

Министерство науки и высшего образования Российской Федерации
Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий

Высшая школа интеллектуальных систем и суперкомпьютерных технологий



**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ
РАБОТА БАКАЛАВРА**

**РАЗРАБОТКА СИСТЕМЫ ВЫЯВЛЕНИЯ ФИТОСАНИТАРНЫХ УГРОЗ С
ИСПОЛЬЗОВАНИЕМ НЕЙРОННЫХ СЕТЕЙ**

Студент гр. 3530901/60201 Д.Ю. Еременко

Санкт-Петербург

2020

Министерство науки и высшего образования Российской Федерации
Санкт-Петербургский политехнический университет Петра Великого
Высшая школа интеллектуальных систем и суперкомпьютерных технологий
Кафедра «Компьютерные системы и программные технологии»

Работа допущена к защите
Директор школы

_____ Ицыксон В. М.

«__» _____ 2020 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

РАЗРАБОТКА СИСТЕМЫ ВЫЯВЛЕНИЯ ФИТОСАНИТАРНЫХ УГРОЗ С ИСПОЛЬЗОВАНИЕМ НЕЙРОННЫХ СЕТЕЙ.

по направлению подготовки 09.03.01 «Информатика и вычислительная техника»

по образовательной программе

09.03.01_02 «Технологии разработки программного обеспечения»

Выполнил студент
гр. 3530901/60201

Д. Ю. Еременко

Руководитель,
ст. преподаватель

А. А. Федотов

Нормоконтролер,
старший преподаватель

С. А. Нестеров

Санкт-Петербург
2020

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий Высшая школа интеллектуальных
систем и суперкомпьютерных технологий

УТВЕРЖДАЮ

Директор высшей школы

_____ В.М. Ицкисон

«___» _____ 2020 г.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы

студенту Еременко Даниле Юрьевичу, группа 3530901/60201 1.

Тема работы: «Разработка системы выявления фитосанитарных угроз с использованием нейронных сетей».

2. Срок сдачи студентом законченной работы: июнь 2020 года.

3. Исходные данные по работе: отечественная и зарубежная литература и периодика.

4. Содержание работы (перечень подлежащих разработке вопросов):

1) сравнить существующие методы классификации изображений и выбрать наиболее подходящий для классификации болезней картофеля по видимым на фотографии признакам

2) разработать утилиту, позволяющую формировать обучающие выборки по фотографиям полей для передачи специалисту по защите растений с целью получения обучающей выборки

3) разработать утилиту, позволяющую производить аугментацию обучающих выборок с целью избежания переобучения модели

4) разработать утилиту, позволяющую обучать модели-классификаторы изображений

5) протестировать обученный классификатор

6) протестировать разработанный модуль

5. Дата выдачи задания: «__»_____2020 г.

Руководитель ВКР _____ А.А. Федотов

Задание приняла к исполнению «__»_____2020 г.

Студент _____ Д.Ю. Еременко

РЕФЕРАТ

72 страниц, 45 рисунков, 3 таблицы, 19 источников, 1 приложений
МАШИННОЕ ОБУЧЕНИЕ, КЛАССИФИКАЦИЯ ИЗОБРАЖЕНИЙ,
СВЕРТОЧНЫЕ НЕЙРОННЫЕ СЕТИ, U-NET, ЗАБОЛЕВАНИЯ РАСТЕНИЙ

Данная работа посвящена решению проблемы автоматизации своевременного обнаружения на полях больных растений с целью сохранения урожая.

В процессе выполнения работы выполнено сравнение существующих методов классификации изображений и в качестве наиболее подходящего метода выбран метод сверточных нейронных сетей.

Разработано программное обеспечение, позволяющее формировать обучающие выборки по фотографиям полей, производить аугментацию обучающих выборок с целью избежания переобучения модели, обучать модели-классификаторы изображений и использовать реализованные модели-классификаторы.

При помощи разработанного программного обеспечения с привлечением специалистов по защите растений подготовлена обучающая выборка состоящая из 407 примеров фитофтороза, 344 примеров здорового куста, 501 примера мозаики картофеля и 288 примеров сорняков. Выборка была расширена при помощи алгоритмов генерации изображений.

Предобученная модель DenseNet121 показала наилучший результат на исходных данных и была дообучена на данных с аугментацией. Точность обученного классификатора на экзаменационных данных с аугментацией составила 98-99 процентов.

ABSTRACT

72 pages, 45 figures, 3 tables, 19 sources, 1 appendices

**MACHINE LEARNING, IMAGE CLASSIFICATION, CONVOLUTIONAL
NEURAL NETWORKS, U-NET, PLANT DISEASES**

This work is devoted to solving the problem of automation of timely detection of diseased plants in the fields in order to preserve the crop.

In the course of this work was compared existing methods of image classification and selected the convolutional neural network method as the most appropriate method.

Software has been developed that allows forming training samples based on field photos, augmenting training samples in order to avoid over-training the model, training image classifier models, and using implemented classifier models.

A training sample consisting of 407 examples of late blight, 344 examples of healthy Bush, 501 examples of potato mosaic and 288 examples of weeds was prepared using the developed software with the involvement of plant protection specialists. The selection was expanded using image generation algorithms.

The pre-trained densenet121 model showed the best result on the source data and was further trained on the data with augmentation. The accuracy of the trained classifier on the exam data with augmentation was 98-99 percent.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	9
1. РАЗРАБАТЫВАЕМАЯ СИСТЕМА.....	14
1.1. Архитектура.....	14
1.2. Входные данные.....	16
2. ВЫБОР МЕТОДА КЛАСТЕРИЗАЦИИ.....	17
2.1. Sliding Window.....	17
3. ВЫБОР МЕТОДА КЛАССИФИКАЦИИ ИЗОБРАЖЕНИЙ.....	19
3.1. Методы.....	19
3.1.1. Maximum Likelihood.....	19
3.1.2. Minimum Distance.....	21
3.1.3. Parallelepiped.....	23
3.1.4. Mahalanobis Distance.....	25
3.1.5. K-Nearest Neighbor.....	26
3.1.6. Decision Tree.....	27
3.1.7. ANN.....	27
3.1.8. Image Segmentation.....	28
3.2. Критерии выбора метода.....	29
3.3. Выбор метода классификации.....	31
4. ВЫБОР МЕТОДА СЕГМЕНТАЦИИ И ПРЕДОБРАБОТКИ.....	32
4.1. U-Net.....	32
4.2. Обучение Unet.....	34
5. ОБУЧАЮЩАЯ ВЫБОРКА.....	39
5.1. Формирование.....	39
5.2. Аугментация.....	44
6. ИТОГОВЫЙ СТЕК МЕТОДОВ.....	46
7. ВЫБОР БИБЛИОТЕКИ ДЛЯ ГЛУБОКОГО ОБУЧЕНИЯ.....	47
8. МОДУЛЬ КЛАССИФИКАЦИИ.....	48
8.1. Модуль поиска кандидатов.....	48
8.2. Тестирование модуля поиска кандидатов.....	49
8.3. Обучение классификатора.....	50
8.3.1. Выбор предобученных моделей.....	50
8.3.2. Утилита для обучения сверточных нейронных сетей.....	51
8.3.3. Обучение Xception.....	56
8.3.4. Обучение MobileNetV2.....	58

8.3.5. Обучение DenseNet121.....	60
8.3.6. Дообучене DenseNet121 на выборке с аугментацией.....	62
9. ТЕСТИРОВАНИЕ МОДУЛЯ.....	64
ЗАКЛЮЧЕНИЕ.....	67
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	69
ПРИЛОЖЕНИЕ ССЫЛКА НА ПРОЕКТ НА GITHUB.....	72

ВВЕДЕНИЕ

Искусственный интеллект – одна из самых популярных и быстро развивающихся областей науки на сегодняшний день. Алгоритмы искусственного интеллекта используются почти во всех сферах нашей жизни.

Одной из самых популярных задач, решаемых при помощи искусственного интеллекта, является задача классификации. Задача классификации заключается в сопоставлении входных данных одному или нескольким непересекающимся множествам. Входными данными в задаче классификации могут быть изображения, аудиоданные и любые другие данные, которые можно представить в виде цифровой выборки. Задача классификации является весьма актуальной в разных сферах жизни.

Так, например, в медицине - ученые петербургского Политеха вместе с врачами онкоцентра разработали интеллектуальную систему диагностики опухолей в легких, способную распознавать опухоли по компьютерной томографии, а также отличать злокачественные опухоли от доброкачественных.[1]

Также нейронные сети используются для обеспечения безопасности в сфере дорожного движения. Так, например, в статье [2] ученые из google разработали систему, позволяющую детектировать пешеходов с точностью 78 процентов и работающую в режиме реального времени (детектирование происходит менее чем за 0.07 секунд).

Сельское хозяйство — еще одна сфера, требующая внедрения новых технологий. Агробизнес в России достиг определенной зрелости, о чем свидетельствуют стабилизация уровня инвестиций в сельское хозяйство и рост конкуренции среди производителей сельскохозяйственной продукции. В АПК растёт объем и качество применения современных технологий, в том числе

систем сбора, хранения и обработки данных. Применяются данные со спутников, датчиков, из операционных и транзакционных систем. При этом увеличивается как объем данных, так и потребность в их качественной обработке и достоверных выводах, на которые можно полагаться, принимая решения. В результате оформляется спрос на промышленные аналитические системы и, в частности, углубленную аналитику.[15]

Современные технологии дали человеческому обществу возможность производить достаточное количество продовольствия для удовлетворения потребностей более чем 7 миллиардов человек. Однако продовольственная безопасность по-прежнему находится под угрозой из-за ряда факторов, включая изменение климата (Tai et al., 2014), снижение численности опылителей (доклад Пленума межправительственной научно-политической платформы по биоразнообразию экосистемы и услуг о работе ее четвертой сессии, 2016), болезни растений (Strange and Scott, 2005) и другие. Болезни растений представляют собой не только угрозу продовольственной безопасности в глобальном масштабе, но и могут иметь катастрофические последствия для мелких фермеров, чьи средства к существованию зависят от здоровых культур. В развивающихся странах более 80% сельскохозяйственной продукции производится мелкими фермерами (UNEP, 2013), и сообщения о потере урожая более чем на 50% из-за вредителей и болезней являются распространенными (Harvey et al., 2014). Кроме того, самая большая часть голодающих людей (50%) живет в мелких фермерских хозяйствах (Sanchez and Swaminathan, 2005), что делает мелких фермеров группой, которая особенно уязвима к нарушениям в поставках продовольствия, вызванным патогенами.

Одной из причин резкого снижения урожайности картофеля является поражение растений вирусными болезнями. Наиболее вредоносные среди них — полосатая, морщинистая и обыкновенная мозаики, скручивание и мозаичное

закручивание листьев, аукуба-мозаика и др. Особая опасность вирусных заболеваний картофеля в том, что больные растения не поддаются лечению, а возбудители накапливаются в последующих вегетативных поколениях клубней и являются причиной «вырождения» культуры. Вирусные и виroidные болезни картофеля являются причиной огромных недоборов урожая клубней. Степень снижения урожайности зависит от вида вируса, а также сорта и условий выращивания культуры.

При поражении посадок вирусом 5 недобор урожая может достигать 15-20%, вирусом X - от 10-15% до 45-50% в зависимости от штамма. Вредоносность вируса Y также варьируется в широких пределах — от 15-20% до почти полного отсутствия урожая, плюс резко снижается товарность клубней больных растений. Вирус M (мозаичное закручивание листьев) обычно более вредоносен, чем вирусы X и S: снижение урожая составляет 15-45%, а содержание крахмала в клубнях уменьшается на 2-3%. Снижение продуктивности растений от вируса L (скручивание листьев) может достигать 50-70%, а от виroidа веретеновидности — 85%.

При комплексных вирусных инфекциях недоборы урожая еще ощутимее. Например, при одновременном поражении растений вирусами X и M урожайность снижается на 55-60%, а вирусами X, M и Y — на 80% и более, т. е. инфицированные растения урожая практически не дают.[13]

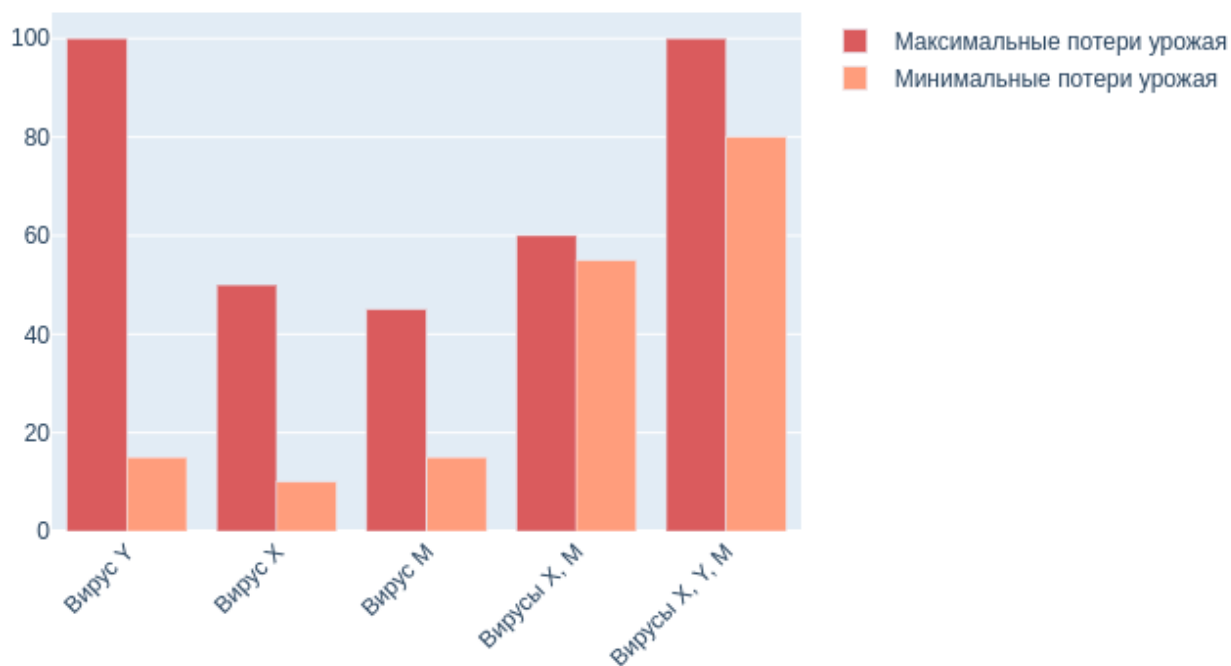


Рис. В.1. график вредоносности разных болезней

Таким образом, можно сделать вывод, что существует необходимость своевременного обнаружения на полях больных растений для принятия мер по их изоляции от остальных растений. Для быстрого поиска больных растений по видимым признакам можно использовать фотографии поля, сделанные с беспилотника. В роли детектора может выступать специалист-агроном, однако у такого подхода есть несколько очевидных минусов:

- при выполнении алгоритмической работы человек теряет концентрацию и качество выполнения работы заметно падает
- человек уступает компьютеру в скорости детектирования, особенно на больших данных

Система автоматического детектирования больных растений здесь подошла бы куда лучше.

Целью данной бакалаврской работы является разработка программного модуля выявления зараженных растений с использованием методов машинного обучения с учителем.

Для достижения поставленной цели необходимо решить следующие задачи:

- разработать графическую утилиту, позволяющую отмечать на изображениях больные участки и сохранять результаты разметки для дальнейшего формирования обучающих выборок
- разработать графическую утилиту, позволяющую производить аугментацию исходной выборки изображений
- разработать утилиту, позволяющую производить обучение и тестирование классификатора
- разработать графическую утилиту, позволяющую тестировать разработанный модуль

На первый взгляд может показаться, что бакалаврская работа повторяет магистерскую работу [18], однако стоит заметить, что ключевым отличием целевой системы является ее способность производить поиск больных растений на полях по фотографиям сделанным с беспилотника в уличных условиях. То есть ориентирована целевая система в первую очередь на автоматизированную обработку больших объемов данных при минимальном участии человека.

