

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО
ITMO University

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА/GRADUATION THESIS

Разработка алгоритма автоматической классификации растений по фотографии
листа

Автор/ Author

Рогачёв Глеб Сергеевич

Направленность (профиль) образовательной программы/Major

Компьютерная фотоника и обработка изображений 2017

Квалификация/ Degree level

Бакалавр

Руководитель ВКР/ Thesis supervisor

Гуров Игорь Петрович, профессор, доктор технических наук, Университет ИТМО,
инженерно-исследовательский факультет, профессор (квалификационная категория " ординарный профессор")

Группа/Group

В3419

Факультет/институт/кластер/ Faculty/Institute/Cluster

инженерно-исследовательский факультет

Направление подготовки/ Subject area

12.03.03 Фотоника и оптоинформатика

Обучающийся/Student

| | |
|---------------------------|--|
| Документ подписан |  |
| Рогачёв Глеб Сергеевич | |
| 31.05.2021 | |

(эл. подпись/ signature)

Рогачёв Глеб
Сергеевич

(Фамилия И.О./ name
and surname)

Руководитель ВКР/
Thesis supervisor

| | |
|-------------------------|--|
| Документ подписан |  |
| Гуров Игорь Петрович | |
| 31.05.2021 | |

(эл. подпись/ signature)

Гуров Игорь
Петрович

(Фамилия И.О./ name
and surname)

**Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО
ITMO University**

**ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ /
OBJECTIVES FOR A GRADUATION THESIS**

Обучающийся / Student Рогачёв Глеб Сергеевич

Группа/Group В3419

Факультет/институт/кластер/ Faculty/Institute/Cluster инженерно-исследовательский факультет

Квалификация/ Degree level Бакалавр

Направление подготовки/ Subject area 12.03.03 Фотоника и оптоинформатика

Направленность (профиль) образовательной программы/Major Компьютерная фотоника и обработка изображений 2017

Специализация/ Specialization

Тема ВКР/ Thesis topic Разработка алгоритма автоматической классификации растений по фотографии листа

Руководитель ВКР/ Thesis supervisor Гуров Игорь Петрович, профессор, доктор технических наук, Университет ИТМО, инженерно-исследовательский факультет, профессор (квалификационная категория "ординарный профессор")

Срок сдачи студентом законченной работы до / Deadline for submission of complete thesis 31.05.2021

Техническое задание и исходные данные к работе/ Requirements and premise for the thesis

Разработать и реализовать в виде программной системы алгоритм классификации растений по изображению листа

Содержание выпускной квалификационной работы (перечень подлежащих разработке вопросов)/ Content of the thesis (list of key issues)

Введение;

информационный обзор, демонстрирующий существующие подходы к реализации классификатора;

исследование методов предварительной обработки входных данных в зависимости от выбранного классификатора;

разработка алгоритма классификации растений по изображению листа и его программной реализации;

анализ результатов практического испытания разработанной программной системы;

заключение;

список использованных информационных источников;

приложения (при необходимости).

Перечень графического материала (с указанием обязательного материала) / List of

graphic materials (with a list of required material)

Примеры подлежащих анализу изображений (обязательно).

Схемы и изображения, иллюстрирующие этапы обработки входных данных и работу всей программной системы в целом (обязательно).

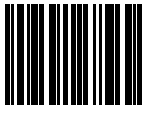
Исходные материалы и пособия / Source materials and publications

1. Гонсалес Р., Вудс Р. Цифровая обработка изображений. – Техносфера, 2012.
2. Bradski G., Kaehler A. Learning OpenCV 3. Computer vision in C++ with the OpenCV library. - Gravenstein Highway North, Sebastopol, CA, USA: O'Reilly Media, 2016.
3. Рейнхард Клетте. Компьютерное зрение. Теория и алгоритмы. М.: Спрингер, 2019.

Дата выдачи задания/ Objectives issued on 24.05.2021

СОГЛАСОВАНО / AGREED:

Руководитель ВКР/
Thesis supervisor

| | |
|-------------------------|--|
| Документ подписан |  |
| Гуров Игорь Петрович | |
| 24.05.2021 | |

(эл. подпись)

Гуров Игорь
Петрович

Задание принял к
исполнению/ Objectives
assumed by

| | |
|---------------------------|--|
| Документ подписан |  |
| Рогачёв Глеб Сергеевич | |
| 25.05.2021 | |

(эл. подпись)

Рогачёв Глеб
Сергеевич

Руководитель ОП/ Head
of educational program

| | |
|-------------------------|--|
| Документ подписан |  |
| Гуров Игорь Петрович | |
| 28.05.2021 | |

(эл. подпись)

Гуров Игорь
Петрович

**Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО
ITMO University**

**АННОТАЦИЯ
ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ /
SUMMARY OF A GRADUATION THESIS**

Обучающийся/ Student

Рогачёв Глеб Сергеевич

Наименование темы ВКР / Title of the thesis

Разработка алгоритма автоматической классификации растений по фотографии листа

Наименование организации, где выполнена ВКР/ Name of organization

Университет ИТМО

**ХАРАКТЕРИСТИКА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ/
DESCRIPTION OF THE GRADUATION THESIS**

1. Цель исследования / Research objective

Разработка высокоточного алгоритма классификации растений по изображению листа

2. Задачи, решаемые в ВКР / Research tasks

1) Проведение анализа публикаций и разработок по теме работы; 2) исследование характерных признаков листа, которые могут быть выделены на изображении и использованы для разработки классификатора; 3) рассмотрение существующих алгоритмов классификации изображений, их сравнение и выбор наиболее соответствующего условиям задачи; 4) изучение особенностей выбранного для реализации метода и определение наиболее подходящей конфигурации для разработки классификатора; 5) оценка результатов разработанного классификатора и сравнение его с одним из рассмотренных алгоритмов; 6) обеспечение работы на мобильных устройствах; 7) формулирование выводов по проделанной работе.

3. Краткая характеристика полученных результатов / Short summary of results/conclusions

Был проведен анализ публикаций и разработок по теме работы, исследованы характерные признаки листьев, по которым может быть осуществлена идентификация растений, проведено сравнение методов классификации. В результате был выбран путь разработки классификатора на основе свёрточной нейронной сети MobileNetV2. Разработанный классификатор был реализован с помощью нейросетевой библиотеки Keras и показал высокие результаты точности предсказаний, на базе него разработаны мобильное приложение и Telegram-бот для осуществления моментальной идентификации растения.

4. Наличие публикаций по теме выпускной работы/ Have you produced any publications on the topic of the thesis

5. Наличие выступлений на конференциях по теме выпускной работы/ Have you produced any conference reports on the topic of the thesis

6. Полученные гранты, при выполнении работы/ Grants received while working on the thesis

7. Дополнительные сведения/ Additional information

Обучающийся/Student

| | |
|---------------------------|--|
| Документ подписан |  |
| Рогачёв Глеб Сергеевич | |
| 31.05.2021 | |

(эл. подпись/ signature)

Рогачёв Глеб
Сергеевич

(Фамилия И.О./ name
and surname)

Руководитель ВКР/
Thesis supervisor

| | |
|-------------------------|--|
| Документ подписан |  |
| Гуров Игорь Петрович | |
| 31.05.2021 | |

(эл. подпись/ signature)

Гуров Игорь
Петрович

(Фамилия И.О./ name
and surname)

СОДЕРЖАНИЕ

| | |
|--|----|
| СОДЕРЖАНИЕ | 6 |
| ВВЕДЕНИЕ..... | 8 |
| ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ | 10 |
| 1.1. Анализ опубликованных работ | 10 |
| 1.2. Рассмотрение признаков, которые могут быть выделены на фотографии листа растения | 12 |
| 1.2.1. Форма листовой пластинки | 12 |
| 1.2.2. Тип листа | 13 |
| 1.2.3. Поверхность листа..... | 14 |
| 1.3. Методы реализации классификатора | 15 |
| 1.3.1. Методы сравнения с образцом..... | 16 |
| 1.3.2. Статистические методы | 19 |
| 1.3.3. Полносвязные нейронные сети | 19 |
| 1.3.4. Свёрточные нейронные сети | 21 |
| 1.3.5. Сравнение рассмотренных методов классификации | 23 |
| 1.4. Выводы | 23 |
| ГЛАВА 2. ИССЛЕДОВАНИЕ ВЫБРАННОГО МЕТОДА РЕАЛИЗАЦИИ КЛАССИФИКАТОРА И ПРЕДВАРИТЕЛЬНОЙ ОБРАБОТКИ ВХОДНЫХ ДАННЫХ | 25 |
| 2.1. Выбор архитектуры нейронной сети | 25 |
| 2.2. Подробное рассмотрение выбранной нейронной сети..... | 27 |
| 2.3. Выбор способа обучения нейронной сети | 28 |
| 2.4. Выбор методов подготовки исходных данных..... | 29 |
| 2.5. Выбор программных средств для реализации классификатора | 32 |
| 2.6. Выводы | 32 |
| ГЛАВА 3. РАЗРАБОТКА АЛГОРИТМА КЛАССИФИКАЦИИ РАСТЕНИЙ ПО ИЗОБРАЖЕНИЮ ЛИСТА И ЕГО ПРОГРАММНОЙ РЕАЛИЗАЦИИ | 34 |
| 3.1. Обработка данных для обучения нейронной сети | 34 |

| | |
|--|----|
| 3.2. Программная реализация алгоритма классификации растений по изображению листа..... | 36 |
| 3.3. Анализ результатов работы алгоритма | 38 |
| 3.4. Сравнение результатов разработанного классификатора с классификатором на основе метода IDSC | 40 |
| 3.5. Разработка программного обеспечения для осуществления классификации с мобильных устройств | 41 |
| 3.5.1. Мобильное приложение..... | 42 |
| 3.5.2. Telegram-бот..... | 45 |
| 3.6. Выводы | 47 |
| ЗАКЛЮЧЕНИЕ | 48 |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ | 50 |
| ПРИЛОЖЕНИЕ А..... | 53 |

ВВЕДЕНИЕ

В современном мире компьютерные технологии имеют очень широкое распространение и развиваются высокими темпами. Почти все сферы человеческой деятельности так или иначе связаны с компьютерными технологиями, что позволяет автоматизировать быстрое и эффективное выполнение большого количества задач. Последние десятилетия одной из наиболее активно развивающихся областей является область искусственного интеллекта. Этим понятием обозначают широкую сферу, состоящую из большого количества задач, решение которых принято считать прерогативой человека, таких как: анализ текста, распознавание и обработка аудиоинформации, обучение, компьютерное зрение, игры и т.д. В частности, рассматриваемая в данной работе задача определения вида растения по фотографии его листа с помощью информационных систем, относится к области компьютерного зрения.

Задача идентификации растения по его листу стандартными методами является довольно сложной, т.к. для этого необходимо владеть большим количеством информации, такой как: жизненная форма, к которой данное растение относится (кустарник, дерево, полукустарник, трава и т.д.), внешние характеристики листа (форма и членение долей, тип поверхности), типы края и основания, среда обитания и т.д. Обывателю успешно выполнить все эти действия и прийти к однозначному заключению представляется трудным. Для специалиста в данной области задача выглядит проще, если ему удаётся определить все основные характерные признаки и идентифицировать растение с помощью какой-либо таксономической базы данных, но даже в этом случае требуется затратить немало времени. Применение компьютерных технологий является наилучшим путём для решения задачи идентификации, поскольку позволяет быстро, эффективно и с высокой точностью установить, к какому виду относится растение по фотографии его листа.

Данные классификаторы наиболее удобны и полезны для использования в следующих целях:

- идентификация растений при исследовании и контроле состояния флоры определённых регионов;
- определение безопасности растения для человека;
- применение в образовательных учреждениях, таких как: школы, музеи, высшие учебные заведения;
- обработка больших объёмов данных, связанных с ботаникой.

Перечисленные области применения можно определить как научно-исследовательские, требующие от классификатора быстрой и точной работы.

