообщение об ошибке

Рисунок 23 – Подробные результаты тестирования изображения

Закрыв это окно пользователь вернётся на окно с результатами тестирования всех изображений.

ЗАКЛЮЧЕНИЕ

В ходе выполненной работы были проанализированы проблемы предметной области связанные с заболеваниями и повреждениями растений. Анализ работ по теме показал что для решения данной задачи наиболее эффективными являются сверточные нейронные сети.

Найдены и подготовлены наборы данных для обучения.

Рассмотрены принципы работы сверточных нейронных сетей, предложена архитектура нейронной сети и выполнен подбор гиперпараметров.

Обучена нейронная сеть, классифицирующая изображения растений по нескольким классам, показывающая точность (ассигасу) около 87-89%.

Разработано автономное приложение, позволяющее распознавать грибковые заболевания растений по фотографии.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Common Tomato Diseases / Illinois Extension // UIUC [Электронный ресурс]. URL: <https://extension.illinois.edu/blogs/good-growing/2019-07-09-common-tomato-diseases> (дата обращения: 9.06.2023).

2 Брагинский, М.Я. Оценка состояния растений с использованием сверточных нейронных сетей / М.Я. Брагинский, Д.В. Тараканов // Вестник киберленинки. – 2021. – № 1. – С. 41-49.

3 Plant Pathology 2020 : EDA + Models // Kaggle [Электронный ресурс]. URL: <https://www.kaggle.com/code/tarunpaparaju/plant-pathology-2020-eda-models/notebook> (дата обращения 10.12.2022).

4 Чирков, А.В. Выбор модели искусственной сверточной нейронной сети для определения заболеваний яблоневых деревьев / А.В. Чириков, Л.Г. Гагарина // Инновационные подходы к решению техника-экономических проблем : материалы международной конференции / Национальный исследовательский университет «МИЭТ» . – Москва, 2022. – С. 15–23.

5 Лисовский, М.Г. Решение задачи распознавания заболеваний растений на основе нейронных сетей / М.Г. Лисовский, Ю.Ю. Дюличева // Студенческая наука: актуальные вопросы, достижения и инновации : материалы II Международной научно-практической конференции / «Крымский федеральный университет имени В. И. Вернадского» . – Пенза, 2021. – С. 14–17.

6 Tomato Disease Multiple Sources / Kaggle [Электронный ресурс]. URL: <https://www.kaggle.com/datasets/cookiefinder/tomato-disease-multiple-sources> (дата обращения: 9.06.2023).

7 Tomato Leaves Dataset // Kaggle [Электронный ресурс]. URL: (дата обращения: 9.06.2023).

8 API Documentation // TensorFlow v2.12.0 // TensorFlow [Электронный ресурс]. URL: https://www.tensorflow.org/api_docs (дата обращения: 10.12.2022).

9 Team K. Keras documentation: Keras API reference [Электронный ресурс]. URL: <https://keras.io/api/> (дата обращения: 10.12.2022).

Приложение А

Код с реализацией нейронной сети

```
train_sample = 10000
validation_sample = 1000
epochs = 15
lot_size = 15 #batch_size
if K.image_data_format() != 'channels_first':
    input_shape = (image_width, image_height, 3)
else:
    input_shape = (3, image_width, image_height)

pattern = Sequential() # Создание модели
# Первый слой нейросети
pattern.add(Conv2D(32, (3, 3), input_shape=input_shape))
pattern.add(Activation('relu'))
pattern.add(MaxPooling2D(pool_size=(2, 2)))

# Второй слой нейросети
pattern.add(Conv2D(32, (3, 3)))
pattern.add(Activation('relu'))
pattern.add(MaxPooling2D(pool_size=(2, 2)))

#Третий слой нейросети
pattern.add(Conv2D(32, (3, 3)))
pattern.add(Activation('relu'))
pattern.add(MaxPooling2D(pool_size=(2, 2)))

#Четвертый слой нейросети
pattern.add(Conv2D(32, (3, 3)))
pattern.add(Activation('relu'))
pattern.add(MaxPooling2D(pool_size=(2, 2)))

#Пятый слой нейросети
pattern.add(Conv2D(64, (3, 3)))
pattern.add(Activation('relu'))
pattern.add(MaxPooling2D(pool_size=(2, 2)))

pattern.add(Flatten())
pattern.add(Dense(64))
pattern.add(Activation('relu'))
pattern.add(Dropout(0.3))
pattern.add(Dense(10))# число классов
pattern.add(Activation('softmax'))
```

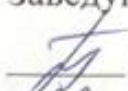
Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

УТВЕРЖДАЮ

Заведующий кафедрой



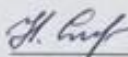
 О.В. Непомнящий

«22» 06 2023 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 – Информатика и вычислительная техника

Классификация изображений растений на основе нейронной сети

Руководитель	 22.06.23	доцент, канд. техн. наук	Н.Ю. Сиротинина
	подпись, дата	должность, ученая степень	инициалы, фамилия
Выпускник	 22.06.23		Г.Ю. Аникутин
	подпись, дата		инициалы, фамилия
Нормоконтролер	 22.06.23		Н.Ю. Сиротинина
	подпись, дата		инициалы, фамилия

Красноярск 2023