1. РАЗРАБАТЫВАЕМАЯ СИСТЕМА

1.1. Архитектура

На основе поставленной задачи сформируем архитектуру целевой системы. Система состоит из 4 модулей:

- модуль кластеризации — модуль, преобразующий изображение поля в набор изображений кустов
- модуль сегментации и предобработки — модуль, занимающийся усилением интересующих видимых признаков распознаваемых классов растений
- модуль классификации — модуль сопоставляющий изображению класс растения
- модуль построения карты — выходит за рамки бакалаврской.

Архитектура целевой системы представлена на рисунке 1.1. Дальнейшее повествование строится на последовательном описании реализации каждого модуля.

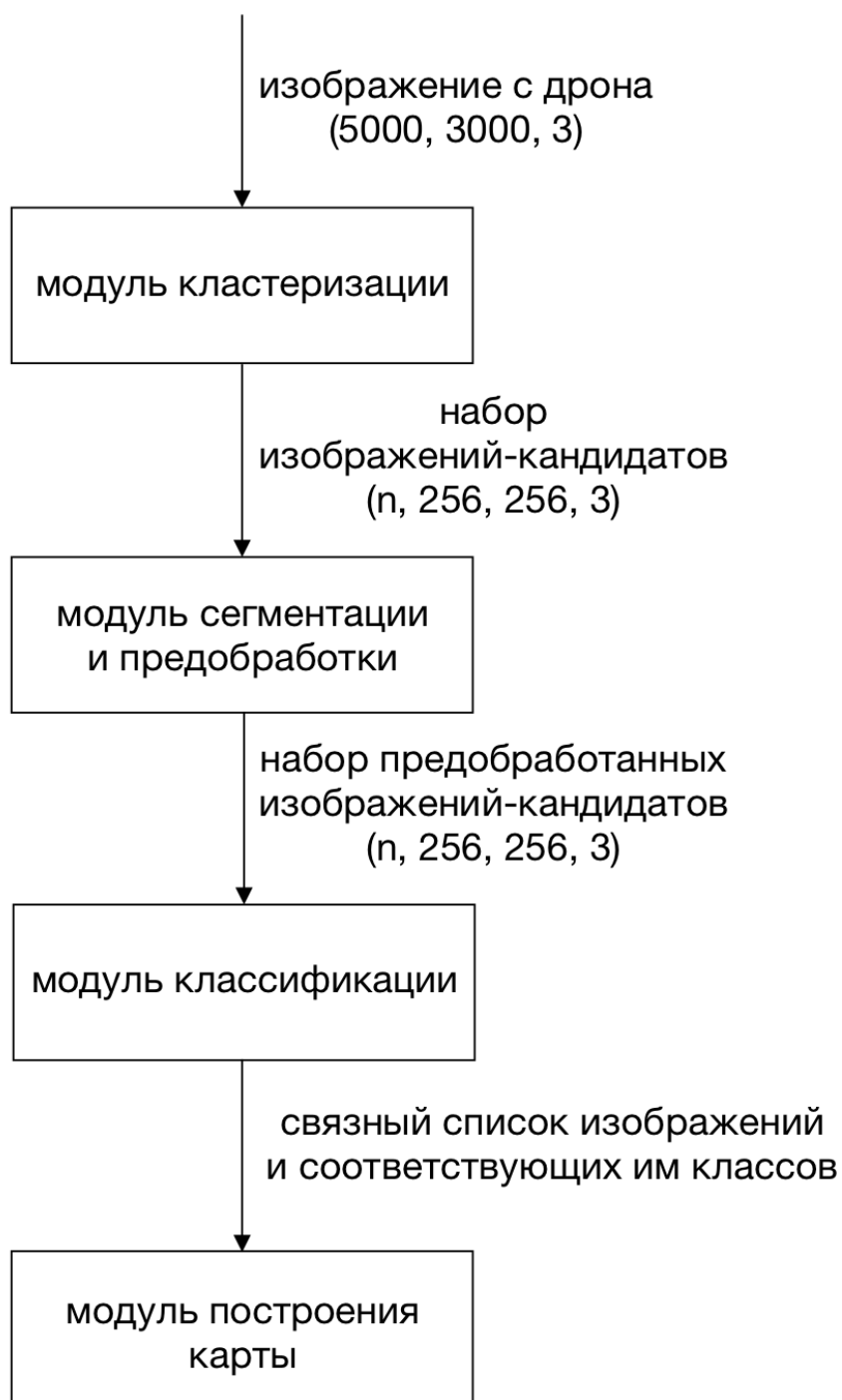


Рис.1.1. Архитектура системы

1.2. Входные данные

Входные изображения делаются беспилотником, пролетающим над полем на высоте 7 метров. Данная высота была выбрана исходя из полевых экспериментов — именно на фотографиях сделанных с такой высоты эксперт, формирующий обучающую выборку может распознать болезни по видимым признакам.

Размер входного изображения равен (5280, 3956) пикселей. Размер фотографии обоснован разрешением камеры на имеющемся беспилотнике. Разрабатываемое программного обеспечение способно обрабатывать изображения любого размера.



Рис. 1.2. Пример входного изображения

2. ВЫБОР МЕТОДА КЛАСТЕРИЗАЦИИ

Поскольку все входное изображение состоит из кандидатов для распознавания – кластеризацию реализуем при помощи алгоритма “sliding window” с размером окна равным (256, 256, 3) и перекрытием равным 0 %. Размер окна был выбран на основании высоты съемки.

2.1. Sliding Window

Алгоритм:

- на вход алгоритму подается большое изображение
- создается окно необходимого размера (размер должен быть меньше размера исходного изображения)
- ставим окно на позицию (0, 0)
- вырезаем изображение часть изображения, попадающую в окно
- перемещаем окно слева направо сверху вниз, пока не выйдем за пределы координаты y

Таким образом, исходное изображение размером (5280, 3956) пикселей делится на 300 изображений размером (256, 256) пикселей.

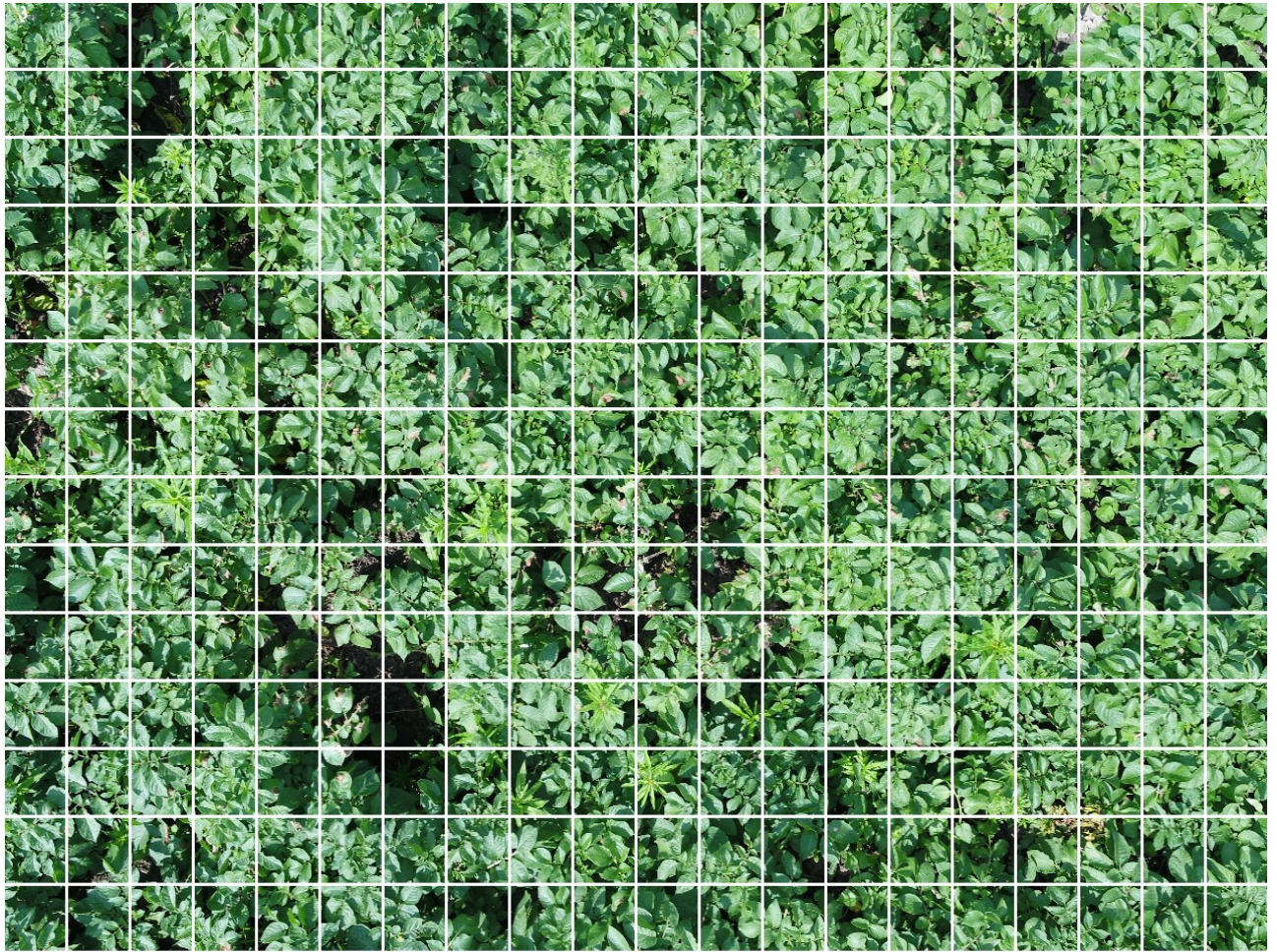


Рис. 2.1. Кластеризация исходного изображения при помощи алгоритма Sliding Window

3. ВЫБОР МЕТОДА КЛАССИФИКАЦИИ ИЗОБРАЖЕНИЙ

3.1. Методы

На рисунке 3.1 из статьи [12] предложены следующие методы классификация изображений, сделанных со спутника.

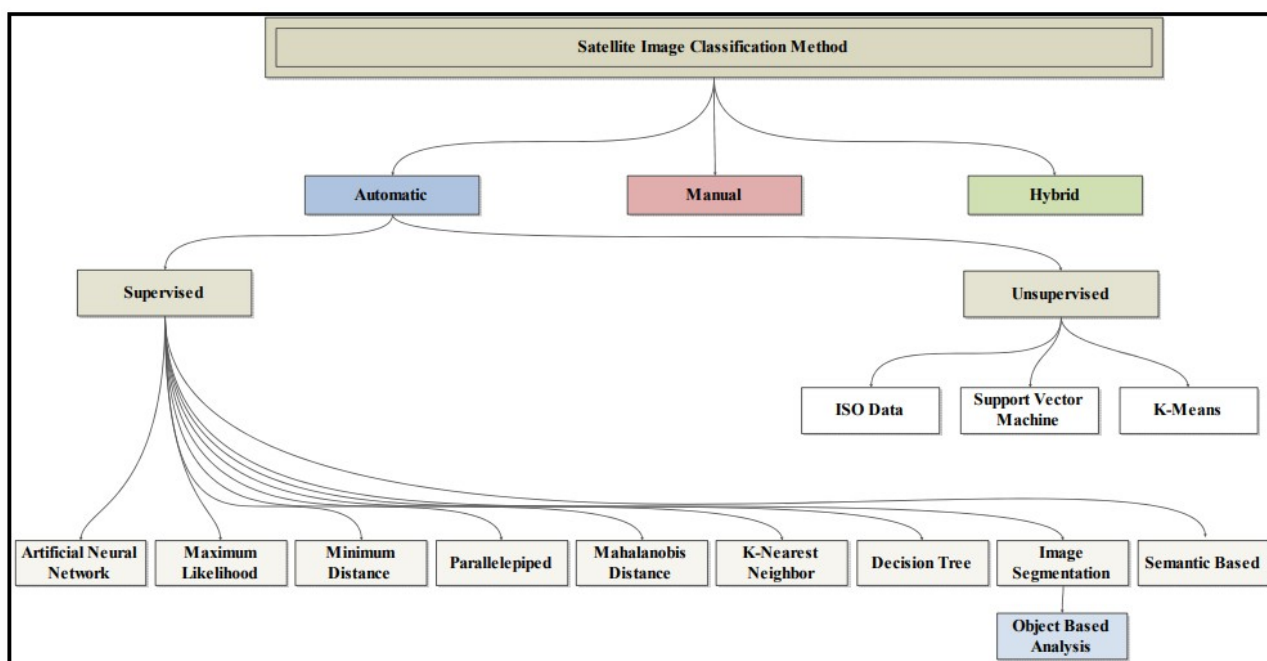


Рис. 3.1. Методы классификации изображений

Рассмотрим предложенные методы.

3.1.1. Maximum Likelihood

Описание: [5]Метод классификации максимального правдоподобия применяет теорию вероятностей к задаче классификации.

- из классов обучающего набора вычисляются центры классов и вариативность значений раstra в каждом входном диапазоне для каждого класса. Эта информация позволяет классификатору определить вероятность того, что данному изображению во входном наборе

принадлежит конкретный учебный набор. Вероятность этого зависит от расстояния от изображения до центра класса,

- метод максимального правдоподобия вычисляет вероятности для каждого класса и относит пример к классу с наибольшим значением вероятности.

Плюсы:

- дает более точный результат чем Minimum Distance, когда изображения значительно различаются по размеру и форме
- предсказание происходит за $O(1)$
- может подойти для сегментации изображения
- пример сегментации при помощи алгоритма можешь увидеть на рисунке 3.2

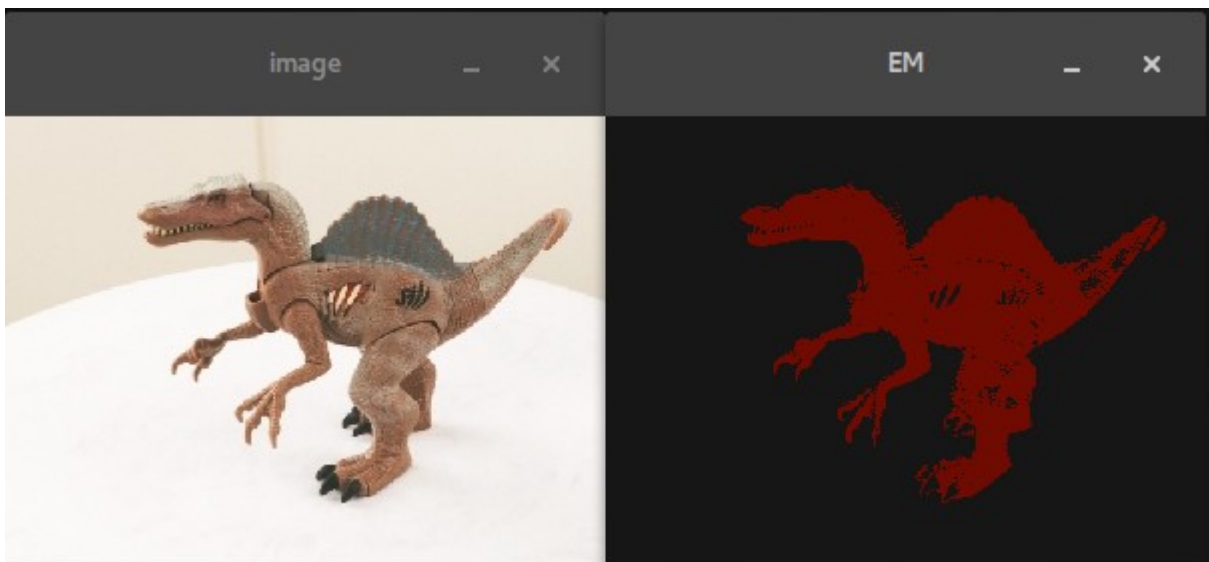


Рис. 3.2. Пример сегментации изображения при помощи алгоритма Maximum Likelihood

Минусы:

- не устойчив к изменению входных данных

3.1.2. Minimum Distance

Описание:[7]Рассмотрим простейший случай в 2-мерном спектральном пространстве. Оси соответствуют спектральным полосам изображения. Каждый пиксель изображения соответствует определенной точке в пространстве объектов. На рисунке 3.3 показаны три класса, которые находятся в красной, зеленой и синей точках. Красное точечное облако накладывается на зеленое и синее. Существует также облако черных точек, которое не принадлежит ни к одному классу. После классификации изображения эти точки будут соответствовать классифицированным пикселям.