Целью данной работы является разработка высокоточного алгоритма классификации растений по изображению листа.

В соответствии с целью сформулированы следующие **задачи**:

- 1) провести анализ публикаций и разработок по теме работы;
- 2) исследовать характерные признаки листа, которые могут быть выделены на изображении и использованы для разработки классификатора;
- 3) рассмотреть существующие алгоритмы классификации изображений, провести их сравнение и выбрать наиболее соответствующий условиям задачи;
- 4) изучить особенности выбранного для реализации метода и определить наиболее подходящую конфигурацию для разработки классификатора;
- 5) оценить результаты работы разработанного классификатора и провести его сравнение с одним из рассмотренных алгоритмов;
- 6) обеспечить работу на мобильных устройствах;
- 7) сформулировать выводы по проделанной работе.

ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Анализ опубликованных работ

Проведённый анализ опубликованных исследований и разработанных систем классификации показал, что на данный момент разработано несколько мобильных приложений для смартфонов на базе ОС Android и iOS, направленных на помощь пользователям в идентификации растений, в которых данная задача решается путём ручного поиска по встроенной базе видов указанием характерных для интересующего растения признаков (т.е. такие приложения представляют собой электронные версии определителей растений), либо обсуждением наименования растения с другими пользователями сервиса. Однако данные пути идентификации растений не позволяют автоматически, без участия пользователя однозначно установить вид растения за короткое время.

На фоне всех доступных мобильных приложений выделяются сервисы NatureId [1] и Leafsnap [2], позволяющие по фотографии растения в течение 5 секунд получить от 1 до 10 наиболее вероятных наименований видов в NatureId и от 5 до 30 в случае Leafsnap, к которым может принадлежать идентифицируемое растение. Анализ отзывов в магазине приложений и собственного опыта использования приложений показал, что данные сервисы имеют большой процент неверных предсказаний. На фотографиях пяти домашних растений и нескольких листьев, отснятых на белом фоне, сервис NatureId дал ошибочные ответы в половине случаев. Результаты сервиса Leafsnap оказались несколько лучше - приложение примерно в 60% случаев выдавало длинный список возможных видов (20-30 наименований), среди которых находился правильный класс растения. Также были протестированы несколько других приложений (Plantsnap [3], Plant Lens [4]), но их результаты оказались значительно хуже вышеперечисленных.

Доступные на данный момент мобильные приложения для идентификации растений предназначены для широкого использования, но при этом не

обеспечивают достаточно высокую точность распознавания. Следовательно, они не могут быть применены в научно-исследовательских целях.

Недостаточно высокая точность идентификации растений мобильными приложениями может быть обусловлена тем, что они располагают чрезмерно большим количеством распознаваемых классов растений, среди которых алгоритму становится проблематично установить вид растения даже среди 20-30 наиболее вероятных вариантов.

В исследовании [5] группа американских учёных разработала алгоритм идентификации растений, работа которого была проверена на трёх отдельных наборах данных. Ими были использованы следующие наборы:

- древесные растения округа Балтимор-Вашингтон (7481 лист, 245 видов);
- растения острова Пламмерс (5013 листьев, 157 видов);
- листья деревьев Центрального парка Нью-Йорка (4320 фотографий, 144 вида).

Обработка исходных наборов данных проводилась следующим образом: лист на каждой фотографии окрашивался в белый цвет, фон – в чёрный. Окрашивание проводилось с помощью адаптированного для цветных изображений статистического алгоритма Ожидания-Максимизации. Затем для каждого изображения был сформирован набор признаков путём выделения на контурах листьев характерных особенностей, свойственных данным видам растений методом IDSC (Inner Distance Shape Context [6]). Идентификация растений проводилась путём поиска наиболее схожих по выделенным признакам видов методом KNN (K-Nearest Neighbors – К-ближайших соседей [7]). Данный метод позволил достичь точности в 90% среди 10 наиболее вероятных предсказаний на наборе данных острова Пламмерс, состоящем не только из представителей деревьев, но и трав, кустарников и т.д. Данная разработка представляет собой наиболее точную систему из всех

рассмотренных, и впоследствии нашла применение в Национальном музее естественной истории Смитсоновского института. На данный момент эта система устарела в связи с тем, что со времени её публикации были разработаны новые алгоритмы и методы построения классификаторов, позволяющие достичь более высокой точности. Обзор классификаторов приводится в пункте 1.3.

1.2. Рассмотрение признаков, которые могут быть выделены на фотографии листа растения

Использование ботанического определителя для классификации растений подразумевает использование информации не только о внешнем виде листа, а ещё и информацию о морфологических признаках, таких как: корень, стебель, место обитания, покров растения, наличие цветов, плодов, цвет составных частей растения, размеры. Так как в данной работе разрабатывается система, способная осуществлять классификацию в случае, когда в распоряжении пользователя есть только фотографии листа и отсутствует вся остальная информация о растении, необходимо установить основные признаки, которые могут быть выявлены на изображении листа. Ниже перечислены некоторые признаки, на основании которых сервис Plantarium [8] позволяет осуществить идентификацию растения с помощью базы, размещенной на данном сервисе. Рассмотрение этих признаков поможет определить требования к классификатору.

1.2.1. Форма листовой пластинки

Листовая пластинка – это основная часть листа, которая выполняет основные функции, например, фотосинтез и газообмен, её форма может быть разделена на три составляющие: верхушка, основание и край. Для выделения данных признаков достаточно изображения, содержащего информацию о контуре листа (можно использовать изображение в оттенках серого либо черно-белое изображение, разделённое на области листа и фона). Ниже

приведен рисунок 1.1, наглядно демонстрирующий различные вариации составляющих листовой пластинки.

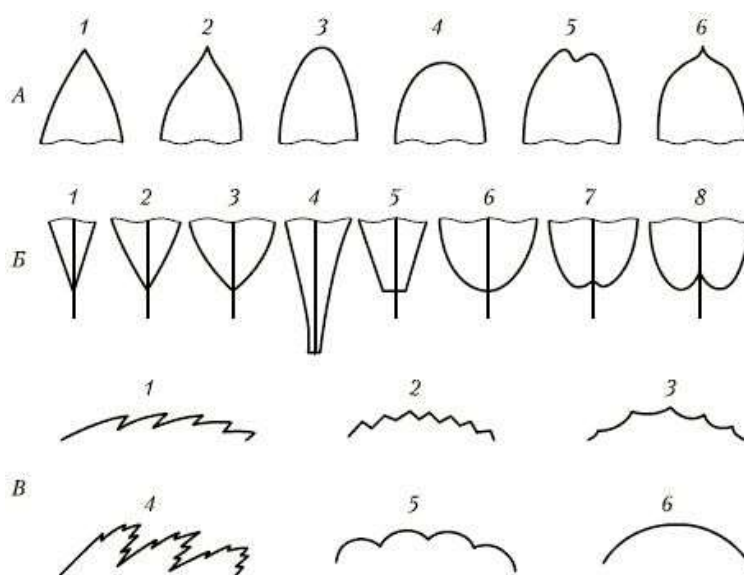


Рисунок 1.1 – Примеры разнообразных форм верхушки, основания и края листовой пластинки [9], где :А – верхушка: 1 – острая; 2 – оттянутая; 3 – туповатая; 4 – округлая; 5 – выемчатая; 6 – с остроконечием; Б – основание: 1 – узкоклиновидное; 2 – клиновидное; 3 – ширококлиновидное; 4 – нисбегающее; 5 – усеченное; 6 – округлое; 7 – выемчатое; 8 – сердцевидное; В – край: 1 – пильчатый; 2 – зубчатый; 3 – выемчатый; 4 – двоякопильчатый; 5 – городчатый; 6 – цельный)

1.2.2. Тип листа

Данный признак подразумевает разделение на два класса: простые и сложные листья. Простым листом называется тот, который имеет единую листовую пластинку (в некоторых случаях она может быть сильно рассечена). Сложными листьями называют те, листовая пластинка которых состоит из нескольких обособленных долей (листочков, сегментов, в некоторых случаях они могут иметь собственные черешки). Ниже на рисунке 1.2 приведен пример простых и сложных листьев. В случае сложного листа возможно, что для идентификации будет предоставлено изображение не всего листа, а его части

в виде отдельного листика, таким образом, классификатору необходимо идентифицировать оба возможных варианта как представителей одного вида.



Рисунок 1.2 – Примеры простого (лист берёзы простой) и сложного (лист рябины обыкновенной) листьев [10]

1.2.3. Поверхность листа

На поверхности листа находится большое число признаков, которые должны быть учтены в процессе классификации, таких как: жилкование листа (пример, демонстрирующий разнообразие данного признака, приведен на рисунке 1.3), тип его поверхности (глянцевый/матовый, гладкий/с опушением), цвет листа, наличие вкраплений и т.д. Для выделения наибольшего числа признаков необходимо, чтобы классификатор поддерживал работу с цветным изображением, учитывал возможную вариативность в распределении этих признаков внутри представителей одного вида.



Рисунок 1.3 – Виды жилкования растений [11]

1.3. Методы реализации классификатора

Традиционно в области компьютерного зрения под классификатором понимается устройство или алгоритм, позволяющий по характерным признакам объекта, выявленным на входных данных, установить, к какому классу данный объект относится. В общем виде можно выделить несколько стадий работы классификатора:

- передача исходного изображения в алгоритм;
- предварительная обработка для приведения изображения к стандарту классификатора;
- выявление наиболее значимых признаков;
- предсказание вероятностей принадлежностей к классам на основании выделенных признаков.

Ниже рассмотрены методы, наиболее часто используемые для построения классификаторов, разделенные на три группы: методы сравнения с образцом, статистические методы и нейронные сети [12].

1.3.1. Методы сравнения с образцом

Методы данной группы основаны на классификации векторов признаков по ближайшему среднему значению и по расстоянию до ближайшего соседнего значения.

- Классификация по ближайшему среднему значению

В данном методе реализации классификатора идентификация объекта осуществляется путём нахождения класса, чьё среднее значение наиболее близко к вектору признаков идентифицируемого объекта. Ниже на рисунке 1.4 схематически изображено представление классов, используемое в данном методе.

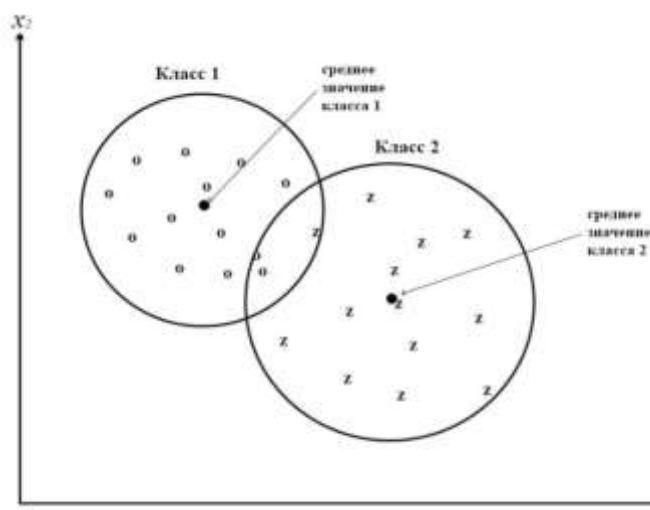


Рисунок 1.4 – Метод классификации по ближайшему среднему значению [12]

У данного метода есть значительный недостаток, проявляющийся в случае, если некоторые объекты класса ближе к среднему значению другого класса, пример такого случае продемонстрирован на рисунке 1.5, где красным цветом обозначены векторы признаков объектов, в действительности принадлежащих классу 3, но которые должны быть распознаны классификатором как элементы класса 1, т.к. ближайшим средним является среднее класса 1.