Процесс классификации:

- находим среднее значение по всем цветам для каждого класса в выборке
- определяем максимальное расстояние до каждого класса
- изображения представляются как точки в пространстве каналов.

Определяем класс, к которому изображение ближе всего и относим изображение к этому классу, если расстояние до его средних параметров меньше заданного.

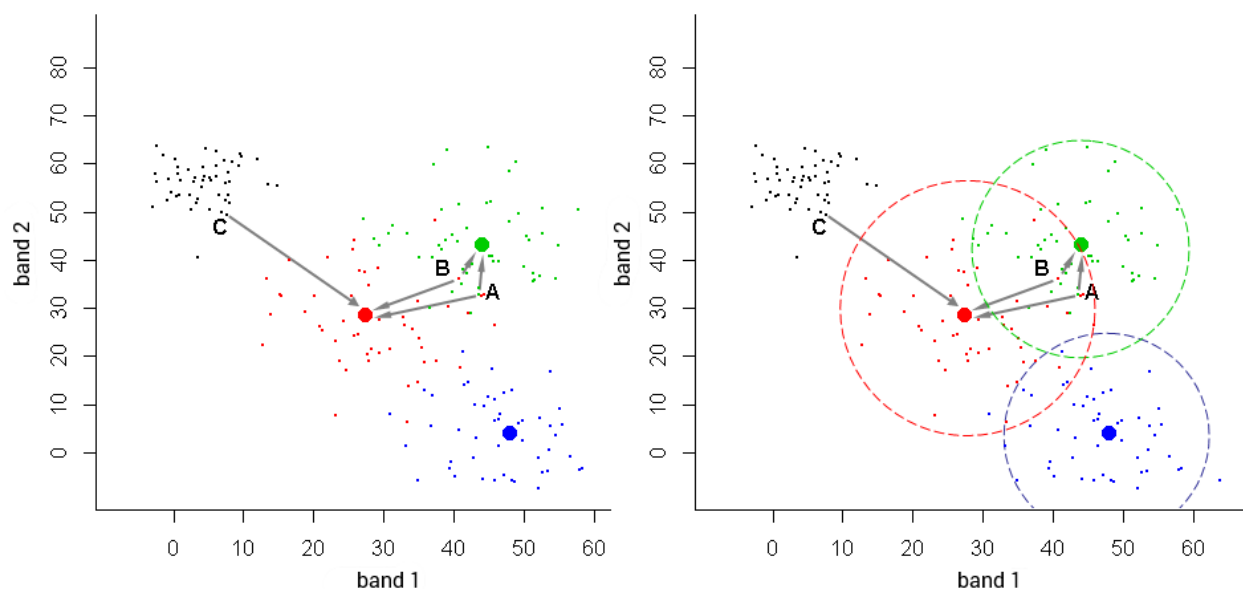


Рис. 3.3. Визуализация работы алгоритма Minimum Distance для двух каналов

Также алгоритм можно использовать для пост-анализа ответов других алгоритмов. Например, у нас есть нейронная сеть, способная классифицировать изображения трех классов — А, В и С. Мы подаем на ее вход изображения, а на выходе получаем массив [0.25, 0.75, 0.25]. Согласно алгоритму Minimum Distance, входное изображение будет отнесено к классу В.

Плюсы:

- требует предварительных расчетов
- можно использовать для пост анализа

Минусы:

- не устойчив к изменению входных данных при использовании в качестве классификатора

3.1.3. Parallelepiped

Описание: [7] Рассчитываем среднее значение яркости и стандартное отклонение от среднего значения по всем каналам изображения.

Процесс классификации:

- устанавливаем центры для каждого класса в пространстве спектральных признаков
- находим экстремальные точки для каждого класса
- прокладываем линии от центров классов, параллельно осям пространства объектов. Длина линий равна $N * \text{stddev}$ - где N — варьируемый параметр для каждого класса
- изображения представляются как точки в пространстве каналов. Если изображение попадает в параллелепипед соответствующего класса— относим его к этому классу. Ниже приведен пример использования алгоритма, если во входных данных имеется лишь 2 канала.

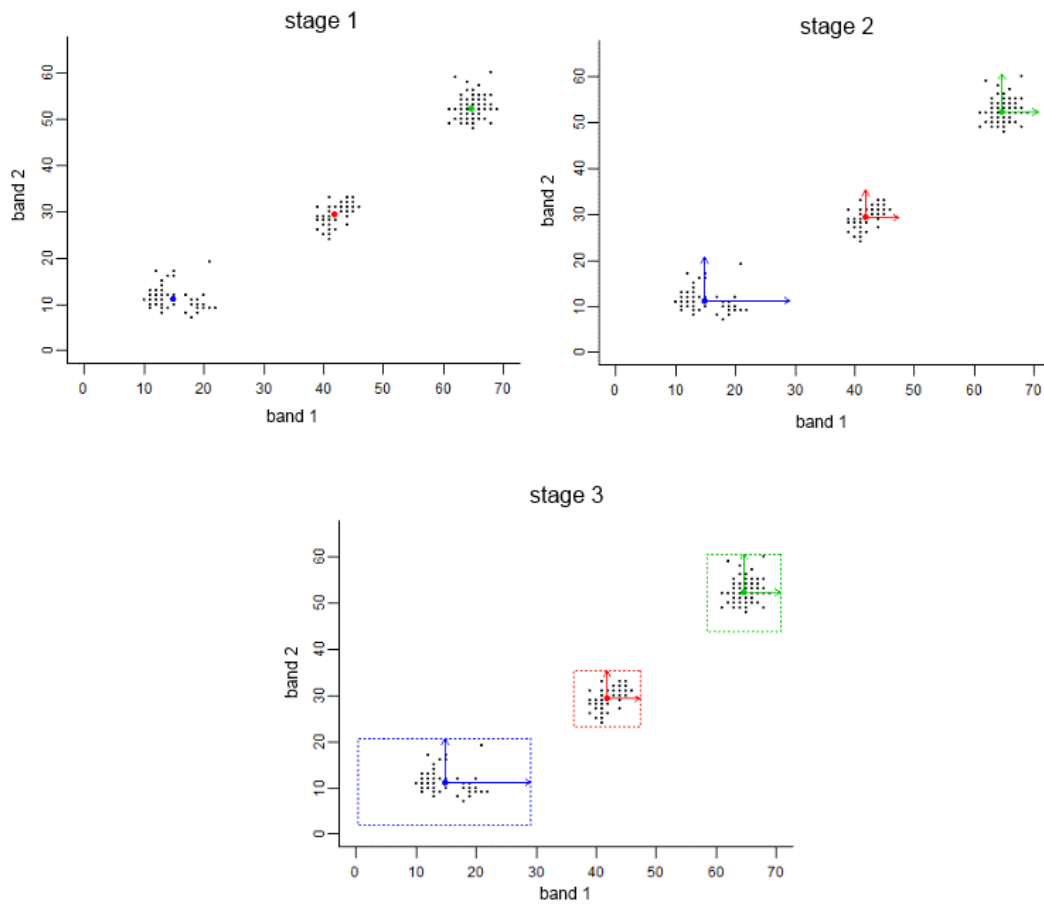


Рис. 3.4. Визуализация работы алгоритма Parallelepiped для двух каналов

Плюсы:

- требует наличия тренировочной выборки
- требует расчета статистических параметров выборки
- предсказание происходит за $O(1)$

Минусы:

- не устойчив к изменению входных данных

3.1.4. Mahalanobis Distance

Описание: [10]Эффективная многомерная метрика расстояния, которая измеряет расстояние между точкой (вектором)и распределением. В отличие от Евклидова расстояния учитывает корреляцию каналов во входных данных. Так, например, на рисунке 3.5 справа Евклидово расстояние для Point1 и Point2 равно, однако точка находится ближе к кластеру.

Изображение 3.5 справа является простым примером положительной корреляции двух переменных «x» и «у» — при увеличении значения переменной «x» увеличивается значение и другой переменной «у».

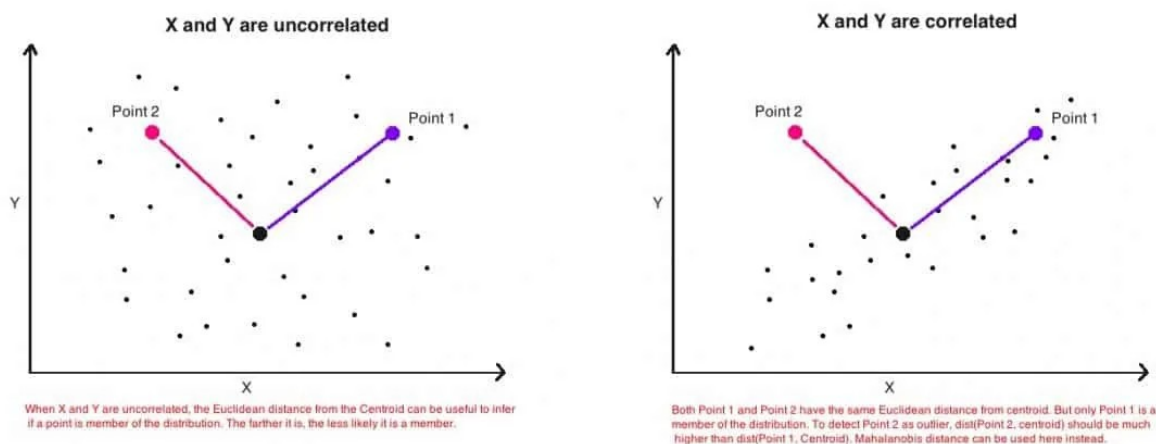


Рис. 3.5. Пример неправильной работы Евклидова расстояния в качестве метрики

Считается расстояние Махаланобиса следующим образом:

$$D^2 = (x - m)^T \cdot C^{-1} \cdot (x - m)$$

Рис. 3.6. формула расстояния Махаланобиса

- D — расстояние Махаланобиса
- x — каналы в исходных данных
- m — вектор средних значений в выборке
- C — обратная ковариационная матрица независимых переменных

Плюсы:

- подходит для классификации по сильно несбалансированным наборам данных
- подходит для классификации одного класса
- учитывает корреляцию каналов входных данных
- не требует обучения

Минусы:

- не устойчив к изменению входных данных

3.1.5. K-Nearest Neighbor

Описание: [7] находим K -ближайших соседей, например, используя MAE (mean absolute error) и смотрим распределение в полученной выборке.

Плюсы:

- не требует обучения

Минусы:

- высокая трудоемкость предсказания (предсказание происходит за $O(N)$, где N – размер обучающей выборки)
- не устойчив к изменению входных данных

3.1.6. Decision Tree

Описание: абстрактный алгоритм учета результатов нескольких классификаторов, не подходит для классификации только по видимым признакам, так как требует самостоятельного извлечения признаков. Можно использовать, как пост-анализатор, на который поступают предсказания с классификатора изображений и дополнительная мета информация, например информация о соседях.

3.1.7. ANN

Описание: [9] семейство алгоритмов, моделирующих поведение нейронных сетей при решении той или иной задачи (CNN, MLP, RNN). Для классификации изображений обычно используются CNN сети, поскольку:

- по сравнению с MLP, имеет гораздо меньшее количество настраиваемых весов, так как одно ядро весов используется целиком для всего изображения, вместо того, чтобы делать для каждого пикселя входного изображения свои персональные весовые коэффициенты. Это подталкивает нейронную сеть при обучении к обобщению демонстрируемой информации, а не попиксельному запоминанию каждой показанной картинке, как это делает перцептрон.
- Удобное распараллеливание вычислений, а следовательно, возможность реализации алгоритмов работы и обучения сети на графических процессорах.
- Относительная устойчивость к повороту и сдвигу распознаваемого изображения.
- Обучение при помощи классического метода обратного распространения ошибки.

Плюсы:

- при наличии обучающей выборки не требует глубокого понимания предметной области
- низкая трудоемкость предсказания($O(1)$)
- производит классификацию, анализируя объекты на картинке

Минусы:

- требуется время для обучения

3.1.8. Image Segmentation

Описание: семейство алгоритмов предобработки изображений (U-Net, Edge Detection, Mask R-CNN). Подходит для предобработки исходного изображения, перед передачей классификатору.



Рис. 3.7. Пример сегментации исходного изображения при помощи U-Net

Плюсы:

- позволяет усилить интересующие видимые признаки

Минусы:

- некоторые реализации требуют обучения

3.2. Критерии выбора метода

- *Способность предсказания исключительно по видимым признакам* - входными данными классификатора является изображение в формате RGB, следовательно в первую очередь нас интересуют алгоритмы классификации изображений по видимым признакам
- *Низкая трудоемкость предсказания* - поскольку система будет работать с большими объемами данных (необходимо примерно 300 предсказаний на 1 фотографию с беспилотника) скорость работы алгоритма является одним из самых важных критериев.
- *Устойчивость к изменениям входных данных* - одной из ключевых проблем алгоритмов классификации изображений является их неустойчивость к следующим изменениям входных изображений:
 - Изменение перспективы(viewpoint variation) - один экземпляр объекта может быть ориентирован по отношению к камере многими способами.
 - Изменение масштаба(scale variation) - визуальные классы часто проявляют вариативность в своих размерах.
 - Деформация(deformation) - многие объекты интереса не являются твердыми телами и могут быть деформированы в экстремальных условиях.
 - Окклюзия(occlusion) - объекты интереса могут быть закрыты. Иногда только небольшая часть объекта (всего лишь несколько пикселей) может быть видна.
 - Условия освещения(illumination conditions) - эффекты освещения очень сильны на уровне пикселей.

- Помехи(background clutter) - объекты интереса могут сливаться с окружающей средой, что затрудняет их идентификацию.

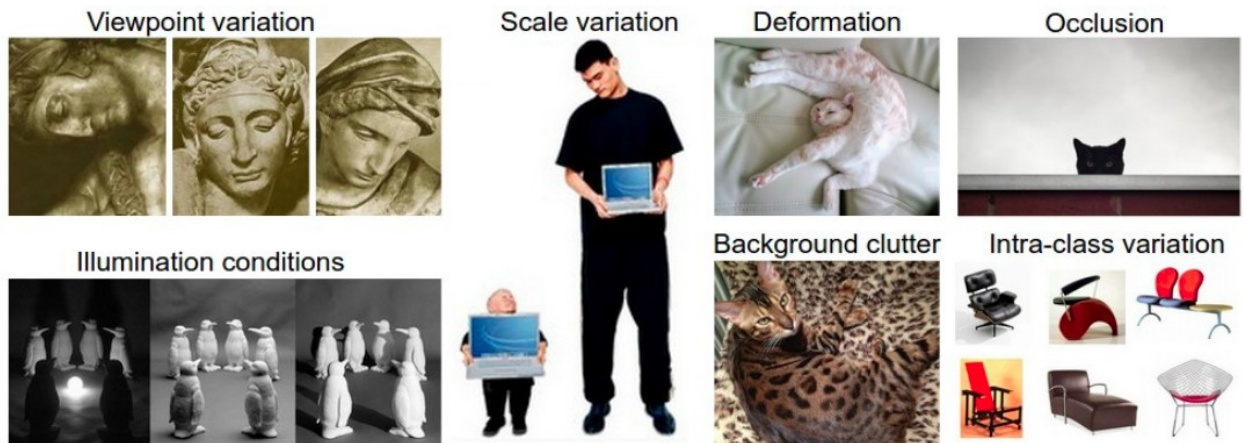


Рис. 3.8. Примеры изменений входных данных

3.3. Выбор метода классификации

Большинство рассмотренных алгоритмов классификации предполагают, что рассматриваемые классы различаются по статистическим параметрам, однако в решаемой задаче основным различием между классами являются контуры объектов(листов). Таким образом основным критерием при выборе метода классификации изображений была возможность различия изображений по контурам. Наиболее подходящим методом для решения такой задачи являются сверточные нейронные сети.

4. ВЫБОР МЕТОДА СЕГМЕНТАЦИИ И ПРЕДОБРАБОТКИ

4.1. U-Net

[6] Наиболее современным методом сегментации изображения является использование нейронной сети U-Net.

U-Net считается одной из стандартных архитектур CNN для задач сегментации изображений, когда нужно не только определить класс изображения целиком, но и сегментировать его области по классу, т. е. создать маску, которая будет разделять изображение на несколько классов. Архитектура состоит из сужающегося пути для захвата контекста и симметричного расширяющегося пути, который позволяет осуществить точную локализацию.

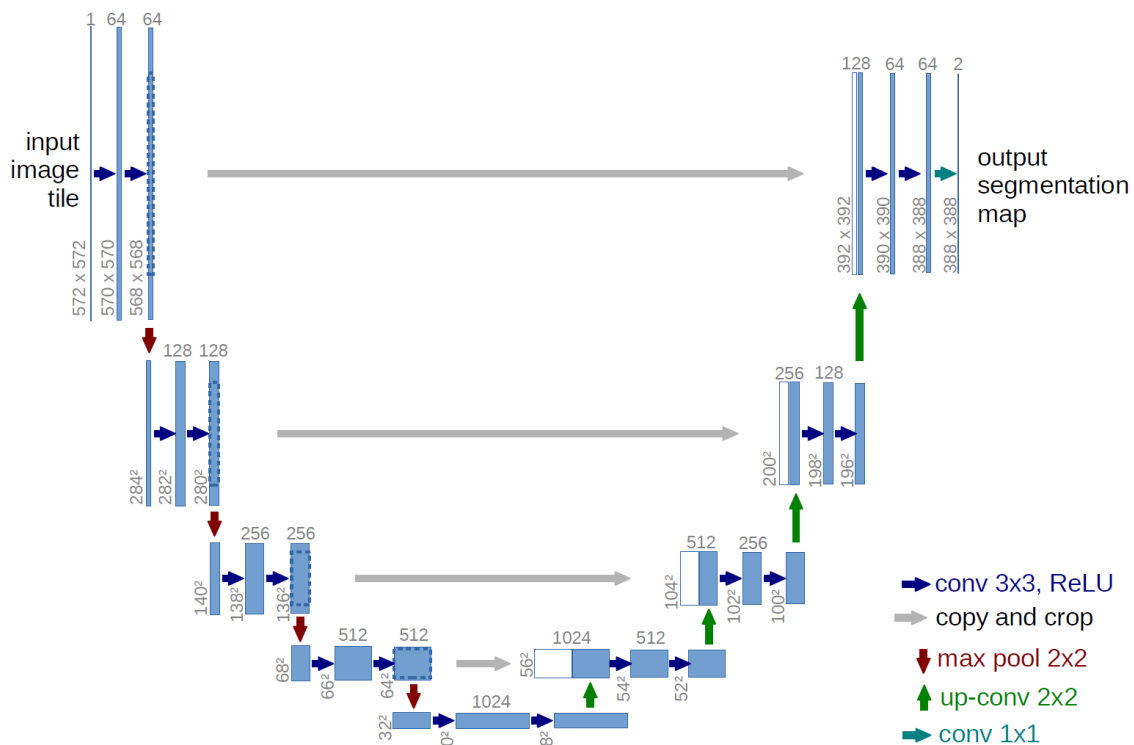


Рис. 4.1. Архитектура U-Net

Архитектура сети приведена на рисунке 1. Она состоит из сужающегося пути (слева) и расширяющегося пути (справа). Сужающийся путь — типичная архитектура сверточной нейронной сети. Он состоит из повторного применения

двух сверток 3×3 , за которыми следуют ReLU и операция максимального объединения (2×2 степени 2) для понижения разрешения.

На каждом этапе понижающей дискретизации каналы свойств удваиваются. Каждый шаг в расширяющемся пути состоит из операции повышающей дискретизации карты свойств, за которой следуют:

- свертка 2×2 , которая уменьшает количество каналов свойств;
- объединение с соответствующим образом обрезанной картой свойств из стягивающегося пути;
- две 3×3 свертки, за которыми следует ReLU.

Обрезка необходима из-за потери граничных пикселей при каждой свертке.

На последнем слое используется свертка 1×1 для сопоставления каждого 64-компонентного вектора свойств с желаемым количеством классов. Всего сеть содержит 23 сверточных слоя.

4.2. Обучение U-Net

Изначально, с помощью сети U-Net планировалась усилить интересующие видимые признаки (области фотографий, содержащие листья кустов) и скрыть не интересующие (области фотографий, не содержащие листья кустов), однако, позже стало понятно, что использование U-Net в стеке с CNN классификатором является избыточным решением, поскольку CNN классификатор также обучается выделять интересующие видимые признаки на скрытых слоях и выполняет ту же работу, что U-Net на своем сужающем пути.

Однако, в учебных целях, в ходе выполнения работы попытка обучить U-Net все-таки была совершена. Для обучения сети были сформированы 2 обучающие выборки и разработан скрипт, позволяющий контролировать обучения через консоль. Также для визуализации текущего результата обучения при помощи библиотеки PyQt5 был разработан графический интерфейс наглядно демонстрирующий результаты обучения U-Net. Каждая выборка состояла из 24 изображений 256*256 пикселей.

Первая выборка была сформирована с закрашиванием не интересующих областей красным цветом вручную, а интересующие области не закрашивались вовсе, поскольку информация о структуре поверхности листа является одним из ключевых видимых признаков, необходимых для классификации.



Рис. 4.2. Пример входного и размеченного вручную выходного изображения

Вторая выборка была сформирована с наложением на изображение фильтров. Все фильтры накладывались индивидуально.



Рис. 4.3. Пример входного и размеченного при помощи наложения фильтров выходного изображения

Добиться ожидаемого результата при обучении U-Net не удалось, при обучении модель возвращала изображение, в котором ненулевыми были значения лишь одного канала - R, G или B. Подобрать параметры для обучения, которые бы привели к лучшему результату не удалось. Результаты обучения можно увидеть на рисунках 4.4 и 4.5.

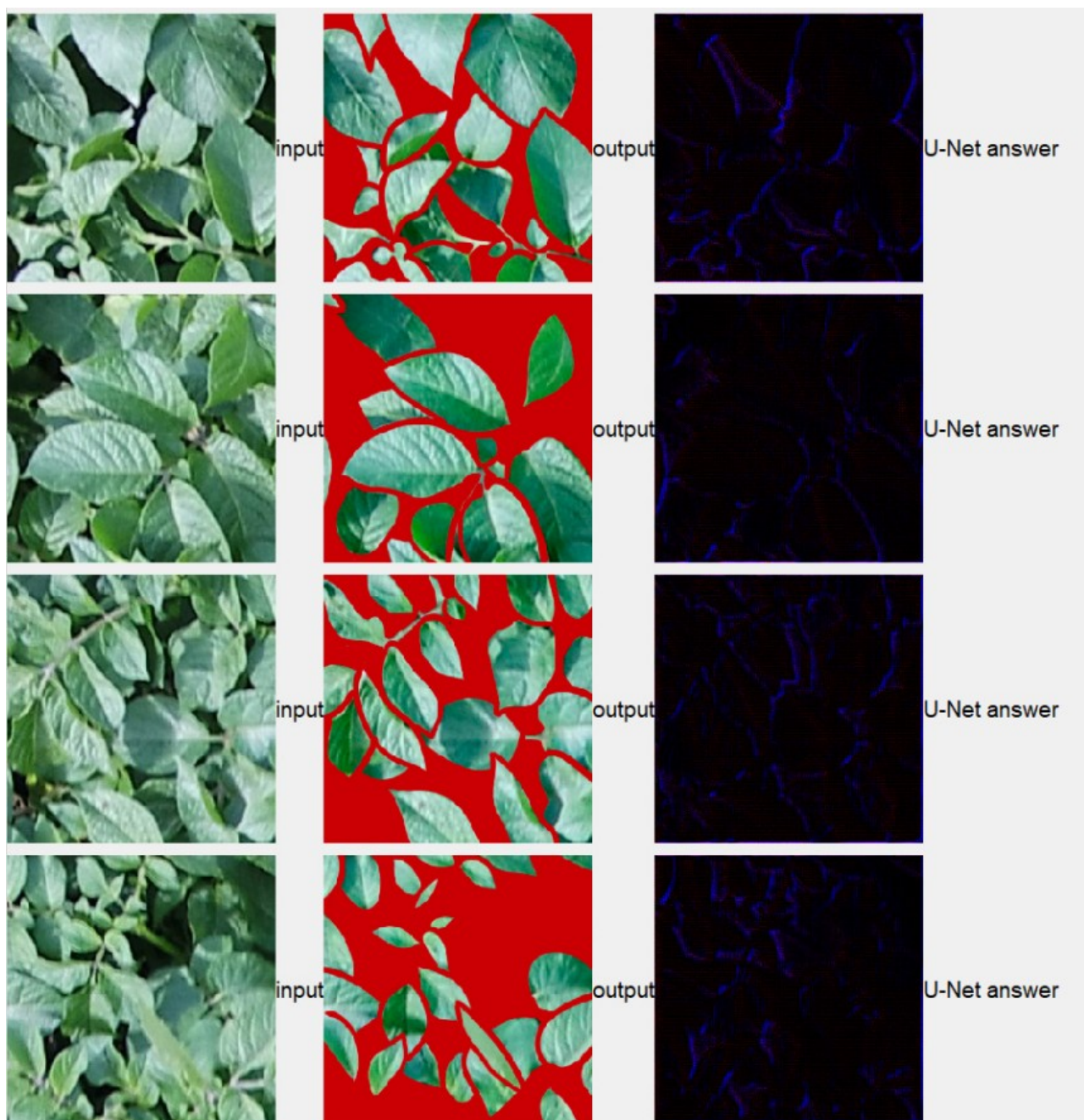


Рис. 4.4. Пример неудачного обучения U-Net на размеченных вручную данных

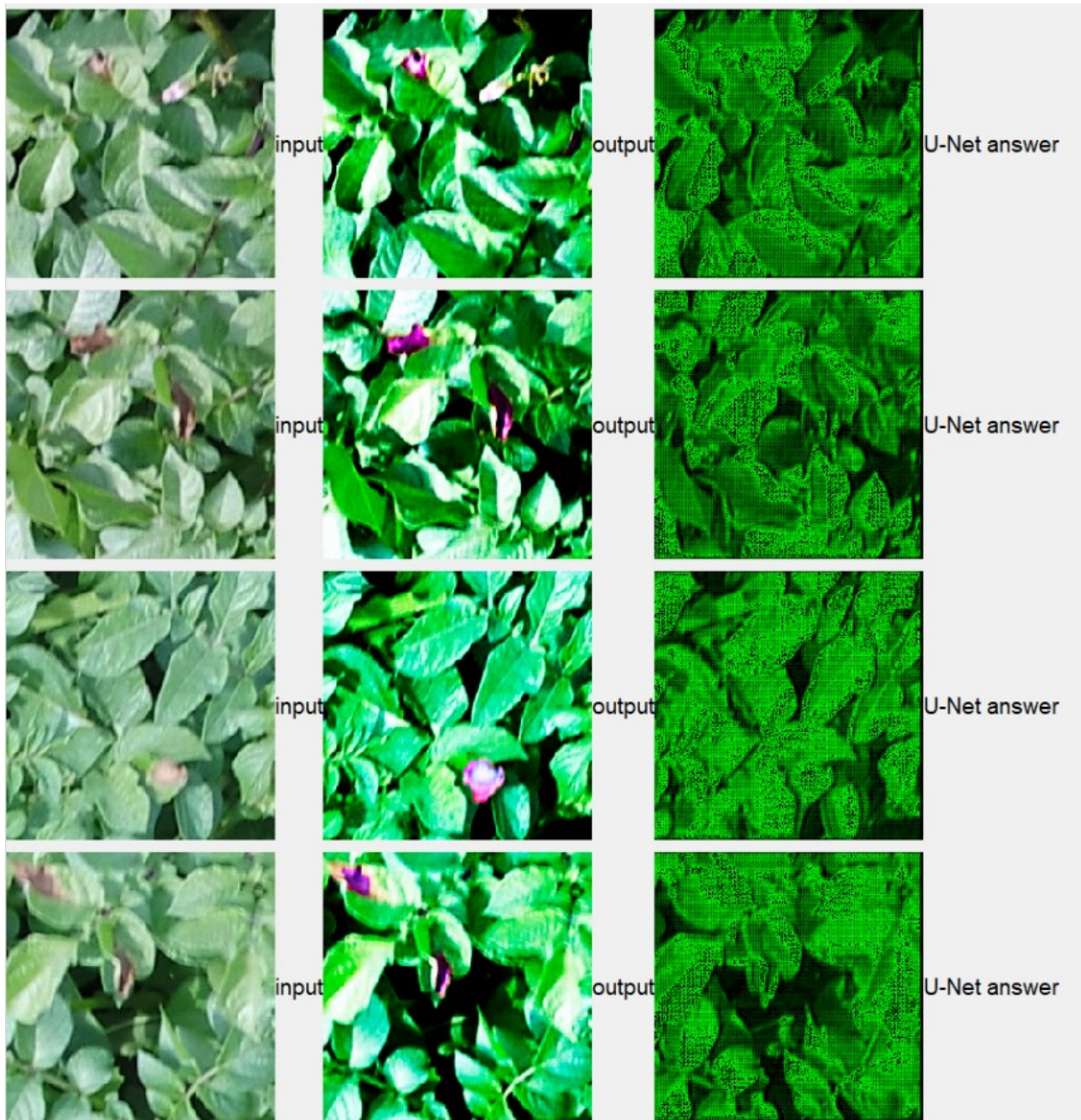


Рис. 4.5. Пример неудачного обучения U-Net на данных, размеченных при помощи фильтров

5. ОБУЧАЮЩАЯ ВЫБОРКА

5.1. Формирование

Для формирования обучающей выборки была разработана и передана специалистам по защите растений специализированная графическая утилита, позволяющая размечать фотографии сделанные с беспилотника и сохранять результаты разметки в файлы.

В приложении доступна разметка 26 видов болезней. Список болезней был получен от специалиста по защите растений. Список болезней хранится в конфигурационном json-файле.

На рисунке 5.1 приведен интерфейс разработанной графической утилиты. В утилите доступно использование 2-ух, 3-ех и 4-ех кратного зумирования.