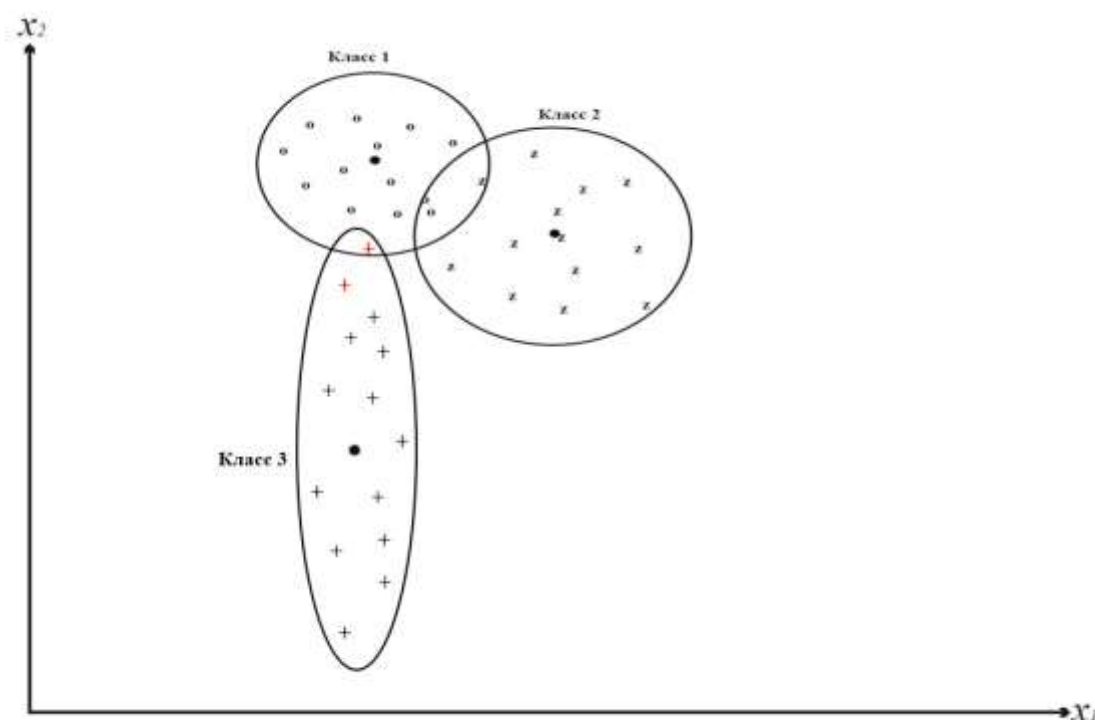


Рисунок 1.5 – Случай ошибочной классификации методом ближайшего среднего

- Классификация методом ближайшего соседа

Для исключения возникновения случаев, описанных в предыдущем абзаце, следует выполнять классификацию путём присвоения идентифицируемому объекту того же класса, к которому принадлежит ближайший объект, называемый ближайшим соседом. На рисунке 1.6 схематично изображен пример классификации методом ближайшего соседа. В данном примере необходимо классифицировать объект X, для этого определяются расстояния до ближайших соседних объектов (из классов **K1** и **K3**), в результате чего идентифицируемому объекту присваивается тот же класс, что и у ближайшего соседнего наблюдения (класс **K1**). Данный метод допускает и несколько другой способ классификации, в ходе которого определяются несколько ближайших соседей и в результате объекту присваивается класс, который наиболее часто встречается среди соседей.

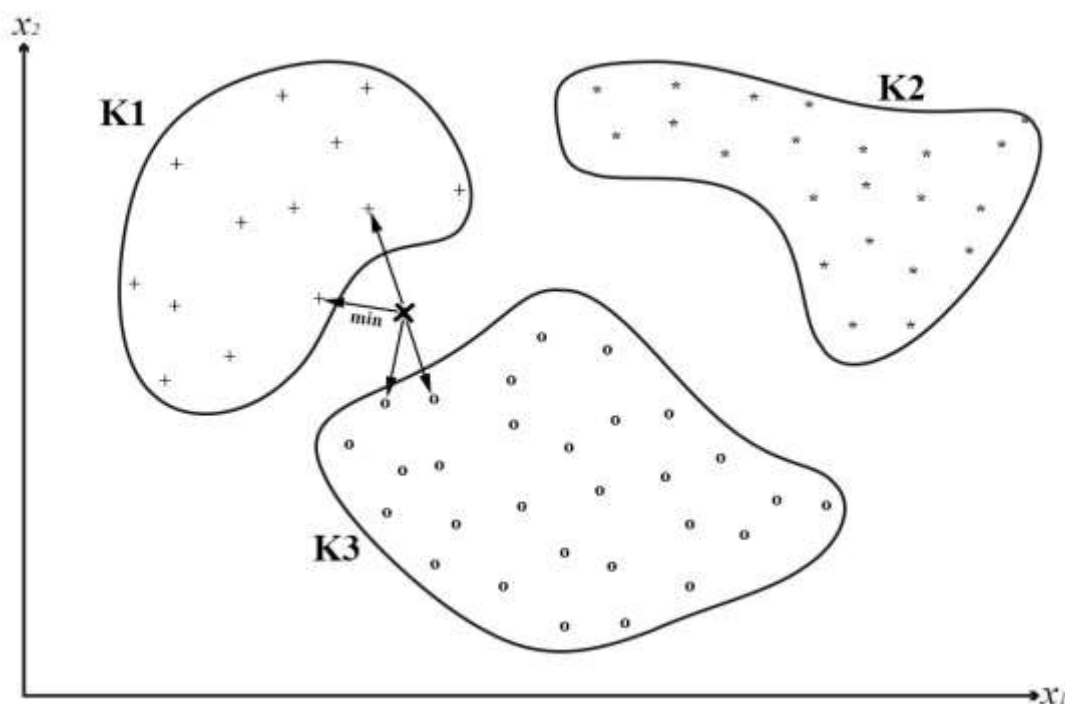


Рисунок 1.6 – Метод ближайшего соседа

Среди преимуществ данного подхода можно выделить простоту реализации, возможность внесения новых векторов признаков в базу данных в любое время и необходимость указания лишь двух параметров для реализации метода (количества ближайших соседей и метода расчёта расстояний) [13]. У данного метода имеются следующие недостатки: некоторые качественные признаки трудно представить в численном виде без потери информации, что приводит к ошибкам в классификации; при увеличении количества учитываемых признаков, точность алгоритма ощутимо снижается, это обусловлено тем, что в многомерном пространстве становится сложно оценивать расстояния до ближайших соседей в каждом из измерений; также скорость работы данного метода снижется с увеличением объема данных.

1.3.2. Статистические методы

Статистические методы основаны на вычислении вероятностей принадлежности идентифицируемого объекта к различным классам. Примером этой группы служит байесовский метод принятия решения. Данный метод основан на теореме Байеса и определении априорных вероятностей, то есть вероятность исходов или принадлежности объекта определенному классу изменяется после получения новых экспертных оценок (подтверждения наличия новых признаков). Появление того или иного образа является случайным событием и вероятность этого события можно описать с помощью закона распределения вероятностей многомерной случайной величины. Зная элементы обучающей выборки, можно восстановить вероятностные характеристики этой среды. Байесовский классификатор на основе наблюдаемых признаков относит объект к классу, к которому этот объект принадлежит с наибольшей вероятностью.

Недостаток данного метода заключается в том, что в он плохо применим для решения сложных задач, в которых размерность признаков может быть высокой и для оценки вероятности многомерной случайной величины может не хватить числа наблюдений из списка с данными [12].

1.3.3. Полносвязные нейронные сети

Полносвязные нейронные сети – математическая модель, построенная по образу биологических нейронных сетей, состоящая из искусственных нейронов. Искусственный нейрон (изображён на рисунке 1.7) представляет собой аналог биологического нейрона, активация которого происходит при превышении входящим сигналом порогового значения. В математическую модель нейрона входят следующие составляющие: сумматорная функция, осуществляющая взвешенное сложение всех входных сигналов и активационная функция (определяющая порог активации), результат которой подаётся на выход нейрона.

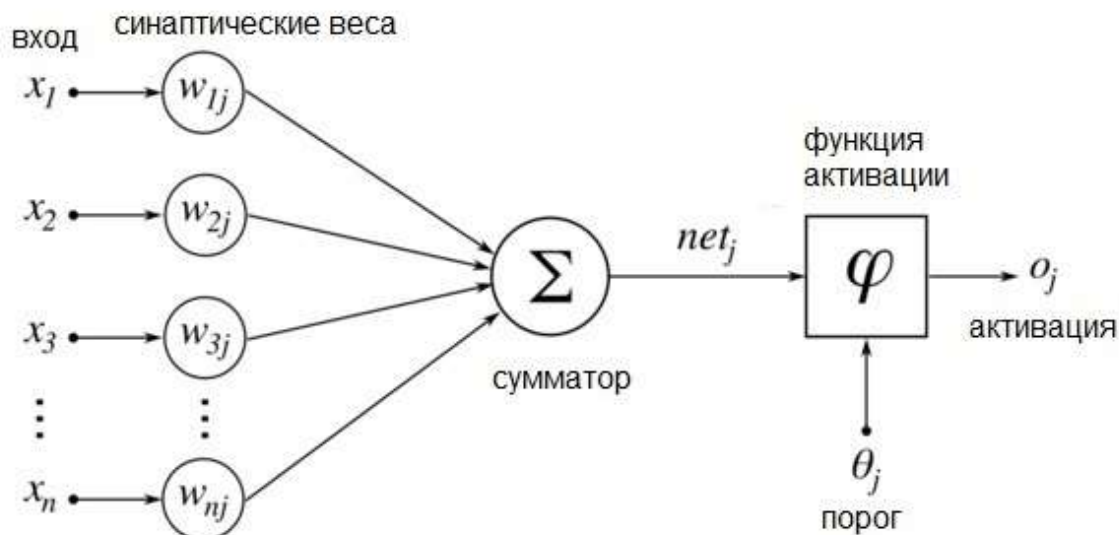


Рисунок 1.7 – Структура искусственного нейрона [14]

Как следует из названия, в полносвязной нейронной сети каждый нейрон одного слоя связан со всеми нейронами следующего слоя, сигнал распространяется через все слои в направлении от входного слоя к выходному. Принято выделять входной и выходной слои, а также скрытые слои (находящиеся между входным и выходным).

На рисунке 1.8 приведено схематическое изображение нейронной сети, состоящей из входного слоя, одного скрытого и выходного слоёв.

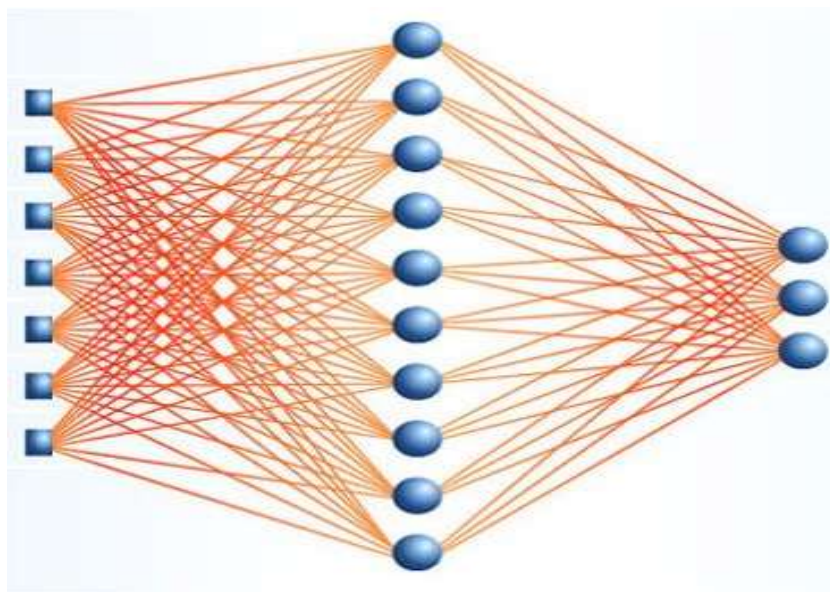


Рисунок 1.8 – Схематическое изображение полносвязной нейронной сети [15]

В случае применения полносвязных нейронных сетей для задач классификации изображений входное изображение подаётся попиксельно на нейроны входного слоя (также возможна подача непосредственно признаков). Таким образом входное изображение представляется в виде линейной цепочки байт, в результате теряется информация о взаимосвязи отдельных частей изображения и о цельных образах. Такие сети могут быть эффективно использованы для классификации простых объектов на небольшое количество классов. В случае более сложной задачи следует разрабатывать классификаторы на основе свёрточных нейронных сетей.