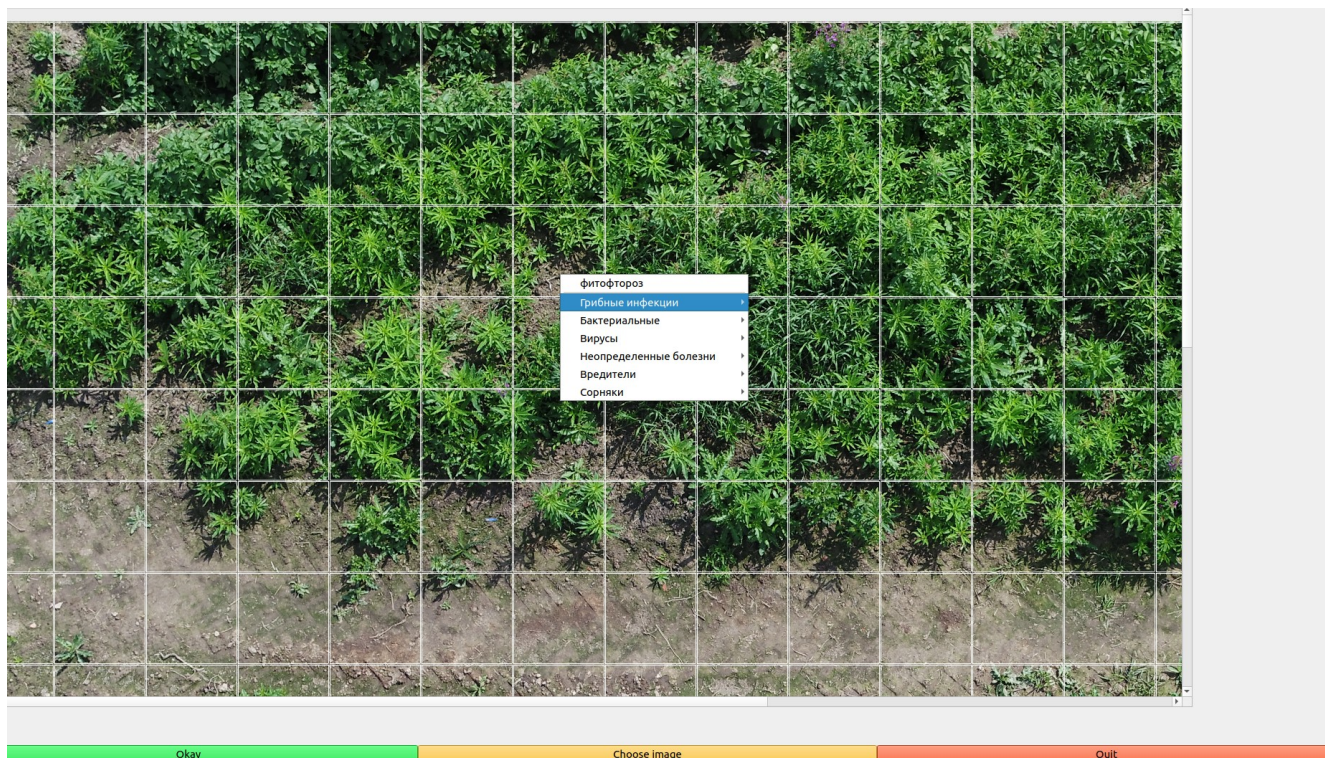


Рис. 5.1. Интерфейс графической утилиты для разметки полей

Рассмотрим полученную от специалистов по защите растений выборку.

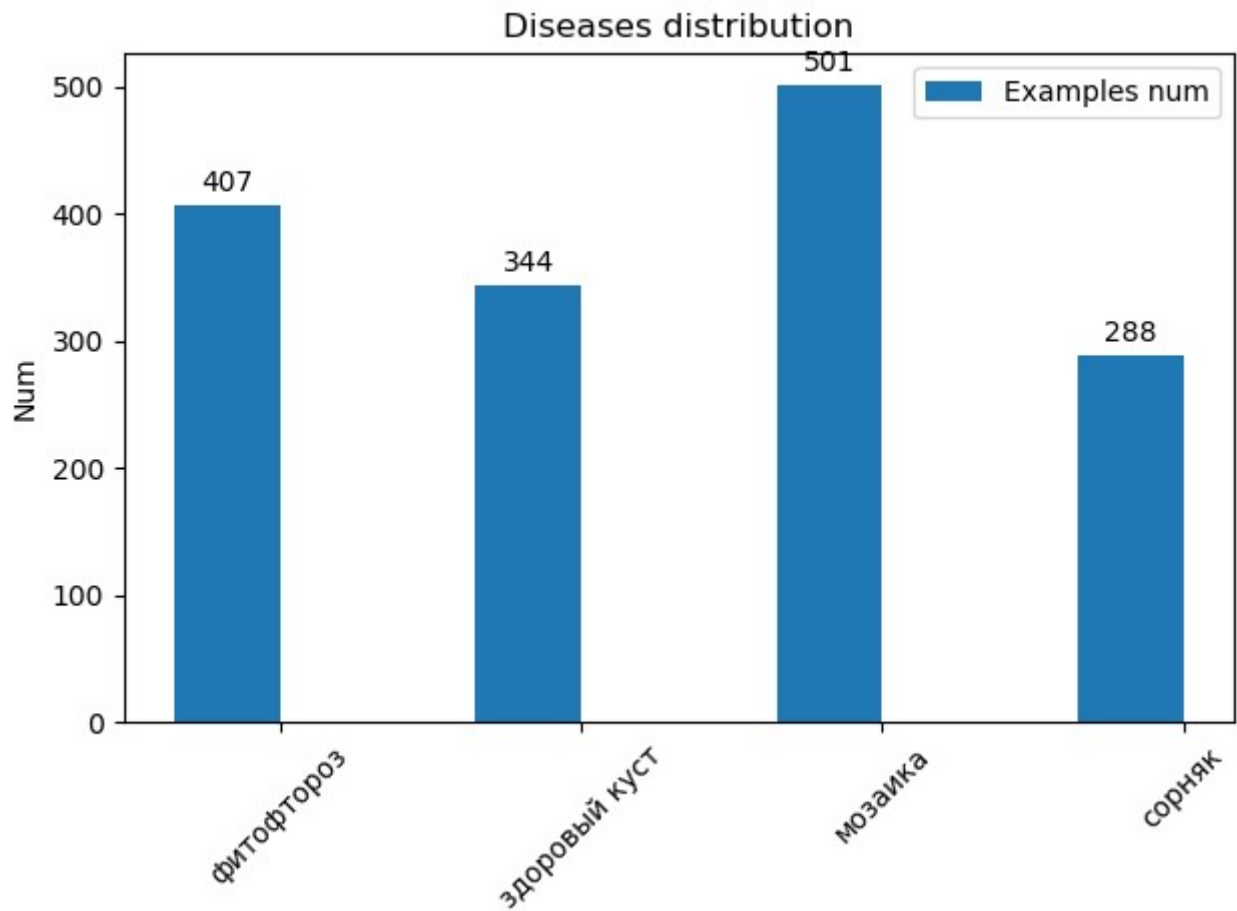


Рис. 5.2. Распределение примеров в обучающей выборке

На рисунках 5.3 — 5.6 отображены примеры каждого распознаваемого класса кустов.

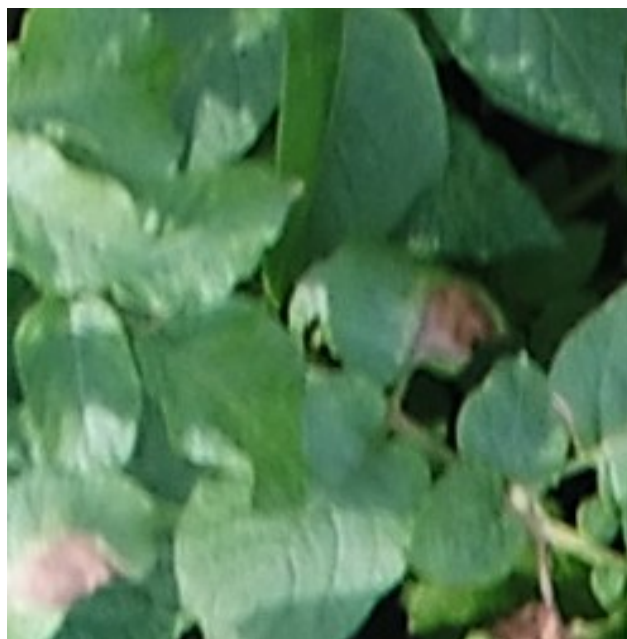


Рис. 5.3. Пример фитофтороза

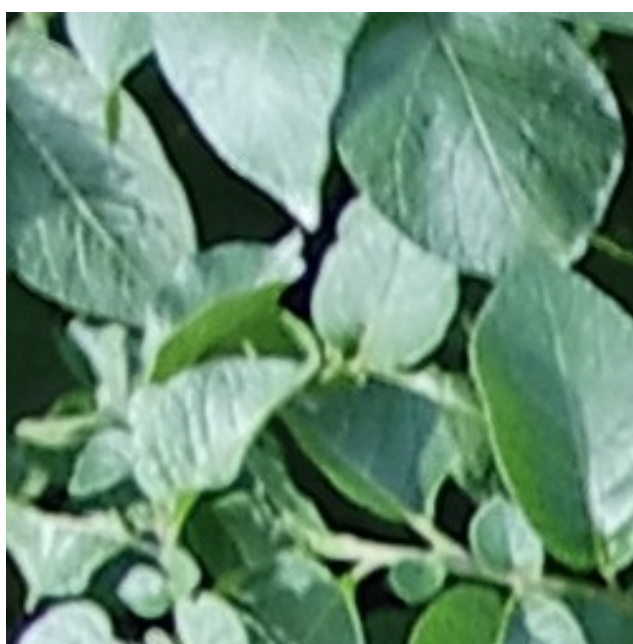


Рис. 5.4. Пример здорового куста



Рис. 5.5. Пример мозаики



Рис. 5.6. Пример сорняка

Все изображения классов, находящиеся в выборке удалось успешно отобразить в виде изображений в формате RGB, также все примеры классов можно отличить по видимым признакам, следовательно обучающая выборка была сформирована корректно.

5.2. Аугментация

Аугментация данных — методика создания дополнительных обучающих данных из имеющихся данных.

В библиотеках обучения нейронных сетей существуют генераторы данных, позволяющие случайным образом в процессе обучения применять к изображениям в исходной выборке такие преобразования как:

- зеркальное отражение изображения по вертикали и горизонтали
- поворот изображения на 90 градусов
- зумирование изображения с низким коэффициент

Также в задаче распознавания кустов на полях предварительно можно расширить исходную выборку добавлением в нее изображений с пониженным и повышенным контрастом. Для добавление в выборку изображений был разработан специальный скрипт с графическим интерфейсом, позволяющий проконтролировать корректность аугментации.



Рис. 5.6. Проверка корректности аугментации исходной выборки при помощи контраста

На рисунке 5.7 приведено распределение в полученной выборке.

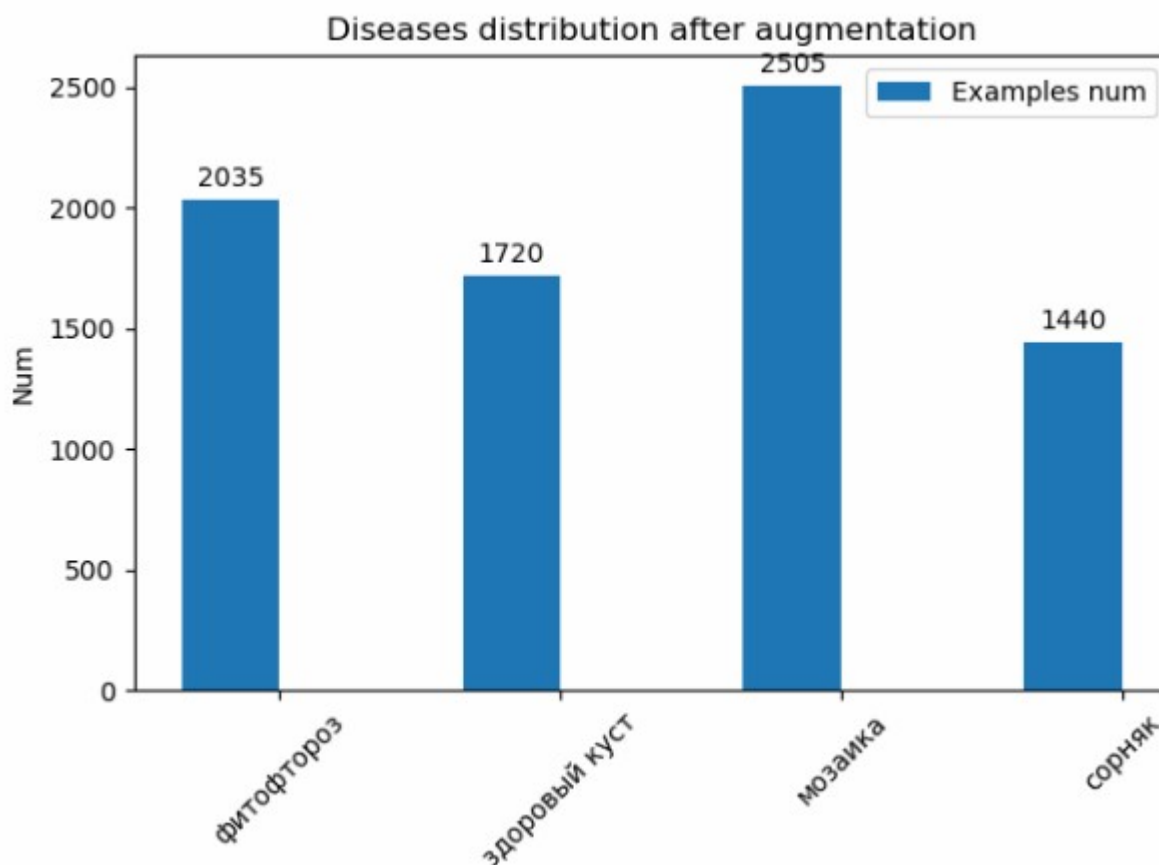


Рис. 5.7. Распределение полученной после аугментации выборки

Помимо изменения контраста утилита также поддерживает аугментацию при помощи размытия и аффинных преобразований. Также в утилите предусмотрена возможность добавления новых алгоритмов изменения изображений. В дальнейшем, полученная выборка будет использоваться при обучении моделей в пункте 8.

6. ИТОГОВЫЙ СТЕК МЕТОДОВ

Как было замечено ранее в пункте 4.2, поскольку в качестве метода классификации были выбраны сверточные нейронные сети, необходимость использования сегментатора отпадает, поскольку сверточные нейронные сети сами обучаются усиливать и обнаруживать интересующие видимые признаки на фотографиях.

Таблица 6.1.

Реализуемые стеки алгоритмов

Кластеризация	Сегментация	Классификация	Пост-анализ
Sliding Window	-	CNN	Minimum Distance

7. ВЫБОР БИБЛИОТЕКИ ДЛЯ ГЛУБОКОГО ОБУЧЕНИЯ

[8] Keras, TensorFlow и PyTorch входят в тройку лучших библиотек для глубокого обучения, которые предпочитают специалисты по обработке данных, а также новички в области глубокого обучения. Сравним эти библиотеки по следующим параметрам:

Таблица 7.1.

Сравнение библиотек глубокого обучения

	Документация	Производительность	Архитектура	Популярность	Предобученные модели	Сумма
TensorFlow	5	5	4	4	5	23
Keras	5	4	5	5	5	24
PyTorch	4	5	5	4	5	23

По сумме баллов победил Keras, к тому же наличие хорошей документации сыграло решающую роль при выборе библиотеки.

8. МОДУЛЬ КЛАССИФИКАЦИИ

8.1. Модуль поиска кандидатов

В итоговой системе, на вход нейронной сети могут поступать изображения, на которых нет растений, например, когда на обрабатываемом поле грядки расположены далеко друг от друга. Для того, чтобы снизить ненужную нагрузку на нейронную сеть, в модуль классификации была добавлен модуль поиска растений в массиве фотографий.

Модуль поиска растений реализован следующим образом:

- На вход модулю поступает пороговое значение зеленого цвета, а также массив изображений в формате RGB
- Для каждого изображения высчитывается содержание зеленого цвета (от 0 до 1)
- Если содержание зеленого цвета превышает пороговое значение — изображение считается кандидатом для классификации
- Индексы всех изображений-кандидатов сохраняются в список, который возвращает функция

8.2. Тестирование модуля поиска кандидатов

Для тестирования модуля была размечена выборка изображений земли и растений. Ниже приведены примеры классов в выборке. Выборка содержит около двухсот примеров кустов и земли. Точность работы детектора растений на выборке составляет 100 процентов. Проверка точность модуля была оформлена в виде регрессионного теста.

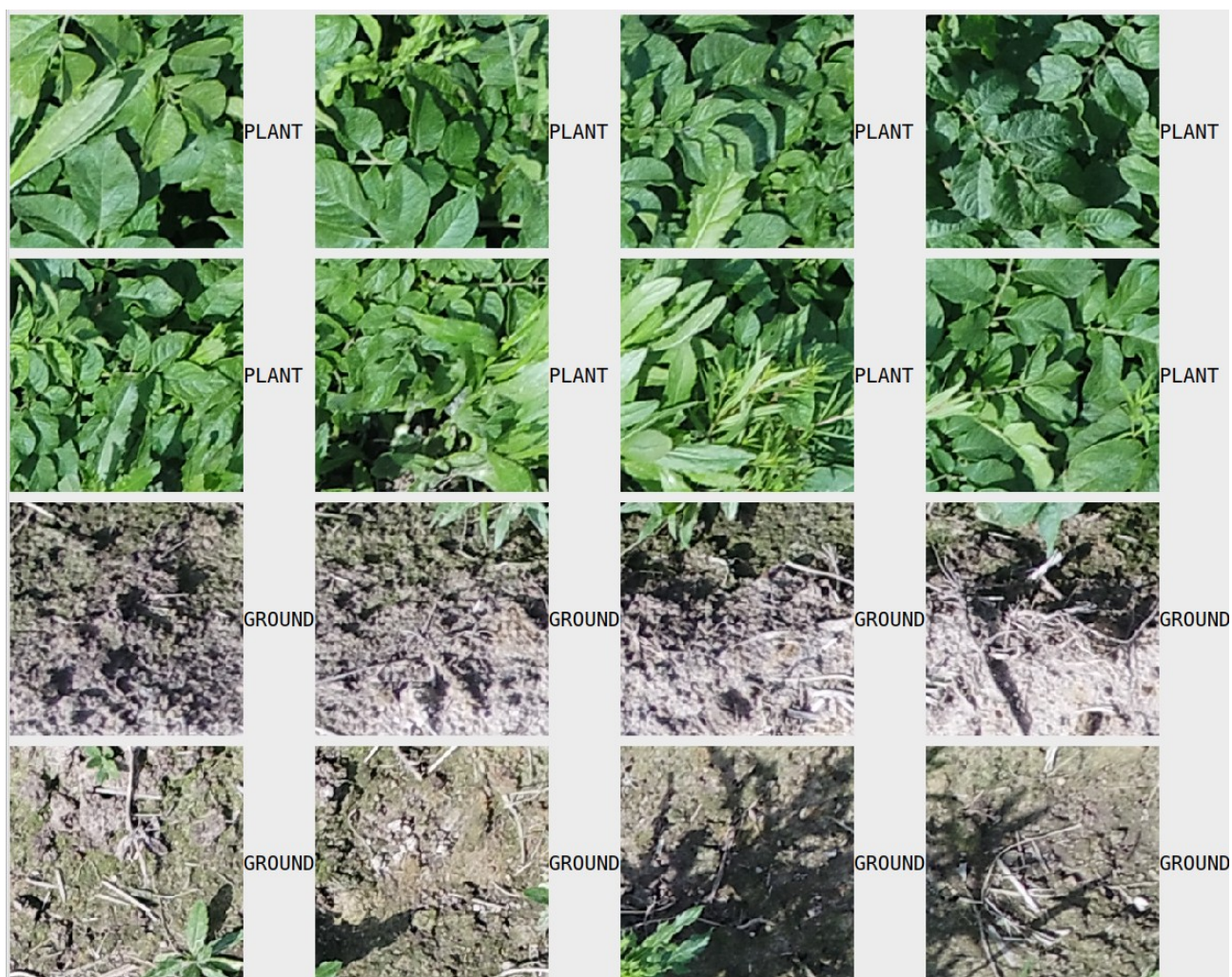


Рис 8.1 Примеры фотографий в выборке для тестирования модуля поиска кандидатов

8.3. Обучение классификатора

8.3.1. Выбор предобученных моделей

Рассмотрим предобученные модели, доступные в библиотеке Keras.

Таблица 8.2.

Предобученные модели доступные в Keras

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.713	0.901	138,357,544	23
VGG19	549 MB	0.713	0.900	143,667,240	26
ResNet50	98 MB	0.749	0.921	25,636,712	-
ResNet101	171 MB	0.764	0.928	44,707,176	-
ResNet152	232 MB	0.766	0.931	60,419,944	-
ResNet50V2	98 MB	0.760	0.930	25,613,800	-
ResNet101V2	171 MB	0.772	0.938	44,675,560	-
ResNet152V2	232 MB	0.780	0.942	60,380,648	-
InceptionV3	92 MB	0.779	0.937	23,851,784	159
InceptionResNetV2	215 MB	0.803	0.953	55,873,736	572
MobileNet	16 MB	0.704	0.895	4,253,864	88
MobileNetV2	14 MB	0.713	0.901	3,538,984	88
DenseNet121	33 MB	0.750	0.923	8,062,504	121
DenseNet169	57 MB	0.762	0.932	14,307,880	169
DenseNet201	80 MB	0.773	0.936	20,242,984	201
NASNetMobile	23 MB	0.744	0.919	5,326,716	-
NASNetLarge	343 MB	0.825	0.960	88,949,818	-

Сравним 1 среднюю по весу модель с хорошими результатами:

- Xception

И 2 более легковесные модели с менее хорошими результатами:

- MobileNetV2
- DenseNet121

Изначально обучение проводилось на исходной выборке, без аугментации при помощи изменения контраста.

8.3.2. Утилита для обучения сверточных нейронных сетей

С использованием библиотеки keras была разработана утилита, позволяющая производить обучение сверточных нейронных сетей. Параметры обучения задаются при помощи конфигурационного json-файла, имеющего структуру показанную на рисунке 8.3.

```
{
  "data": {
    "create_flow_dir": false,
    "flow_dir": "Datasets/final_dataset/big_flow_dir",
    "train_json": "Datasets/ds_classifier/ds_7500/big_ds.json"
  },
  "fit": {
    "baseline": {
      "train": 0.90,
      "test": 0.95,
      "eval": 0.95
    }
  },
  "preprocessor": {
    "use": false,
    "name": "contrast",
    "args": {
    }
  },
  "model": {
    "create_new": false,
    "new": {
      "type": "DenseNet121"
    },
    "exist": {
      "weights": "models/DenseNet_99_99_100/model_DenseNet121_75.hd5",
      "structure": "models/DenseNet_99_99_100/model_DenseNet121_75.json",
      "history": "models/DenseNet_98_96_98/model_DenseNet121_50_history.json"
    }
  }
}
```

Рис. 8.3. Пример конфигурационного json-файла утилиты обучения

В последней версии по ключу `train_json` указывается относительный путь json-файлу, содержащему сопоставление меток классов к относительным путям изображений. Json-файл с данными о выборке предварительно формируется ранее рассмотренной утилитой предназначенной для аугментации выборки. На рисунке 8.4. приведен пример сопоставления относительных путей изображений к меткам классов.

	id	label
0	Datasets/final_dataset/big_dir_ds/big_ds/1959.JPG	фитофтороз
1	Datasets/final_dataset/big_dir_ds/big_ds/1960.JPG	фитофтороз
2	Datasets/final_dataset/big_dir_ds/big_ds/1965.JPG	фитофтороз
3	Datasets/final_dataset/big_dir_ds/big_ds/1966.JPG	фитофтороз
4	Datasets/final_dataset/big_dir_ds/big_ds/1997.JPG	фитофтороз
5	Datasets/final_dataset/big_dir_ds/big_ds/1999.JPG	мозаика
6	Datasets/final_dataset/big_dir_ds/big_ds/2000.JPG	мозаика
7	Datasets/final_dataset/big_dir_ds/big_ds/2001.JPG	мозаика
8	Datasets/final_dataset/big_dir_ds/big_ds/2002.JPG	мозаика
9	Datasets/final_dataset/big_dir_ds/big_ds/2003.JPG	мозаика
10	Datasets/final_dataset/big_dir_ds/big_ds/2004.JPG	мозаика
11	Datasets/final_dataset/big_dir_ds/big_ds/2005.JPG	мозаика
12	Datasets/final_dataset/big_dir_ds/big_ds/2006.JPG	мозаика
13	Datasets/final_dataset/big_dir_ds/big_ds/2007.JPG	мозаика
14	Datasets/final_dataset/big_dir_ds/big_ds/2008.JPG	мозаика
15	Datasets/final_dataset/big_dir_ds/big_ds/2009.JPG	мозаика
16	Datasets/final_dataset/big_dir_ds/big_ds/2010.JPG	мозаика
17	Datasets/final_dataset/big_dir_ds/big_ds/2011.JPG	мозаика
18	Datasets/final_dataset/big_dir_ds/big_ds/2012.JPG	мозаика

Рис. 8.4. Пример сопоставления относительных путей изображений к меткам классов

Также на вход утилите подаются требования к точности ответов модели на каждой выборке, информации о препроцессоре изображений и о самой загружаемой модели. Утилита поддерживает режим создания и обучения новой модели, либо загрузки и дообучения предобученной.

Исходная выборка была разделена на 3 подвыборки: обучающую, тестовую и экзаменационную. Количество примеров выделенных в каждую выборку составило 70, 20 и 10 процентов исходной выборки соответственно.

Обучение моделей происходило при помощи генераторов данных. Генератор данных — итератор, возвращающий подвыборку данных, с случайно наложенными на нее заданными преобразованиями. Генераторы по своей сути также выполняют функцию аугментации исходных данных и помогают избежать переобучения модели (ситуации, когда модель запоминает исходную выборку вместо того, чтобы учиться классифицировать входные данные по признакам). В генераторах использовались следующие преобразования:

- Зумирование
- Вертикальное отражение
- Горизонтальное отражение

Создание генератора на основании обучающей выборки приведено на рисунке 8.5. Аналогичным образом создаются генераторы для тестовой и экзаменационной выборок.

```
train_generator = ImageDataGenerator(  
    shear_range=0.1,  
    zoom_range=0.1,  
    horizontal_flip=True,  
    vertical_flip=True,  
    preprocessing_function=preprocess_function  
) \  
    .flow_from_directory(  
        directory=train['flow_dir'],  
        target_size=(256, 256),  
        color_mode='rgb',  
        batch_size=train['batch_size']  
)
```

Рис. 8.5. Создание генератора на основании обучающей выборки

Для обучения модели использовалась функция `fit_generator` библиотеки `keras`. Само обучение было оформлено в виде цикла, управление которым происходит из консоли. В цикле происходит следующая последовательность действий:

- в консоли вводится число эпох, в течение которого модель будет обучаться
- запускается функция обучения модели
- при помощи библиотеки `matplotlib` отображается история обучения - график зависимости точности и функции потерь от эпох в течение обучения
- история обучения сохраняется
- модель тестируется на экзаменационной выборке при помощи функции `evaluate_generator` библиотеки `keras`
- в консоли предлагается запустить функцию визуализации ответов модели на небольшой подвыборке экзаменационной выборки, визуализация происходит при помощи библиотеки `PyQt5`
- предлагается сохранить модель
- предлагается продолжить обучение

Далее будут продемонстрированы результаты обучения выбранных в пункте 8.2.1 предобученных моделей. Каждая модель обучалась по 50 эпох.

8.3.3. Обучение Xception

На тренировочной и тестовой выборке точность предсказания составила 96 и 95 процентов соответственно. Графики истории изменения функции точности и функции потерь приведены на рисунках 8.6. и 8.7.

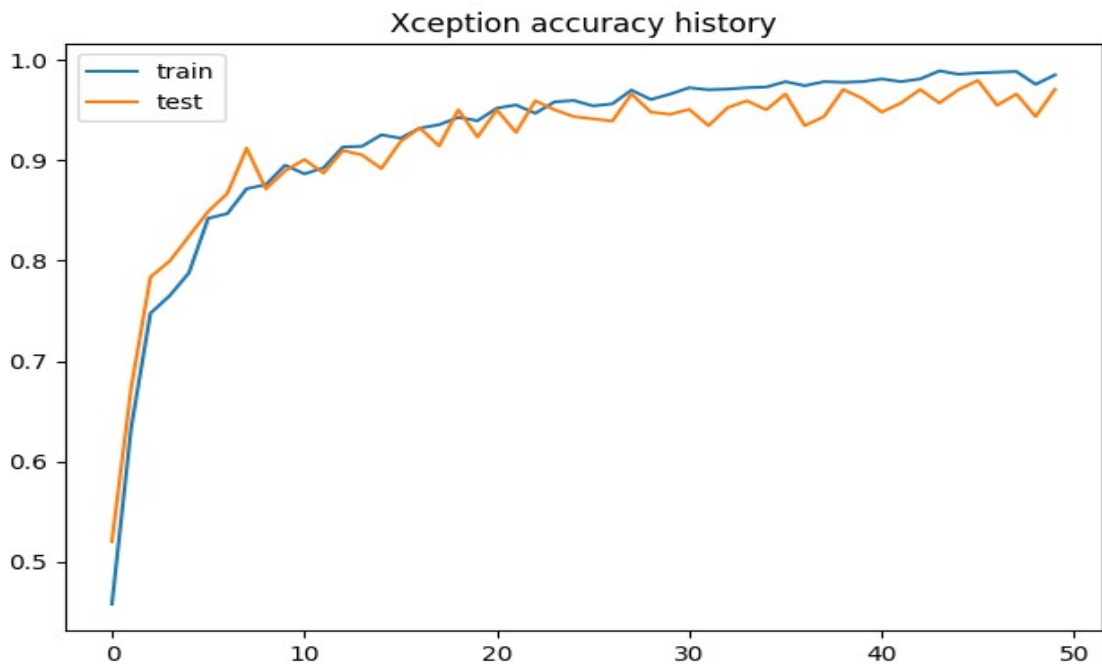


Рис. 8.6. Accuracy history сети Xception

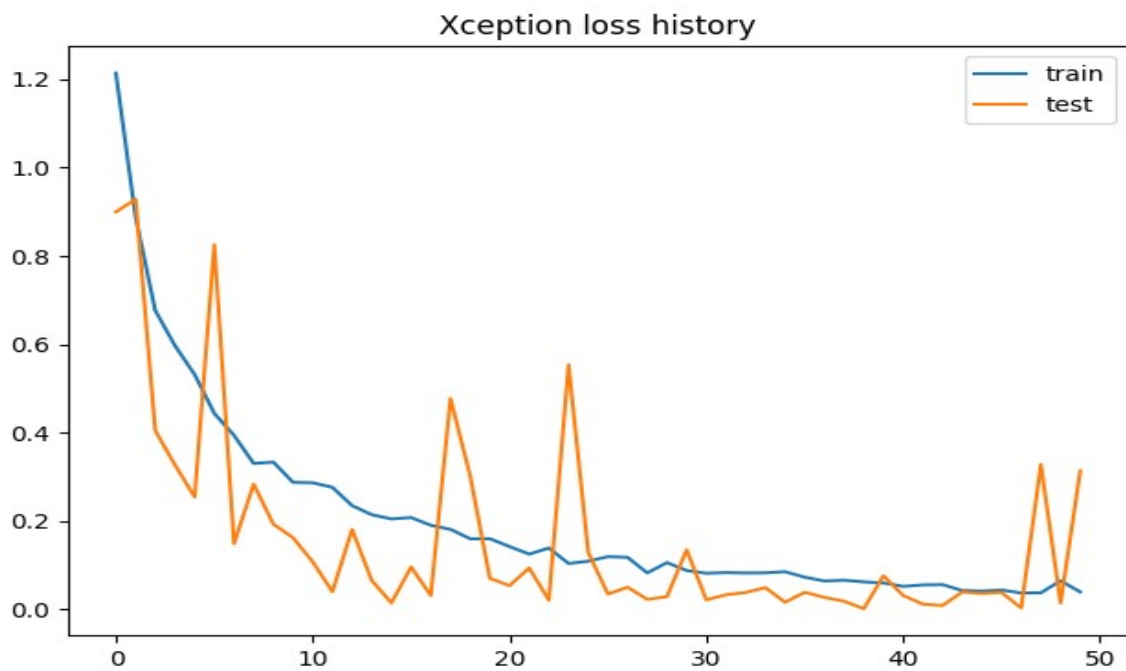


Рис. 8.7. Loss history сети Xception

На экзаменационной выборке точность предсказания сети Xception составила 96-98 процентов. Визуализация результатов предсказания на подвыборке из 16 изображений экзаменационной выборки представлена на рисунке 8.8.