1.3.4. Свёрточные нейронные сети

Решение проблемы потери топологии изображений было найдено американским учёным Яном ЛеКуном, вдохновленным работами нобелевских лауреатов в области медицины. Эти ученые исследовали зрительную кору головного мозга кошки и обнаружили, что существуют так называемые простые клетки, которые особо сильно реагируют на прямые линии под разными углами и сложные клетки, которые реагируют на движение линий в одном направлении. Ян ЛеКун предложил использовать свёрточные нейронные сети и разработал архитектуру свёрточной нейросети LeNet [16].

Идея свёрточных нейронных сетей заключается в чередовании свёрточных слоев, субдискретизирующих слоев и наличии полносвязных слоев на выходе [15], как изображено на рисунке 1.9.

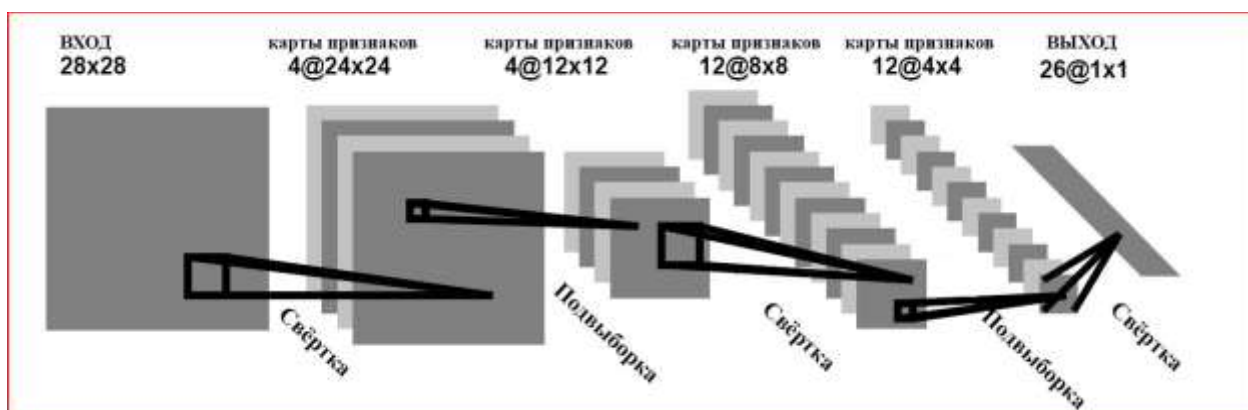


Рисунок 1.9 – Архитектура свёрточной нейронной сети [15]

Выделение объектов на входном изображении обеспечивается операцией свёртки, суть которой заключается в последовательном применении к изображению фильтров. Эти фильтры представляют собой матрицы коэффициентов (ядра свёртки), которые, будучи применёнными ко всем фрагментам изображения, извлекают простые формы, наиболее похожие на сами фильтры. Таким образом, в результате свёртки изображения с неким ядром получается другое изображение, каждый пиксель которого отображает степень схожести фрагментов изображения с фильтром – формируется карта признаков. Слои субдискретизации (пулинга, подвыборки) позволяют в разы уменьшить размерность входящего изображения. За счёт данных операций также обеспечивается инвариантность сети к небольшим искажениям и смещениям входных изображений.

Большое влияние на развитие компьютерного зрения, алгоритмов машинного обучения и применение свёрточных нейронных сетей оказала сеть AlexNet [17], одержавшая в 2012 году победу в конкурсе по распознаванию изображений ImageNet LSVRC-2012, достигнув показателя ошибочных предсказаний в 15,3% (результат второго места составил 26,2%). Сеть AlexNet имеет схожую архитектуру с упомянутой выше сетью LeNet. Основные отличия состоят в том, что AlexNet больше на несколько порядков, имеет несколько дополнительных свёрточных слоёв, размеры ядер свёртки уменьшаются от начальных слоёв нейронной сети к выходным, а также в использовании активационной функции ReLu, операции Dropout (случайное исключение нейронов сети из процесса обучения) и аугментации данных для решения проблемы переобучения сети.

В результате успеха сети AlexNet за свёрточными нейронными сетями закрепилось безусловное лидерство в задачах классификации изображений. Согласно информации из источника [18], начиная с 2012 года в конкурсе ImageNet победу одерживают нейросетевые модели.

1.3.5. Сравнение рассмотренных методов классификации

Сравнение рассмотренных методов построения классификатора приведено в таблице 1.1, в качестве критериев для сравнения выбраны рассмотренные ранее сильные и слабые стороны алгоритмов классификации, влияющие на скорость получения предсказаний и качество результатов.

Таблица 1.1 – Сравнение рассмотренных методов классификации

| | Независимость времени идентификации и от объёма данных | Способность эффективно строить нелинейные зависимости | Устойчивость к большой размерности признаков | Учёт топологии пространства |
|-----------------------------|--|---|--|-----------------------------|
| Методы сравнения с эталоном | - | - | - | - |
| Статистические методы | + | - | - | - |
| Полносвязные нейронные сети | + | + | + | - |
| Свёрточные нейронные сети | + | + | + | + |

1.4. Выводы

В данной главе был проведён обзор публикаций и разработок по теме работы, рассмотрены признаки, которые могут быть выделены на фотографии листа, а также основные методы построения классификаторов. На основании рассмотренных признаков можно заключить, что выбираемый метод классификации должен удовлетворять следующим критериям:

- способность решать задачу классификации на большое количество классов (>100);
- способность выделять признаки не только на контуре листа, но и на листовой пластине;
- высокая скорость идентификации растения, не зависящая от объема эталонных данных;
- учёт топологии пространства изображения;
- высокая точность работы при наличии значительных различий признаков внутри одного класса.

На данный момент оптимальным вариантом для разработки классификатора изображений, способного с высокой точностью выполнять задачу мультиклассификации, является свёрточная нейронная сеть, поскольку по своим возможностям она превосходит все остальные рассмотренные методы и является единственным методом, который соответствует всем выдвинутым требованиям к разрабатываемому классификатору.

ГЛАВА 2. ИССЛЕДОВАНИЕ ВЫБРАННОГО МЕТОДА РЕАЛИЗАЦИИ КЛАССИФИКАТОРА И ПРЕДВАРИТЕЛЬНОЙ ОБРАБОТКИ ВХОДНЫХ ДАННЫХ

2.1. Выбор архитектуры нейронной сети

В предыдущей главе была рассмотрена архитектура нейронной сети AlexNet, которая на данный момент является устаревшей, поскольку за последующие годы было разработано большое количество улучшений за счёт вариаций параметров слоёв и их различных комбинаций. Так как не существует единых принципов построения нейронных сетей, позволяющих разработать наиболее подходящую архитектуру, следует рассмотреть наиболее удачные готовые реализации и сравнить их результаты. Критериями для сравнения выступают результаты точности на тестовом наборе данных, занимаемое дисковое пространство, количество обучаемых параметров и глубина нейронной сети (количество скрытых слоёв). Ниже приведена таблица 2.1, в которой отражены вышеперечисленные характеристики наиболее успешных реализаций свёрточных нейронных сетей, построенная на основании данных источника [19]. Показатели точности распознавания, приведённые в таблице, были получены на наборе данных ImageNet, содержащем 1000 классов объектов.

Таблица 2.1 – Характеристики нейронных сетей

| Название модели | Размер модели, Мбайт | Точность наиболее вероятного предсказания | Точность 5 наиболее вероятных предсказаний | Количество обучаемых параметров | Глубина |
|------------------------|----------------------|---|--|---------------------------------|---------|
| Xception [20] | 88 | 79% | 95% | 22 910 480 | 126 |
| VGG16 [21] | 528 | 71% | 90% | 138 357 544 | 23 |
| ResNet152V2 [22] | 232 | 78% | 94% | 60 380 648 | - |
| InceptionResNetV2 [23] | 215 | 80% | 95% | 55 873 736 | 572 |
| MobileNetV2 [24] | 14 | 71% | 90% | 3 538 984 | 88 |
| NASNetLarge [25] | 343 | 83% | 96% | 88 949 818 | - |

Поскольку задача идентификации растений в большинстве случаев подразумевает потребность в использовании классификатора сразу после обнаружения интересующего растения, классификатор должен соответствовать следующим требованиям:

- высокая скорость работы;
- возможность запуска на маломощных мобильных устройствах (компактный портативный компьютер, планшет, смартфон и т.п.);
- небольшой размер готового классификатора.

Для обеспечения соответствия данным требованиям необходимо, чтобы выбранная модель нейронной сети обладала наименьшим числом используемых параметров, поскольку за счёт этого достигается наиболее высокое быстродействие и меньший размер готовой обученной сети.

По данному критерию безусловным лидером является модель MobileNetV2, она занимает всего 14 Мбайт дискового пространства, и при этом способна обеспечить 90%-ю точность нахождения верного класса среди 5 наиболее

вероятных предсказаний в задаче классификации среди 1000 возможных классов.

2.2. Подробное рассмотрение выбранной нейронной сети

Свёрточная нейронная сеть MobileNet была разработана группой специалистов компании Google и представлена в апреле 2017 года. Данная модель изначально позиционировалась как нейронная сеть, оптимизированная для использования на мобильных устройствах. Её особенностью является отсутствие слоёв подвыборки, функцию которых (снижение пространственной размерности) выполняет свёртка с параметром сдвига, равным двум. Сдвиг обеспечивает применения операции свёртки не к каждому пикселю изображения, а только к тем, чей номер кратен определённому числу.

Свёрточная часть сети MobileNet включает в себя следующие составные части: блок свёртки 3x3 и 13 блоков, изображённых в правой половине рисунка 2.1, в которых постепенно увеличивается число фильтров и снижается пространственная размерность тензора.



Рисунок 2.1 – Обычный свёрточный блок (слева) и базовый блок сети MobileNet (справа) [26], где: BN – Batch Normalization (нормализация входных данных путём центрирования и масштабирования), Conv – свёртка, Depthwise Conv – поканальная свёртка

Важной отличительной чертой архитектуры MobileNetV2 по сравнению с предыдущей версией является то, что в конце модели идёт активируемый линейной функцией блок 1x1-свёртки, понижающий число каналов. По

предположению авторов, таким образом возможно уменьшить размерность пространства без потери полезной информации.

Ниже на рисунке 2.2 изображены архитектурные блоки рассматриваемой нейронной сети.

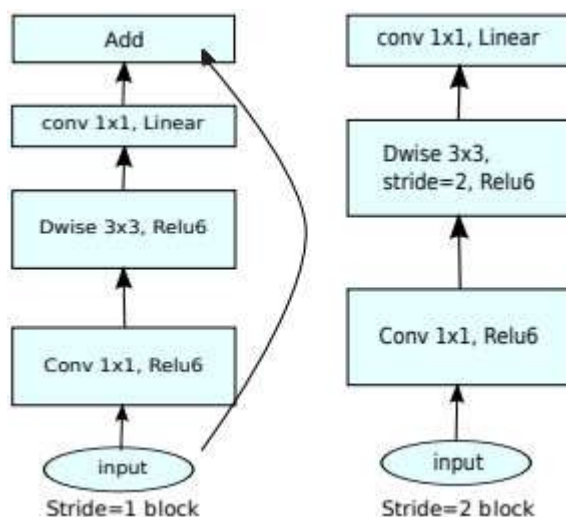


Рисунок 2.2 – Архитектурные блоки нейронной сети MobileNetV2 [26],
где Dwise – depthwise, слой поканальной свёртки

2.3. Выбор способа обучения нейронной сети

Процесс обучения свёрточной нейронной сети является ресурсоёмким, требует больших вычислительных мощностей и, как правило, занимает продолжительное время (от нескольких часов до нескольких дней или, в некоторых случаях, недель). Большую часть от всего времени обучения свёрточной нейронной сети занимает обучение свёрточных слоёв, ответственных за выделение признаков.