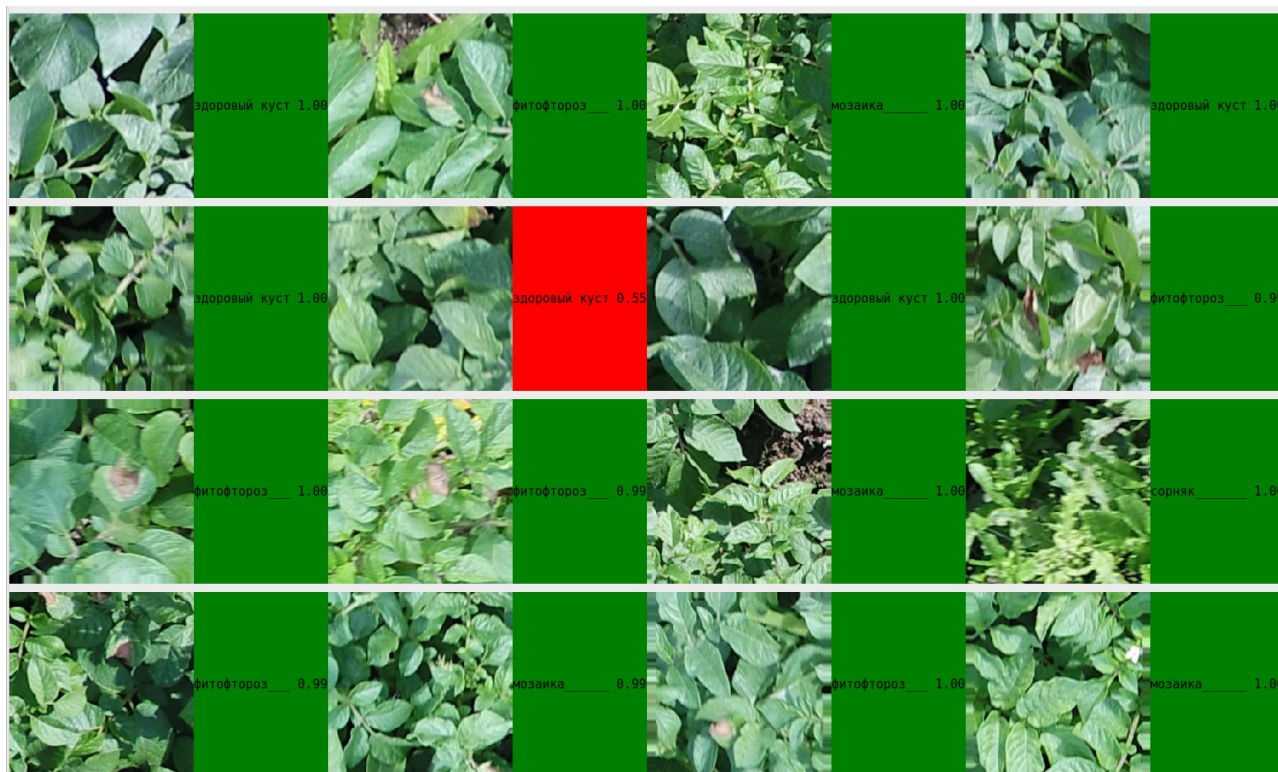


Рис. 8.8. Проверка Xception на оценочной выборке

8.3.4. Обучение MobileNetV2

На тренировочной и тестовой выборке точность предсказания составила 94 и 92 процентов соответственно. Графики истории изменения функции точности и функции потерь приведены на рисунках 8.9 и 8.10.

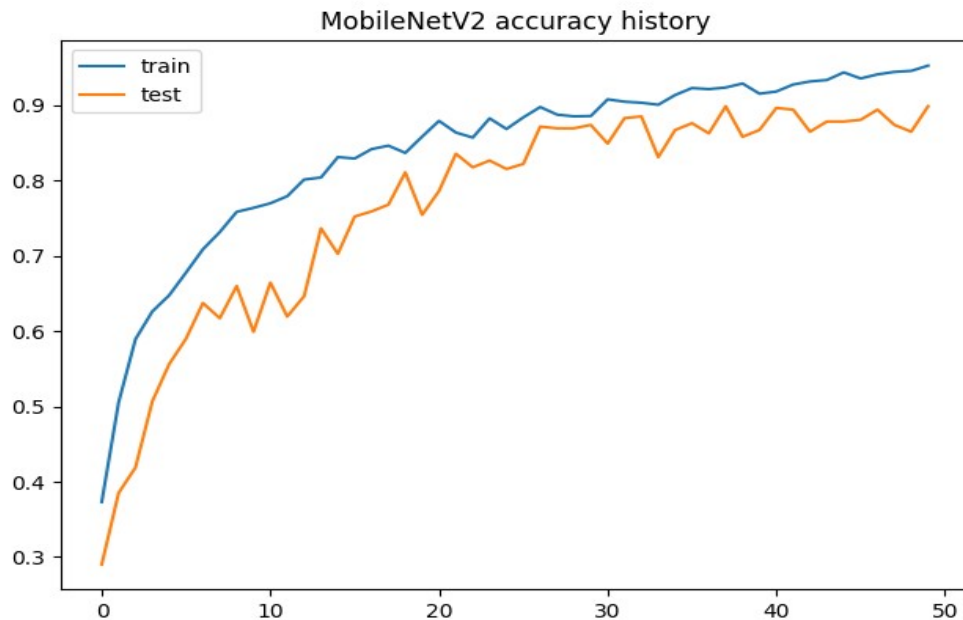


Рис. 8.9. Accuracy history сети MobileNetV2

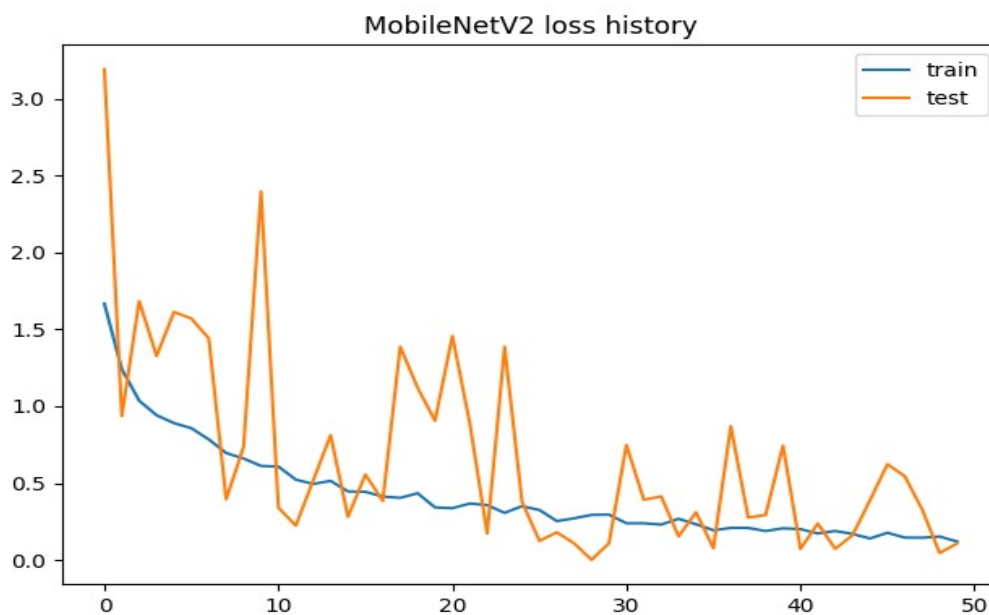


Рис. 8.10. Loss history сети MobileNetV2

На экзаменационной выборке точность предсказания сети MobileNetV2 составила 89-92 процента. Визуализация результатов предсказания на подвыборке из 16 изображений экзаменационной выборки представлена на рисунке 9.11.



Рис. 8.11. Проверка MobileNetV2 на оценочной выборке

8.3.5. Обучение DenseNet121

На тренировочной и тестовой выборке точность предсказания составила 96 и 95 процентов соответственно. Графики истории изменения функции точности и функции потерь приведены на рисунках 8.12 и 8.13.

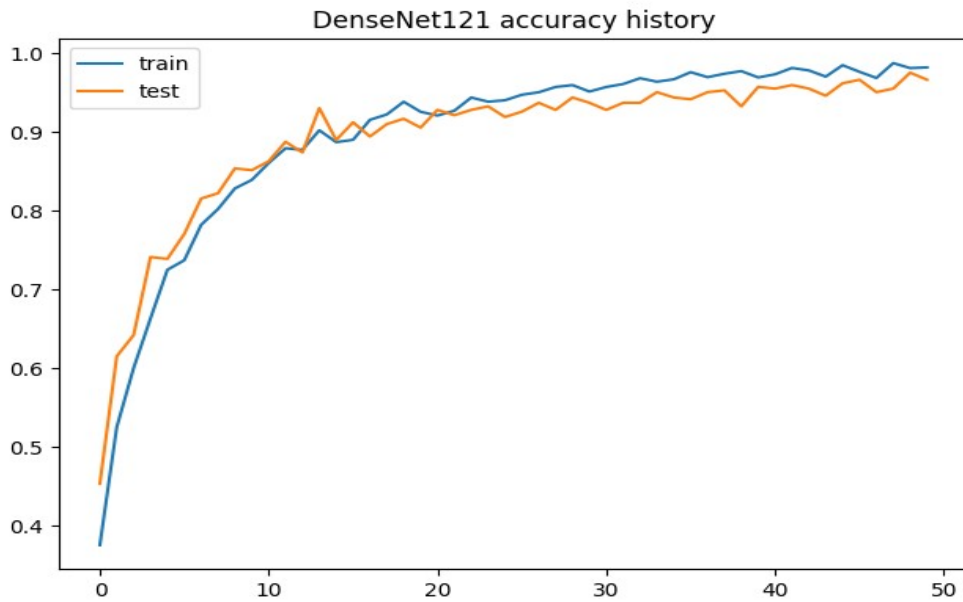


Рис. 8.12. Accuracy history сети DenseNet121

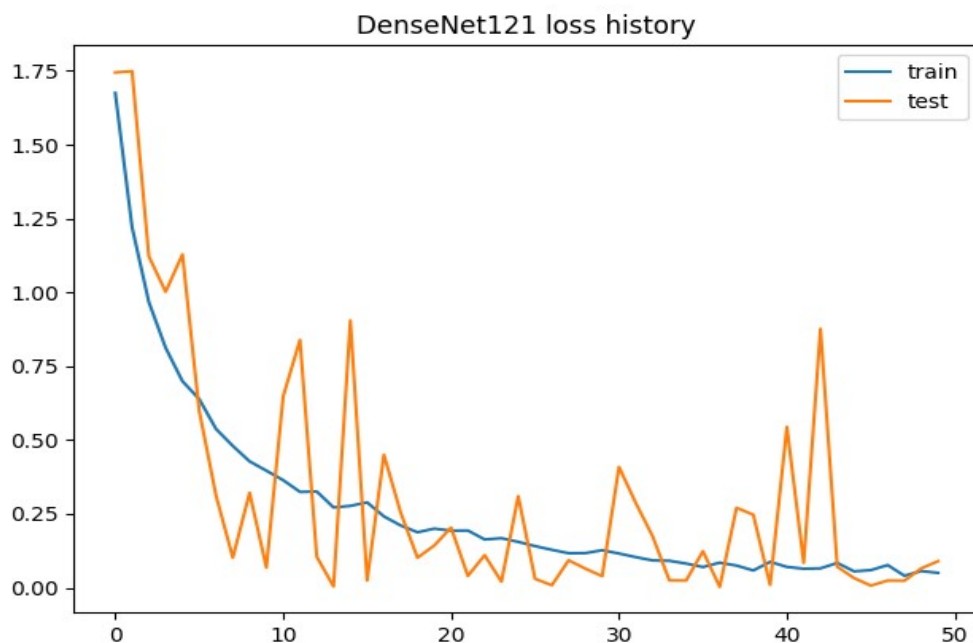


Рис. 8.13. Accuracy history сети DenseNet121

На экзаменационной выборке точность предсказания сети DenseNet121 составила 96-98 процентов. Визуализация результатов предсказания на подвыборке из 16 изображений экзаменационной выборки представлена на рисунке 8.14.



Рис. 8.14. Проверка DenseNet121 на оценочной выборке

8.3.6. Дообучене DenseNet121 на выборке с аугментацией

Поскольку архитектура DenseNet121 показала наилучшие результаты по соотношению точность/количество параметров, дообучим ее на исходной выборке, расширенной при помощи уменьшения и увеличения контраста.

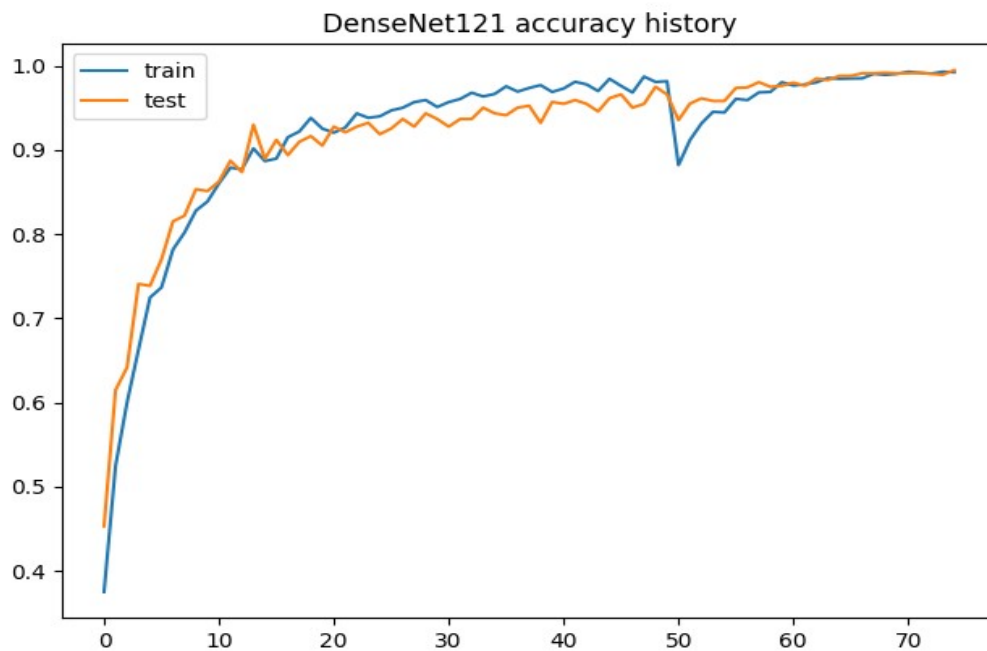


Рис. 8.15. Accuracy history сети DenseNet121 на расширенной выборке

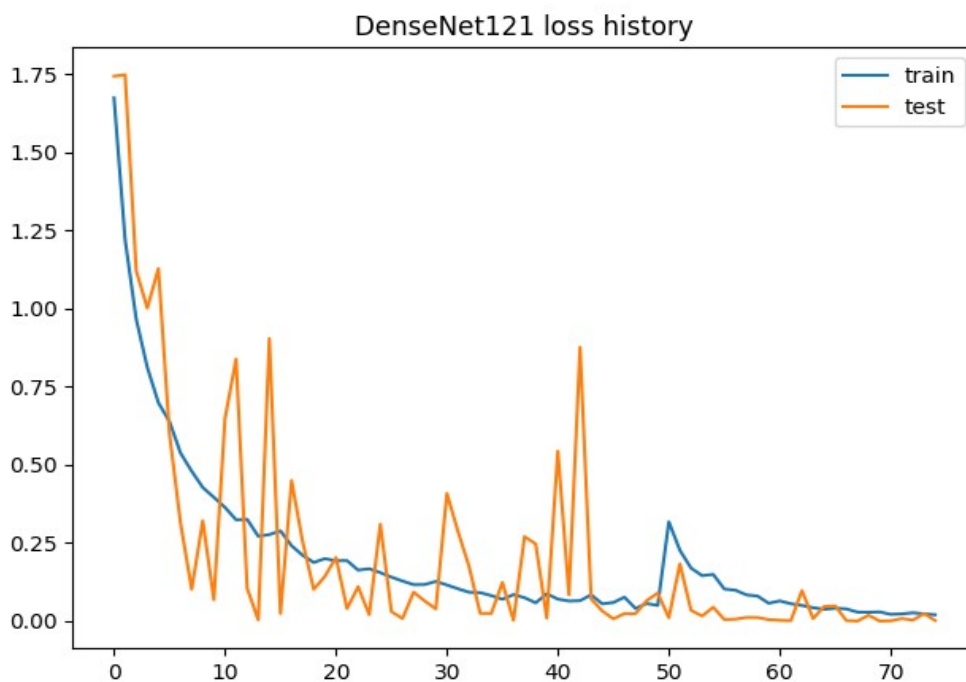


Рис. 8.16. Loss history сети DenseNet121 на расширенной выборке

На экзаменационной расширенной выборке точность предсказания сети DenseNet121 составила 98-99 процентов. Визуализация результатов предсказания на подвыборке из 16 изображений экзаменационной выборки представлена на рисунке 8.14.