Существует способ обучения нейронных сетей под названием Transfer Learning, что можно перевести как «передача обучения», суть которого заключается в использовании готовой модели, уже обученной на каком-то наборе изображений с дальнейшим её применением для решения других типов задач. Данный способ позволяет значительно сократить время обучения нейронной сети, но при этом достичь показателей точности, сопоставимых с

теми, которые были бы получены в результате длительного полного обучения «с нуля». Процесс передачи обучения происходит следующим образом:

- 1) загружается готовая обученная модель;
- 2) из модели удаляются выходные слои;
- 3) оставшиеся слои нейронной сети «замораживаются», т.е. они перестают изменять свои значения в процессе обучения;
- 4) в модель добавляются выходные слои с нужными параметрами;
- 5) запускается обучение выходных слоёв нейронной сети.

Таким образом, передача обучения является эффективным способом обучения модели, поскольку позволяет воспользоваться настроенными входными фильтрами другой модели. И в случае, если она обучалась на большом количестве изображений непохожих объектов, данные фильтры способны выделять множество различных признаков.

В данной работе для обучения классификатора выбран способ передачи обучения, поскольку доступные вычислительные мощности не позволяют провести полноценное обучение свёрточной нейронной сети «с нуля».

2.4. Выбор методов подготовки исходных данных

В данной работе используется набор данных, состоящий из 7039 фотографий листьев на однотонном фоне, относящихся к 196 видам растений, количество изображений для каждого вида варьируется от 10 до 94.

Все фотографии растений из набора тренировочных данных необходимо предварительно специальным образом подготовить для последующего обучения нейронной сети, чтобы избежать ситуации, когда обучение будет проводиться на данных, отличающихся от тех, которые будут переданы ей для идентификации. Для того, чтобы определить, какие преобразования следует применить к исходным изображениям, необходимо рассмотреть два аспекта: специфику модели и вид данных, которые пользователь будет подавать в классификатор для идентификации.

Изначально сеть MobileNetV2 предусматривает следующий формат входных данных: цветное трёхканальное изображение в цветовом пространстве RGB разрешением 224 на 224 пикселя, к такому формату следует привести все обучающие изображения. Разрабатываемая система предполагает, что от пользователя требуется только передать в классификатор фотографию интересующего его растения, сделанную на белом фоне, никаких манипуляций с исходным изображением со стороны пользователя не предусматривается.

Так как выбранная для реализации модель обладает инвариантностью к небольшим изменениям входных данных, важно предоставить для обучения модели фотографии, снятые с различными значительными вариациями положений объектов в кадре, следовательно, набор тренировочных изображений должен соответствовать следующим параметрам: представленность в наборе данных для каждого вида растений фотографий, снятых под различными углами поворота объекта в кадре вокруг его центральной точки, и фотографий, сделанных с различных расстояний до объекта.

В данной работе к фоновой составляющей не применяются преобразования, т.к. она не оказывает негативного эффекта на результат обучения сети, но при этом позволяет натренировать модель на работу с реальными фотографиями. Область листа на изображении также не следует преобразовывать, поскольку какие-либо изменения могут снизить качество идентификации растений, если между обучающими изображениями и фотографиями, переданными сети для распознавания, будут иметься значительные визуальные отличия.

Ситуация, когда набор обучающих данных не включает в себя достаточно различающихся фотографий представителей каждого класса, чтобы в нём в полной мере было представлено большое количество различных вариаций положений объектов в кадре, негативно сказывается на результатах обучения модели, поскольку не позволяет нейронной сети выявить наиболее

характерные для каждого класса признаки. Эта проблема может быть решена двумя способами:

- пополнение набора данных новыми фотографиями объектов;
- расширение объёма обучающих данных за счёт аугментации уже имеющихся в нём фотографий.

Аугментация данных представляет собой процесс генерации новых изображений путём применения некоторых искажений к имеющимся фотографиям. В качестве таких искажений могут быть использованы: поворот изображения, изменение масштаба, растяжение по одной из осей, сдвиг изображения, сдвиг значений каналов цветного изображения, изменение яркости, применение горизонтального и вертикального отражений.

Более предпочтительным способом является пополнение базы новыми фотографиями объектов, поскольку это значительно увеличивает число уникальных признаков, которые могут быть выделены и использованы для обучения классификатора. Но в случае, если таковой возможности нет, следует воспользоваться вторым способом, он также является довольно эффективным и часто применяется при обучении классификаторов на малом наборе обучающих данных. В данной работе нет возможности значительно увеличить объём обучающего набора данных добавлением новых фотографий, поэтому выбран способ аугментации данных.

Для фотографий листьев подходят не все из перечисленных ранее вариантов аугментации данных, например, растягивание изображения исказит форму листовой пластины и окажет негативный эффект на качество обучения. Наиболее подходящими являются следующие преобразования: повороты изображений, отражения по обеим осям и масштабирование, поскольку они не искажают форму листа.

2.5. Выбор программных средств для реализации классификатора

На данный момент наиболее популярным языком программирования для разработки нейронных сетей является Python 3, для него разработано большое количество библиотек, позволяющих быстро и эффективно разрабатывать нейронные сети различных конфигураций. В качестве инструментов разработки классификатора выбраны следующие библиотеки и программы: нейросетевая библиотека Keras, библиотека для работы с изображениями OpenCV, библиотека TelegramBotAPI для реализации Telegram-бота, среда разработки Android Studio (язык программирования Java) и библиотека TensorFlow для портирования готовой модели в приложение для операционной системы Android, поскольку данные средства являются наиболее распространёнными, эффективными и удобными для использования как для разработки, так и для применения.

При разработке нейронной сети важно иметь доступ к вычислительным машинам высокой мощности, оптимальным устройством для обучения нейронных сетей является графический ускоритель. В результате анализа доступных вычислительных мощностей в качестве платформы для разработки и обучения нейронной сети был выбран сервис облачных вычислений Google Colab, позволяющий удалённо использовать графические ускорители высокой вычислительной мощности. В связи с этим было принято решение воспользоваться облачным хранилищем Google Drive для хранения наборов данных и обученных моделей, поскольку это наиболее быстрый и удобный способ обмена файлами с удалённой вычислительной машиной.

2.6. Выводы

В данной главе было проведено сравнение наиболее удачных свёрточных нейронных сетей, в результате которого выбрана сеть MobileNetV2, т.к. она лишь незначительно уступает в качестве распознавания объектов, при этом имея размер и количество параметров в разы меньше, чем у других моделей.

Также были исследованы архитектура и особенности выбранной сети, рассмотрены возможные варианты обучения модели и выбран способ передачи обучения для тренировки классификатора.

Для увеличения числа обучающих данных выбраны следующие преобразования: повороты изображений, горизонтальное и вертикальное отражения, применение масштабирования, как в положительную, так и в отрицательную стороны.

В качестве средств разработки выбраны следующие программные средства: языки программирования Python 3.8 и Java, библиотеки OpenCV, Keras, TensorFlow, TelegramBotAPI, среды разработки PyCharm и Android Studio, сервисы Google Colab и Google Drive.

ГЛАВА 3. РАЗРАБОТКА АЛГОРИТМА КЛАССИФИКАЦИИ РАСТЕНИЙ ПО ИЗОБРАЖЕНИЮ ЛИСТА И ЕГО ПРОГРАММНОЙ РЕАЛИЗАЦИИ

3.1. Обработка данных для обучения нейронной сети

Фотографии листьев, представленные в используемом наборе данных, не имеют фиксированного размера, что было исправлено путём экстраполяции фона (во избежание деформации листа) до квадратных размеров изображения с последующим изменением разрешения до 224x224 пикс. Пример результата данных преобразований приведён ниже на рисунке 3.1 с сохранением пропорций. Исходные изображения были представлены в формате png, результаты преобразований были переведены в формат jpg для уменьшения занимаемого дискового пространства.

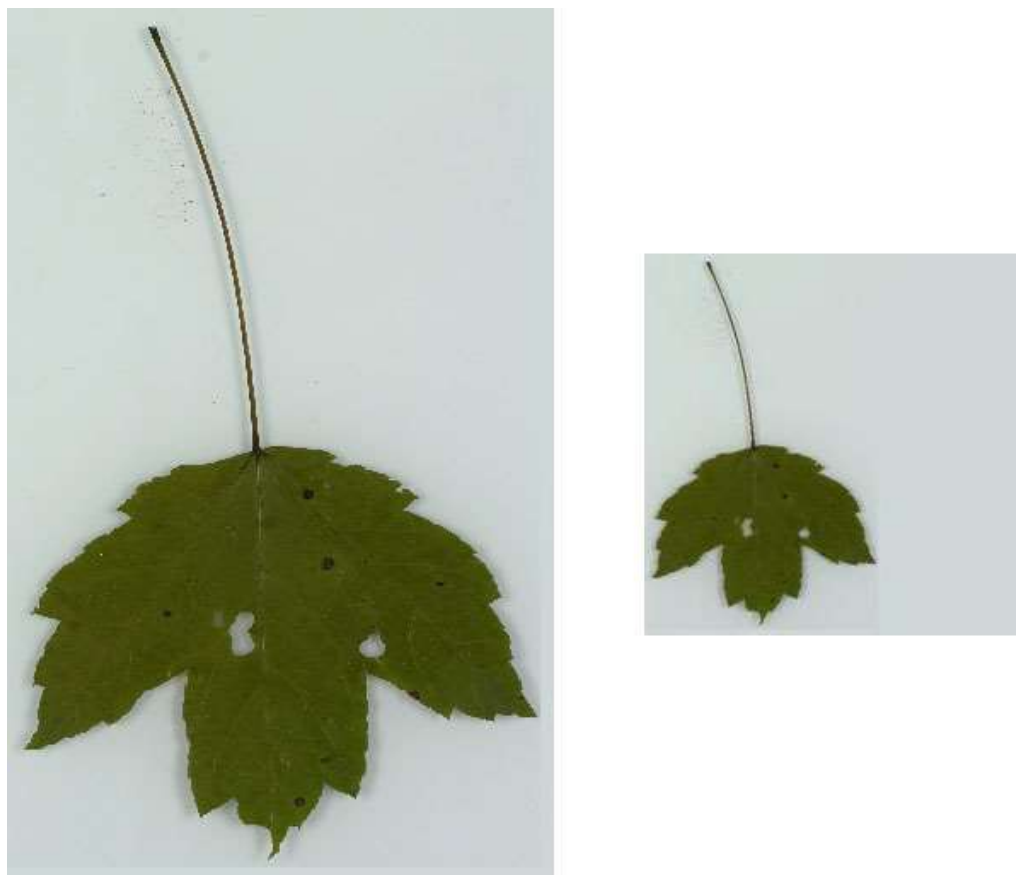


Рисунок 3.1 – Исходное изображение (слева), результат преобразований, подаваемый в нейронную сеть (справа)

Исходный набор изображений был разделён на три подкаталога: train – обучающий, val – валидационный, и test – тестовый наборы в соотношении: 80/10/10 соответственно. Объём обработанных данных составил 37.3Мб (против 1.6 Гб объёма изначального набора данных).

Следующим шагом предварительной обработки входных данных стало применение к тренировочной выборке аугментации данных, реализованной в библиотеке keras методом ImageDataGenerator(), аугментация данных проводилась по следующим параметрам:

- *rescale* = 1. / 255 – перевод значений пикселей изображения из диапазона [0, 255] в диапазон [0, 1];
- *rotation_range* = 30 – максимальный угол поворота изображения, поворот осуществляется в обе стороны: по часовой и против часовой стрелки;
- *zoom_range* = 0.5 – максимальный множитель увеличения/уменьшения масштаба изображения;
- *horizontal_flip* = True – применение к изображению горизонтального отражения;
- *vertical_flip* = True – применение к изображению вертикального отражения.

Перевод значений пикселей к диапазону [0,1] обусловлен тем, что такой формат представления данных более пригоден для нейронных сетей. Значения угла поворота и множителя масштаба выбраны на основе визуального анализа изображений, представленных в наборе обучающих данных, т.к. были сочтены наиболее подходящими для увеличения разнообразия представленных положений листьев на фотографии. Ниже на рисунке 3.2 изображены примеры результата аугментации данных.

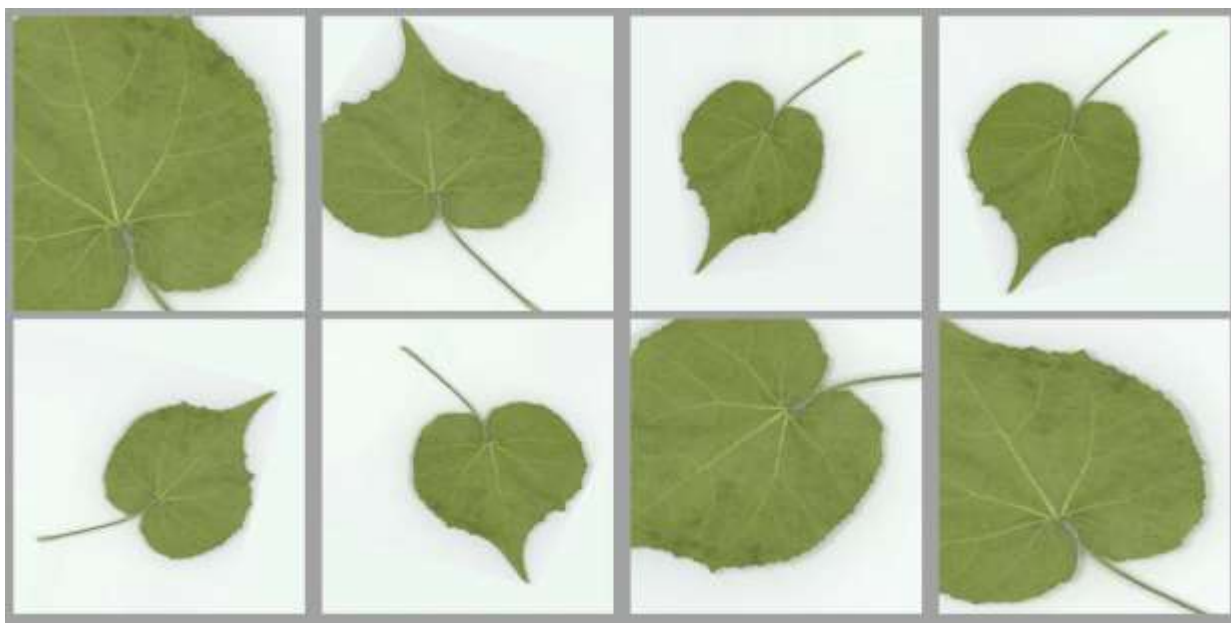


Рисунок 3.2 – Примеры аугментированных данных

Как видно из рисунка 3.2, аугментация данных действительно позволяет достичь большого числа различных вариаций положения листа в кадре за счёт использования лишь одного изображения, что в конечном итоге позволяет нейронной сети извлечь большое количество признаков, находящихся на изображении одного листа.

3.2. Программная реализация алгоритма классификации растений по изображению листа

Разработка модели была реализована в интерактивной вычислительной среде Google Colab, построенной на основе Jupyter Notebook. В ходе программной реализации классификатора на базе нейронной сети MobileNetV2 использовались язык программирования Python 3 и нейросетевая библиотека Keras. Последовательность создания и обучения нейронной сети можно представить в виде следующих этапов:

- 1) подключение к проекту всех необходимых модулей, предназначенных для построения модели нейронной сети, обмена файлами с Google Drive, аугментации данных и оценки результатов обучения нейронной сети;
- 2) загрузка данных из Google Drive в память вычислительной машины;
- 3) загрузка из библиотеки Keras модели сети MobileNetV2 с весами сети, обученной на наборе данных «Imagenet», без выходных слоёв;
- 4) блокировка обучаемости слоёв загруженной сети;
- 5) добавление к сети выходного слоя, включающего 196 нейронов (в соответствии с количеством классов растений);
- 6) создание трёх объектов типа ImageDataGenerator, представляющих собой тренировочный, валидационный и тестовый наборы данных;
- 7) применение к тренировочному набору изображений аугментации данных с указанием параметров, рассмотренных в предыдущем разделе данной главы;
- 8) запуск обучения нейронной сети на 100 эпох, batch_size=64;
- 9) оценка точности работы на тестовом наборе данных;
- 10) сохранение обученной модели на Google Drive.

Код программы с подробными пояснениями выполняемых действий приведён в Приложении А к ВКР.

3.3. Анализ результатов работы алгоритма

Визуализация процесса обучения приведена на рисунке 3.3.

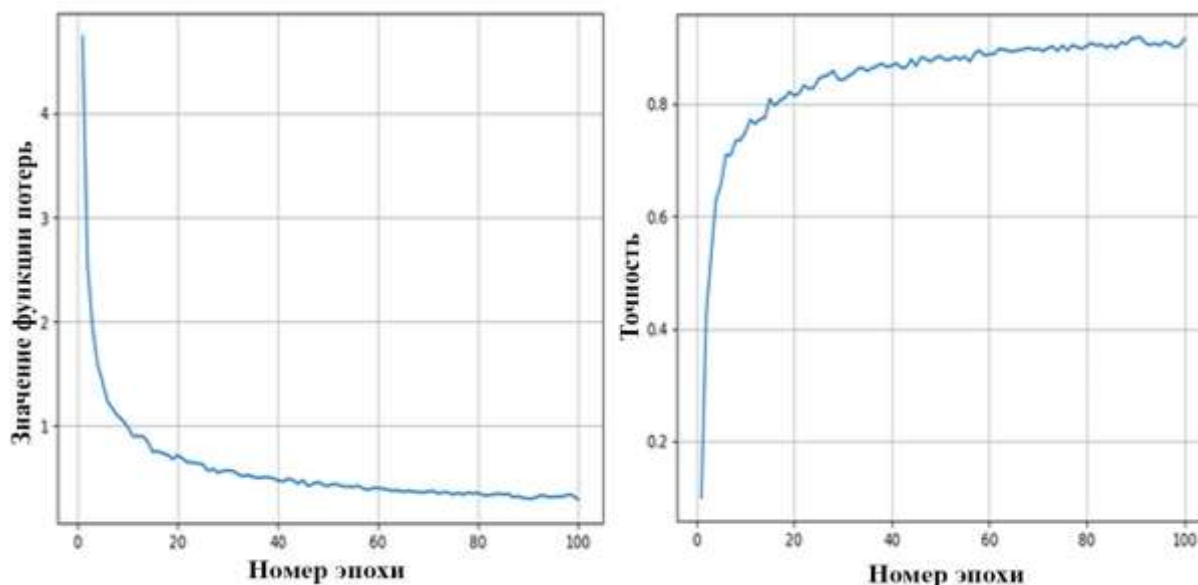


Рисунок 3.3 – Визуализация процесса обучения нейронной сети, слева – функция потерь, справа – функция точности

По результатам 100 эпох обучения точность 1 наиболее вероятного предсказания нейронной сети на тестовой выборке составила 85%. Точность 5 наиболее вероятных – 97%.

Время обучения сети составило 2 часа 4 минуты.

Для получения предсказания обученной модели, запускаемой на ПК, пользователю необходимо указать путь к фотографии растения, предсказание для которой требуется получить, и исполнить соответствующий блок кода.

Результатом выполнения является список из пяти наиболее вероятных классов растений, отсортированных в порядке убывания вероятности. Для демонстрации работы классификатора было проведено испытание на реальной фотографии, сделанной на найденных листьях растения Болиголов пятнистый (лат. наименование *Conium Maculatum*), пример переданной для распознавания изображён на рисунке 3.4. Формат ответа нейронной сети приведён на рисунке 3.5.



Рисунок 3.4 – Одна из фотографий растения *Conium Maculatum*, переданная разработанной нейронной сети для распознавания

| | |
|--|---|
| <p>Conium_maculatum1.jpg :</p> <p>Conium_maculatum</p> <p>Geum_canadense</p> <p>Hydrophyllum_virginianum</p> <p>Potentilla_canadensis</p> <p>Tanacetum_vulgare</p> | <p>Conium_maculatum3.jpg :</p> <p>Conium_maculatum</p> <p>Geum_canadense</p> <p>Hydrophyllum_virginianum</p> <p>Potentilla_canadensis</p> <p>Senecio_aureus</p> |
| <p>Conium_maculatum2.jpg :</p> <p>Acer_saccharinum</p> <p>Conium_maculatum</p> <p>Geum_canadense</p> <p>Hydrophyllum_virginianum</p> <p>Tanacetum_vulgare</p> | <p>Conium_maculatum5.jpg :</p> <p>Conium_maculatum</p> <p>Geum_canadense</p> <p>Hydrophyllum_virginianum</p> <p>Senecio_aureus</p> <p>Tanacetum_vulgare</p> |

Рисунок 3.5 – Пример предсказаний нейронной сети в среде Google Colab

3.4. Сравнение результатов разработанного классификатора с классификатором на основе метода IDSC

В главе, посвященной анализу опубликованных работ, была рассмотрена система идентификации растений, построенная на основе методов IDSC и KNN, так как количество использованных в ней эталонных фотографий и классов растений сопоставимо с разрабатываемой системой, можно провести сравнение точности распознавания данных классификаторов. В упомянутой выше работе приводится график зависимости точности классификатора от количества выводимых предсказаний. Для разработанной в данной работе нейронной сети были построены такие же зависимости, вид графиков приведён на рисунках 3.6 и 3.7. Как видно из графиков, при любом количестве предсказаний, выводимых классификатором, разработанная в данной работе нейронная сеть показывает более точный результат.

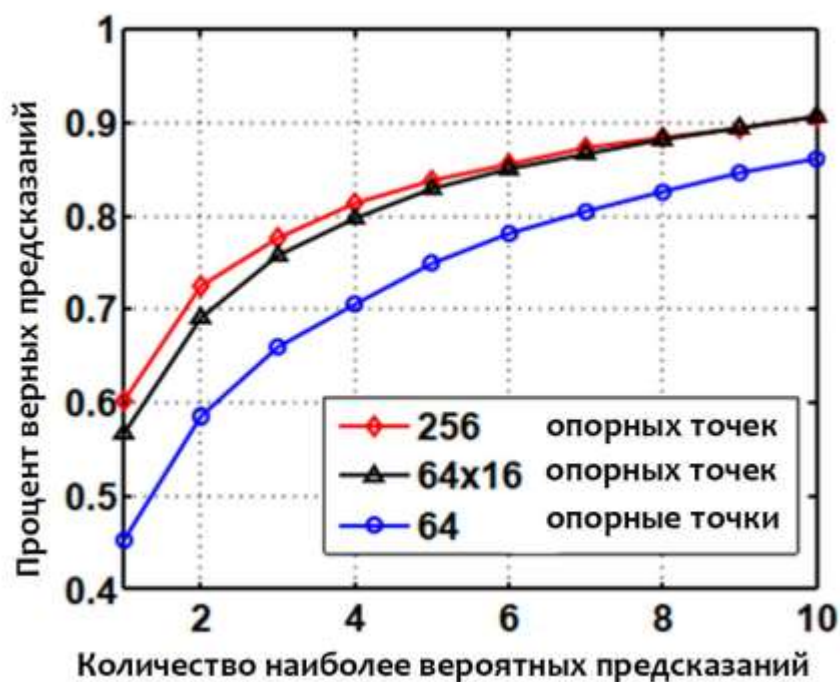


Рисунок 3.6 - Зависимость точности классификатора IDSC от количества выводимых предсказаний [15]

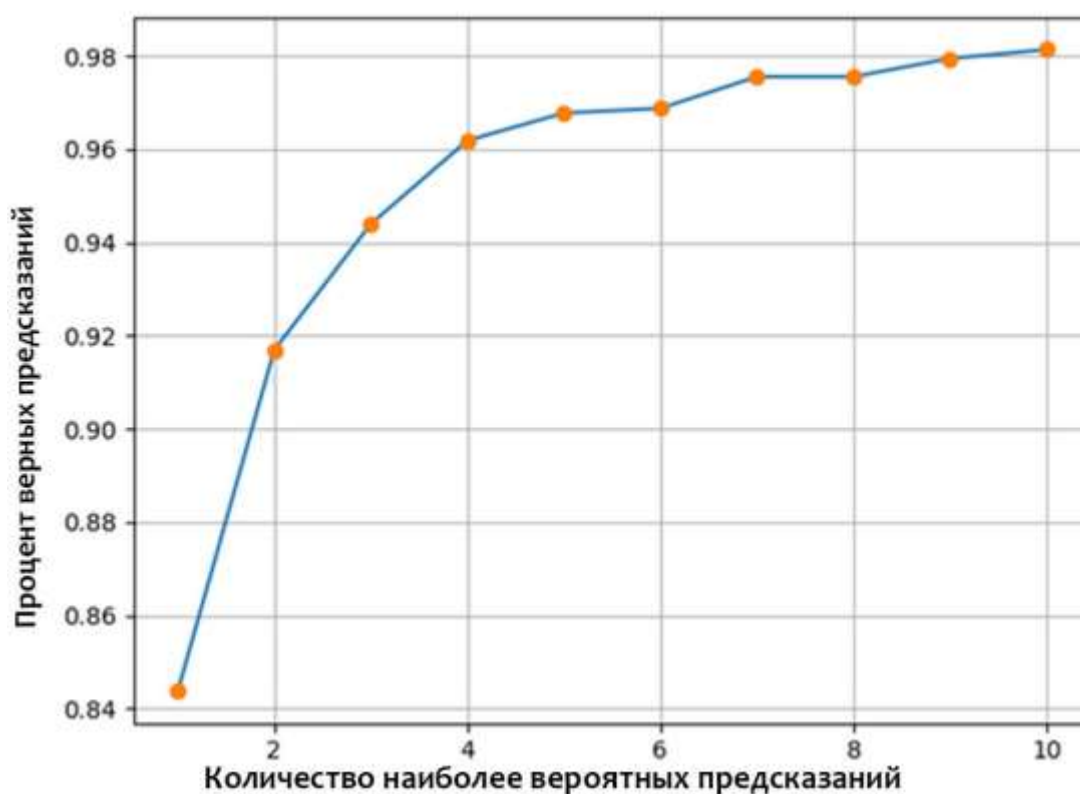


Рисунок 3.7 - Зависимость точности разработанного классификатора от количества выводимых предсказаний

3.5. Разработка программного обеспечения для осуществления классификации с мобильных устройств

Для разработанного классификатора важным требованием является способность обеспечивать пользователю быструю идентификацию растения. Следовательно, необходимо реализовать возможность использования классификатора на различных портативных устройствах. Разработанная нейронная сеть в виде keras-модели наиболее пригодна для использования на компьютерах под управлением операционных систем Windows, Linux, MacOS, большинство устройств под управлением данных систем – это персональные компьютеры, моноблоки и ноутбуки, не являющиеся портативными устройствами, удобными для переноски.

Подавляющее большинство мобильных устройств работают на базе операционных систем iOS и Android. Для использования реализованного

классификатора на этих ОС необходимо разработать подходящее программное обеспечение. Можно выделить два подхода к решению данной задачи:

- классификация на мобильном устройстве;
- классификация на удалённом устройстве путём обмена данными через сеть Интернет.

Осуществить классификацию на устройстве можно разработав мобильное приложение с загруженной в него обученной нейронной сетью. Один из наиболее удобных и эффективных способов разработки мобильного приложения – разработка в среде Android Studio на языке Java.

Классификация на удалённом устройстве может быть осуществлена путём передачи изображения на устройство с работающей моделью и получением предсказаний в ответном сообщении. Данную задачу можно осуществить путём обмена данными через мессенджер Telegram.

3.5.1. Мобильное приложение

Разработанная нейронная сеть с помощью методов библиотеки TensorFlow была сконvertирована в файл формата tflite, позволяющий использовать данную модель на устройствах на базе операционной системы Android. В среде разработки Android Studio было разработано мобильное приложение для классификации растений.

Выполняемая приложением последовательность действий для идентификации растения выглядит следующим образом:

- 1) в приложение загружается фотография интересующего растения;
- 2) разрешение изображения приводится к размерам 224x224 пикс.;
- 3) обработанное изображение подаётся на входы нейронной сети;
- 4) ответы нейронной сети сортируются в порядке убывания вероятности;
- 5) из полученной последовательности извлекаются 5 наиболее вероятных предсказаний, представляющих собой индексы наименований;
- 6) полученным индексам сопоставляются названия видов растений и выводятся на экран приложения.

Данное приложение содержит один рабочий экран, на котором отображаются следующие элементы:

- кнопка «Выбрать фото», открывающая окно выбора фотографий;
- окно вывода выбранного изображения (по умолчанию отображается логотип операционной системы Android);
- надпись «Результат распознавания:», расположенная под окном вывода изображения;
- пять кнопок с наиболее вероятными предсказаниями нейронной сети, по нажатию на которые пользователю демонстрируется пример листа растения;
- кнопка с названием выбранного пользователем изображения.

На рисунке 3.8 приведены снимки экрана приложения, демонстрирующие интерфейс и результаты идентификации растений по фотографии, полученной с камеры смартфона, вид растения *Alliaria Petiolata* идентифицирован верно.



Рисунок 3.8 – Интерфейс разработанного мобильного приложения, слева – начальный экран, справа – экран результатов идентификации

Недостатком данной реализации является то, что разработанное мобильное приложение доступно только для мобильных устройств под управлением операционной системы Android.

3.5.2. Telegram-бот

С использованием библиотеки TelegramBotAPI был разработан Telegram-бот, позволяющий осуществить классификацию растения по фотографии его листа. Для получения предсказаний классов от разработанной нейронной сети пользователю необходимо начать диалог с ботом, после чего отправить фотографию листа интересующего растения. В ответ бот отправляет пользователю сообщение, состоящее из пяти строк с наименованиями предсказанных классов растения в порядке уменьшения вероятности.

Для работы данного способа классификации необходимо обеспечить соблюдение следующих условий:

- наличие серверной части системы, на которой нейронная сеть выполняет вычисления для осуществления предсказания;
- наличие подключения к сети Интернет на мобильном устройстве и серверной части для обмена данными;
- наличие на мобильном устройстве приложения Telegram для обмена данными.

Последовательность выполняемых действий для идентификации растения по фотографии его листа состоит из следующих этапов:

- 1) отправка пользователем фотографии в диалог с ботом;
- 2) получение серверной частью отправленного изображения и приведение его к параметрам, соответствующим требованиям нейронной сети (рассмотрено в пункте 3.1);
- 3) передача предобработанного изображения в нейронную сеть;
- 4) получение предсказаний нейронной сети;
- 5) формирование списка из 5 наиболее вероятных предсказаний, генерация для каждого названия ссылки с запросом в поисковой системе и отправка их пользователю.

Время от отправки фотографии листа до получения предсказаний составляет не более двух секунд.

На рисунке 3.9 приведены снимки экрана диалога с Telegram-ботом, демонстрирующие этапы получения пользователем результатов классификации, вид растения *Acer Negundo* идентифицирован верно.

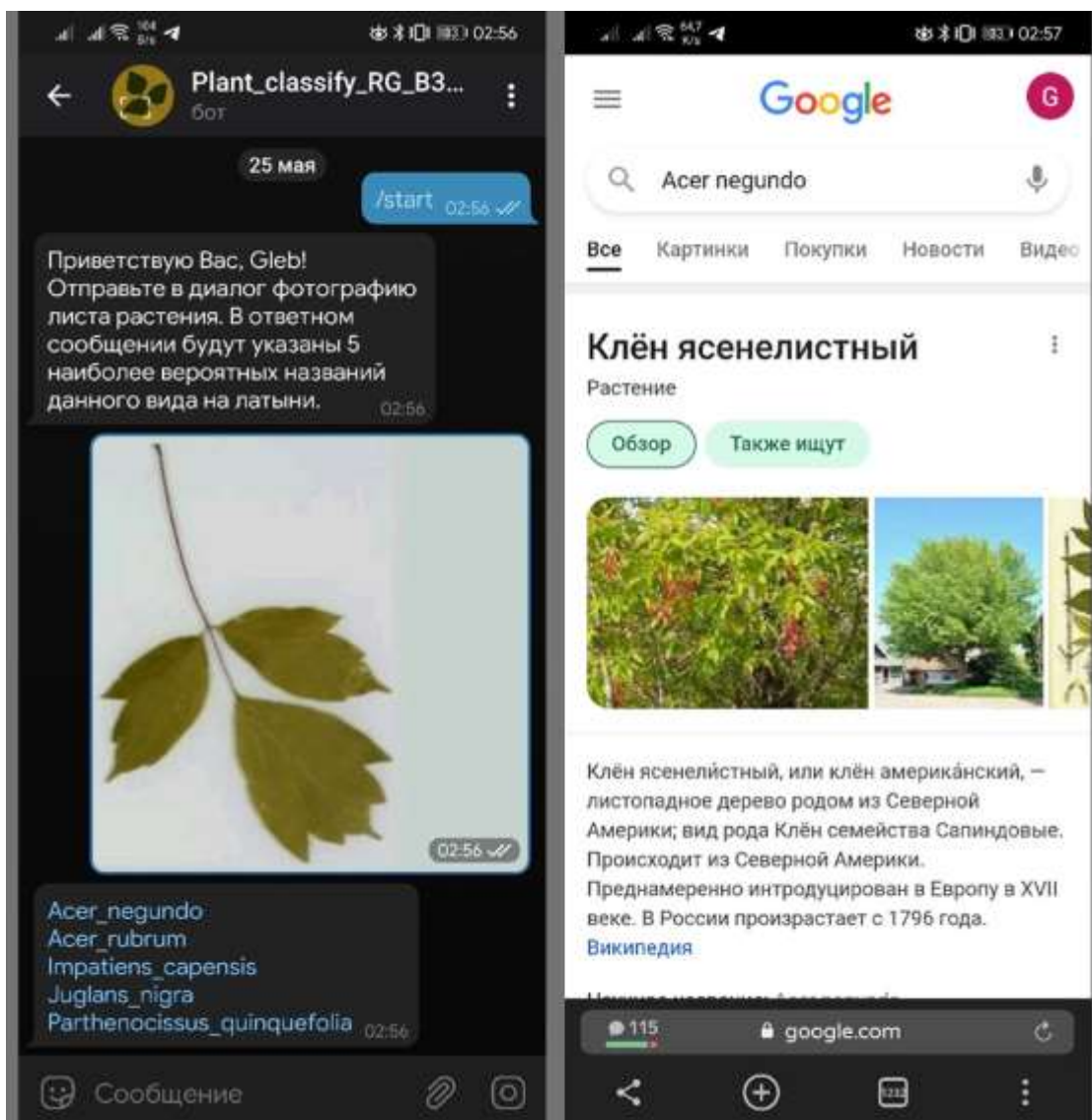


Рисунок 3.9 – Этапы идентификации растения через Telegram

Преимуществом данного метода является возможность его использования на любой платформе, для которой существует клиент мессенджера Telegram (iOS, Android, Windows, Linux, MacOS).

3.6. Выводы

В данной главе была программно реализована предварительная обработка набора данных для обучения классификатора, построена и натренирована путём передачи обучения свёрточная нейронная сеть MobileNetV2, достигнута точность в 97% среди 5 наиболее вероятных предсказаний. Результат сравнения с классификатором, основанном на методе IDSC, показал значительное превосходство разработанной модели по точности распознавания, скорости работы и занимаемому классификатором дисковому пространству. Для соблюдения разработанной моделью требования работы на мобильных устройствах были разработаны два метода получения предсказаний нейронной сети с устройства:

- мобильное приложение для устройств на базе ОС Android, позволяющее в течение нескольких секунд идентифицировать растение в автономном режиме работы;
- Telegram-бот, позволяющий в течение двух секунд получить предсказания нейронной сети на любом современном устройстве при наличии подключения к сети Интернет, а также в одно нажатие перейти на страницу с подробным описанием идентифицированного растения.

Разработанное программное обеспечение протестировано на реальных фотографиях, во всех случаях использования растение было идентифицировано верно.

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы был проведен анализ публикаций и разработок по теме работы, исследованы характерные признаки листьев, по которым может быть осуществлена идентификация растений, проведено сравнение методов классификации. В результате для реализации был выбран путь разработки классификатора на основе свёрточной нейронной сети.

По итогам проведённого сравнительного обзора реализаций свёрточных нейронных сетей была выбрана модель MobileNetV2, поскольку она в полной мере соответствует требованиям, выдвинутым к классификатору: идентифицирует объекты с высокой точностью, является очень компактной, затрачивает мало времени на предсказание, а также может быть использована на большинстве портативных устройств.

Разработанная сеть показала высокие результаты точности предсказаний - среди 5 наиболее вероятных классов растение верно идентифицируется в 97% случаев. Сравнение с рассмотренной во время анализа публикаций системой классификации растений, построенной на основе метода IDSC, показало превосходство разработанной нейронной сети более чем на 20% при выводе одного наиболее вероятного предсказания, и более чем на 7% при выводе 10 наиболее вероятных.

Для осуществления возможности идентификации растений сразу после фотографирования листа, на основе реализации нейронной сети было разработано следующее программное обеспечение:

- Приложение для устройств на базе операционной системы Android, идентифицирующее растение по листу течение нескольких секунд;
- Бот для мессенджера Telegram, который менее чем за 2 секунды после получения фотографии листа отправляет пользователю список пяти наиболее вероятных наименований растения.

Разработанный классификатор соответствует всем выдвинутым к нему на стадии проектирования требованиям, превосходит по точности распознавания разработанные ранее классификаторы растений. Обеспечивает широкий выбор способов его использования – разработанные мобильные приложения позволяют запустить его на любой из современных операционных систем. Готовая модель может быть применена как для использования в научно-исследовательских целях, так и в образовательно-развлекательных. Данная система идентификации объектов по изображению может быть обучена на другом наборе данных и применена для использования в любой области, где требуется быстро и точно идентифицировать объект.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. NatureID. NatureID - Identify plants, flowers, trees, cats and dogs! [Электронный ресурс] URL: <http://www.natureid.com> (дата обращения: 10.04.2021).
2. LEAFSNAP. Plant Identification [Электронный ресурс] URL: <https://plantidentifier.info> (дата обращения: 10.04.2021).
3. Plantsnap. Plant Identifier App [Электронный ресурс] URL: <https://www.plantsnap.com> (дата обращения: 10.04.2021).
4. Google Play. Plant Lens - Plant & Flower Identification [Электронный ресурс] URL: <https://play.google.com/store/apps/details?id=app.plant.identification&hl=ru&gl=US> (дата обращения: 10.04.2021).
5. Peter N. Belhumeur D.C.S.F.D.W.J. Searching the World's Herbaria: A System for Visual Identification of Plant Species // European Conference on Computer Vision. 2008. С. 5-10.
6. Ling H., Jacobs D.W. Shape Classification Using the Inner-Distance [Электронный ресурс] [2002]. URL: https://www.cs.umd.edu/~djacobs/pubs_files/ID-pami-8.pdf (дата обращения: 02.5.2021).
7. Harrison O. Towards Data Science. Machine Learning Basics with the K-Nearest Neighbors Algorithm [Электронный ресурс] URL: <https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761> (дата обращения: 02.05.2021).
8. Plantarium. Plants and lichens of Russia and neighboring countries: open online galleries and plant identification guide. [Электронный ресурс] URL: <https://www.plantarium.ru> (дата обращения: 21.04.2021).
9. Academic. Анатомия и морфология высших растений [Электронный ресурс] URL: https://plant_anatomy.academic.ru/741/ (дата обращения: 24.04.2021).

10. Docplayer. Лист как орган фотосинтеза [Электронный ресурс] URL: <https://docplayer.ru/125871245-10-list-kak-organ-fotosinteza.html> (дата обращения: 22.04.2021).
11. Karatu. Жилкование листа [Электронный ресурс] URL: <https://karatu.ru/zhilkovanie-lista> (дата обращения: 22.04.2021).
12. Зенин А.В. Молодой ученый. Анализ методов распознавания образов 2017. № 16. С. 125-130.
13. Robinson S. StackAbuse. K-Nearest Neighbors Algorithm in Python and Scikit-Learn [Электронный ресурс] URL: <https://stackabuse.com/k-nearest-neighbors-algorithm-in-python-and-scikit-learn> (дата обращения: 24.04.2021).
14. Freshgeek. Искусственный нейрон. структура искусственного нейрона [Электронный ресурс] URL: <https://freshgeek.ru/iskusstvennyj-nejron-struktura-iskusstvennogo-nejrona> (дата обращения: 24.04.2021).
15. Habr. Применение нейросетей в распознавании изображений [Электронный ресурс] URL: <https://habr.com/ru/post/74326> (дата обращения: 24.04.2021).
16. Bengio Y.L.A.Y. Convolutional Networks for Images, Speech, and Time-Series / ред. Arbib M.A. MIT Press, 1995.
17. Krizhevsky A., Sutskever I., E. Hinton G. NeurIPS. ImageNet classification with deep convolutional neural networks [Электронный ресурс] URL: <https://papers.nips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf> (дата обращения: 15.04.2021).
18. Habr. Обзор топологий глубоких сверточных нейронных сетей [Электронный ресурс] URL: <https://habr.com/ru/company/mailru/blog/311706> (дата обращения: 25.04.2021).
19. Keras. Keras Applications [Электронный ресурс] URL: <https://keras.io/api/applications> (дата обращения: 25.04.2021).

20. Chollet F. Paperswithcode. Xception: Deep Learning With Depthwise Separable Convolutions [Электронный ресурс] URL: <https://paperswithcode.com/paper/xception-deep-learning-with-depthwise-1> (дата обращения: 02.05.2021).
21. Simonyan K., Zisserman A. Cornell University. Very Deep Convolutional Networks for Large-Scale Image Recognition [Электронный ресурс] URL: <https://arxiv.org/abs/1409.1556> (дата обращения: 02.05.2021).
22. He K., Zhang X., Ren S., Sun J. Cornell University. Deep Residual Learning for Image Recognition [Электронный ресурс] URL: <https://arxiv.org/abs/1512.03385> (дата обращения: 02.05.2021).
23. Szegedy C., Ioffe S., Vanhoucke V., Alemi A. Paperswithcode. Inception-ResNet-v2 [Электронный ресурс] URL: <https://paperswithcode.com/method/inception-resnet-v2> (дата обращения: 02.05.2021).
24. Sandler , Howard A., Zhu M., Zhmoginov A., Chen L.C. Cornell University. MobileNetV2: Inverted Residuals and Linear Bottlenecks [Электронный ресурс] URL: <https://arxiv.org/abs/1801.04381> (дата обращения: 02.05.2021).
25. Tsang S.H. Sik-Ho Tsang. Review: NASNet — Neural Architecture Search Network (Image Classification) [Электронный ресурс] URL: <https://sh-tsang.medium.com/review-nasnet-neural-architecture-search-network-image-classification-23139ea0425d> (дата обращения: 02.05.2021).
26. Habr. MobileNet: меньше, быстрее, точнее [Электронный ресурс] URL: <https://habr.com/ru/post/352804> (дата обращения: 25.04.2021).

ПРИЛОЖЕНИЕ А

В приложении приводится исходный код реализации нейронной сети с пояснением выполняемых на каждом шаге действий.

Подключение библиотек:

```
from keras.preprocessing.image import ImageDataGenerator
from keras.models import Sequential
from keras.layers import Dense, GlobalAveragePooling2D
from keras.applications import MobileNetV2
from keras.metrics import TopKCategoricalAccuracy
import zipfile
import os
from google.colab import drive
```

Подключение Google Drive:

```
drive.mount('/content/drive')
```

Распаковка набора данных, загруженного на Google Drive

```
zip = 'drive/MyDrive/Plant/ds_sep.zip'
z = zipfile.ZipFile(zip, 'r')
z.extractall()
```

Указание пути к тренировочным, проверочным и тестовым данным:

```
train_dir = 'ds_sep/train'
val_dir = 'ds_sep/val'
test_dir = 'ds_sep/test'
```

Параметры изображения, передаваемого в нейронную сеть:

```
img_width, img_height = 224, 224
```

Размер мини-выборки, передаваемой в нейронную сеть:

```
batch_size = 64
```

Количество изображений для обучения:

```
nb_train_samples = 4570
```

Количество изображений для проверки:

```
nb_validation_samples = 1407
```

Количество изображений для тестирования:

```
nb_test_samples = 1062
```

Загрузка готовой модели сети MobileNetV2, обученной на наборе данных «ImageNet», без верхних слоёв, с указанием входного размера изображений:

```
MN2_net = MobileNetV2(weights='imagenet', include_top=False, input_shape=(224, 224, 3))
```

Отключение обучаемости слоёв загруженной нейронной сети:

```
MN2_net.trainable = False
```

Сборка модели нейронной сети путём объединения загруженной модели с выходным обучаемым слоем:

```
model = Sequential()  
model.add(MN2_net)  
model.add(GlobalAveragePooling2D())  
model.add(Dense(196, activation='softmax'))
```

Компиляция модели:

```
model.compile(loss='categorical_crossentropy',  
              optimizer='adam',  
              metrics=['accuracy', TopKCategoricalAccuracy(k=5)])
```

Создание генератора тренировочных данных, применяющего аугментацию:

```
train_datagen = ImageDataGenerator(rescale=1. / 255,  
                                   rotation_range=30, zoom_range=0.5,  
                                   horizontal_flip=True, vertical_flip=True)
```

Создание генератора проверочных и тестовых данных:

```
datagen = ImageDataGenerator(rescale=1. / 255)
```

Создание готового к использованию обучающего набора данных:

```
train_generator = train_datagen.flow_from_directory(  
    train_dir,  
    target_size=(img_width, img_height),  
    batch_size=batch_size,  
    class_mode='categorical')
```

Создание готового к использованию проверочного набора данных:

```
val_generator = datagen.flow_from_directory(  
    val_dir,  
    target_size=(img_width, img_height),  
    batch_size=batch_size,  
    class_mode='categorical')
```

Создание готового к использованию обучающего тестового данных:

```
test_generator = datagen.flow_from_directory(  
    test_dir,  
    target_size=(img_width, img_height),  
    batch_size=batch_size,  
    class_mode='categorical')
```

Запуск обучения нейронной сети:

```
model.fit_generator(  
    train_generator,  
    steps_per_epoch=nb_train_samples // batch_size,  
    epochs=50,  
    validation_data=val_generator,  
    validation_steps=nb_validation_samples // batch_size)
```

Запуск оценки точности нейронной сети на тестовом наборе данных:

```
scores = model.evaluate_generator(test_generator, nb_test_samples // batch_size)
```

Вывод результатов оценки точности:

```
print(scores)
```

Сохранение готовой обученной модели:

```
model.save("drive/MyDrive/result/MobileNetV2_model.h5")
```