Рис. 8.17. Проверка DenseNet121 на расширенной оценочной выборке

9. ТЕСТИРОВАНИЕ МОДУЛЯ

Тестирование некоторых компонент модуля уже было описано в предыдущих пунктах. Так например в пункте 8.2 было описано тестирование модуля поиска растений, а в пунктах 8.2.3 — 8.2.6 тестирование обученных моделей производилось на экзаменационной выборке вместе с последующей визуализацией ответов на небольших подвыборках экзаменационной выборки.

Также был разработан тест, проверяющий, что точность используемого в системе классификатора на расширенной выборке опускается не ниже заданного порога (95 процентов). Тест был запущен для всех моделей, прошла его только дообученная модель DenseNet121, для остальных моделей тест не прошел, поскольку их точность предсказания составляла около 80-ти процентов.

Помимо этого, была разработана графическая утилита, позволяющая производить классификацию растений по фотографии. Программа была протестирована на фотографиях сделанных с высоты 10, 20 и 30 метров. Проанализировав результаты классификации изображений, сделанных с разной высоты, можно сделать следующее предположение:

- при работе с фотографиями сделанными с большей высоты (10, 20 и 30 метров) обученная нейронная сеть не подходит, поскольку видимые признаки при изменении высоты съемки искажаются. Так скручивание листьев мозаики становится едва заметным, а форма всех листов становится похожа на форму листов сорняка.

На рисунках 9.1. — 9.4. продемонстрирована работа модуля классификации на фотографиях сделанных с разной высоты.

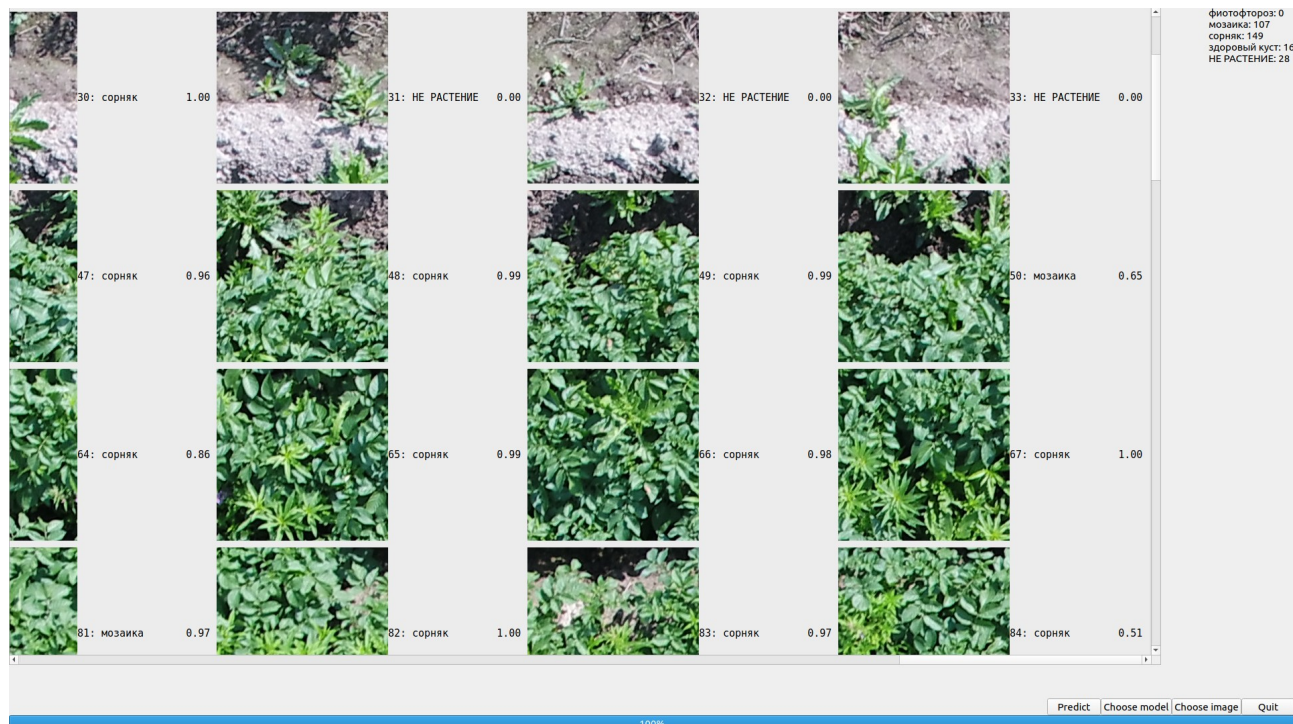


Рис. 9.1. Сорняки, мозаика и земля на фотографии сделанной с высоты 10 метров

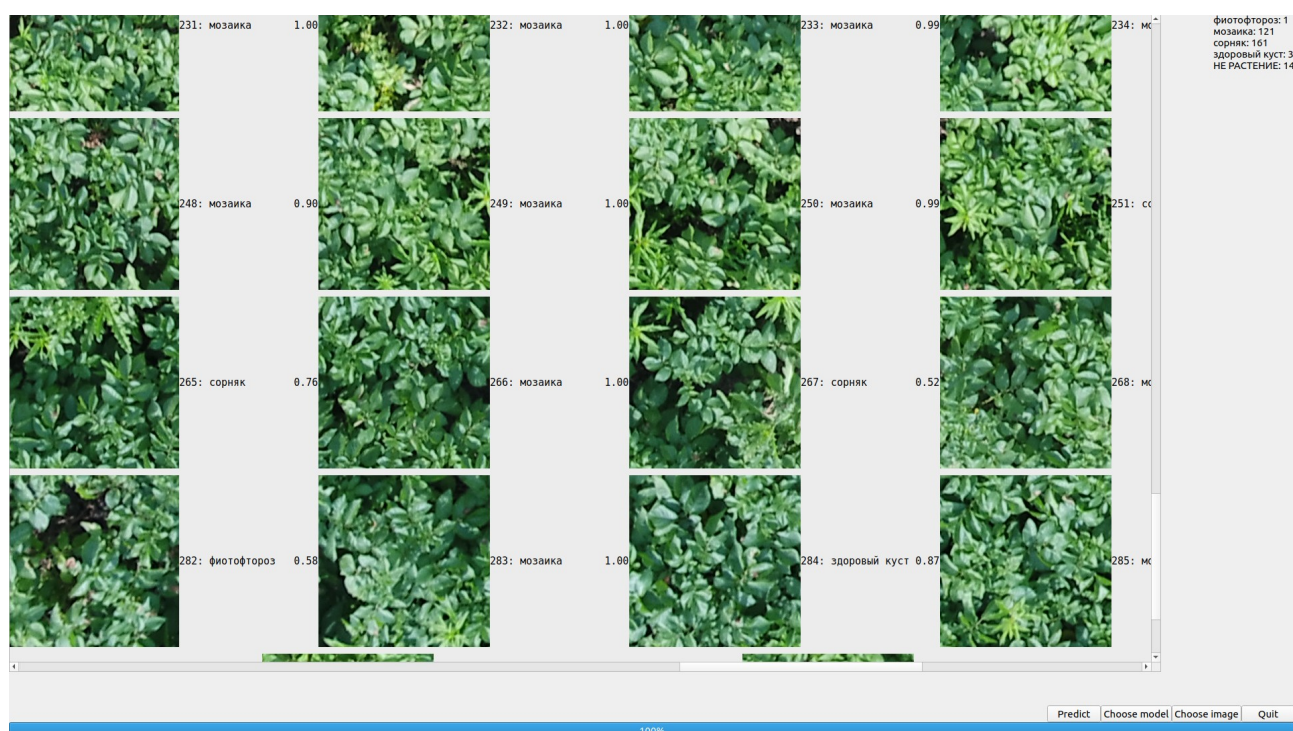


Рис. 9.2. Сорняки, мозаика и фитофтороз на фотографии сделанной с высоты 10 метров



Рис. 9.3. Сорняки и земля на фотографии, сделанной с высоты 20 метров



Рис. 9.4. Сорняки и земля на фотографии, сделанной с высоты 30 метров

ЗАКЛЮЧЕНИЕ

При выполнении работы:

- обоснована актуальность разработки системы выявления фитосанитарных угроз с использованием методов машинного обучения с учителем
- обоснована актуальность использования методов искусственного интеллекта для задачи классификации изображений
- проанализированы методы кластеризации, сегментации и классификации изображений
- разработаны графические утилиты для:
 - разметки фотографий и формирования обучающих выборок
 - аугментации сформированных выборок
 - обучения и тестирования сверточных нейронных сетей
 - тестирования модуля на фотографиях

При помощи разработанного программного обеспечения от специалистов по защите растений была получена выборка состоящая из 407 примеров фитофтороза, 344 примеров здорового куста, 501 примера мозаики картофеля и 288 примеров сорняков. Полученная выборка была расширена и увеличена в 4 раза при помощи добавления изображений с пониженным и повышенным контрастом с целью обучения классификатора распознавать фотографии в разное время суток и при разном освещении, поскольку входными данными классификатора являются фотографии сделанные в уличных условиях.

На исходных данных были дообучены 3 предобученных модели: Xception, MobileNetV2 и DenseNet121. DenseNet121 показала наилучший результат и

была дообучена на данных с аугментацией. Точность классификатора на экзаменационных данных составила 98-99 процентов.

Также для того, чтобы снизить ненужную нагрузку на нейронную сеть, в модуль классификации был добавлен модуль поиска растений в массиве фотографий. Для тестирования модуля была дополнительно сформирована выборка состоящая из 120 фотографий растений и 80 фотографий земли.

Для модуля классификации и модуля поиска растений были написаны регрессионные тесты, проверяющие точность работы модулей на специально сформированных при помощи разработанного ПО выборках. К модулю поиска растений и модулю классификации растений были предъявлены требования по точности определения в 100 и 95 процентов соответственно. Тесты обоими модулями были успешно пройдены.

По итогам работы можно сделать вывод о том, что классификация болезней растений по фотографиям сделанным с небольшой высоты решается методом сверточных нейронных сетей довольно точно. Таким образом, существует возможность своевременно обнаруживать на полях больные растения используя беспилотный летательный аппарат для съемки полей.

В дальнейшем планируется расширять исходную выборку и дообучать классификатор на новых данных, поскольку исходная выборка в основном лишь подтвердила гипотезу о возможности разработки заявленной системы, а для получения универсальной системы детектирования необходимо собрать больше данных и провести больше полевых экспериментов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Alexey Lukashin, Alexander Ilyashenko, Lev Utkin, Vladimir Muliukha . AIzimov: the Platform for Intellectual Diagnostics of Lung Cancer [электронный ресурс]:<https://fruct.org/publications/acm25/files/Luk.pdf>
2. Anelia Angelova, Alex Krizhevsky, Vincent Vanhoucke, Abhijit Ogale, Dave Ferguson. Real-Time Pedestrian Detection With Deep Network Cascades [электронный ресурс]:<https://research.google/pubs/pub43850/>
3. Arthur Ouaknine. Review of Deep Learning Algorithms for Image Semantic Segmentation [электронный ресурс]:https://medium.com/@arthur_ouaknine/review-of-deep-learning-algorithms-for-image-semantic-segmentation-509a600f7b57
4. Dimitrios Marmanis . Mihai Datcu [электронный ресурс]:<https://ieeexplore.ieee.org/abstract/document/7342907>
5. JiaboSun, JianyuYang, ChaoZhang, WenjuYun, JieqingQu. Automatic remotely sensed image classification in a grid environment based on the maximum likelihood method [электронный ресурс]:<https://www.sciencedirect.com/science/article/pii/S0895717711006686>
6. Image Classification [электронный ресурс]:<https://cs231n.github.io/classification/>
7. Понимание сверточных нейронных сетей с помощью визуализаций в PyTorch [электронный ресурс]:<https://ru.sciencewal.com/65986-understanding-convolutional-neural-networks-through-visualizations-in-pytorch-b5444de08b91-97#menu-3>
8. ИТ в агропромышленном комплексе России [электронный ресурс]:http://www.tadviser.ru/index.php/Статья:ИТ_в_агропромышленном_комп_лексе_России

10. Вирусные болезни картофеля и меры по ограничению их распространения [электронный ресурс]:<http://www.agrocounsel.ru/virusnye-bolezni-kartofelya-i-mery-po-ogranicheniyu-ih-rasprostraneniya>
12. SUPERVISED IMAGE CLASSIFICATION USING MINIMUM DISTANCE ALGORITHM [электронный ресурс]:<http://www.50northspatial.org/supervised-image-classification-using-minimum-distance-algorithm/>
13. Простейшая кластеризация изображений методом k-средних (k-means) [электронный ресурс]:<https://habr.com/ru/post/165087/>
14. Keras vs TensorFlow vs PyTorch : Comparison of the Deep Learning Frameworks [электронный ресурс]:<https://www.edureka.co/blog/keras-vs-tensorflow-vs-pytorch/>
15. Randall B. Smith, Ph.D. Image Classification [электронный ресурс]:<https://www.microimages.com/documentation/Tutorials/classify.pdf>
16. Selva Prabhakaran. Mahalanobis Distance – Understanding the math with examples [электронный ресурс]:<https://www.machinelearningplus.com/statistics/mahalanobis-distance/>
17. Sharada P. Mohanty, David P. Hughes, Marcel Salathé. Using Deep Learning for Image-Based Plant Disease Detection [электронный ресурс]:<https://www.frontiersin.org/articles/9.3389/fpls.2016.01419/full>
18. Sunitha Abburu, Suresh Babu Golla . Satellite Image Classification Methods and Techniques: A Review [электронный ресурс]:<http://citeseerx.ist.psu.edu/viewdoc/download?doi=9.1.1.695.1415&rep=rep1&type=pdf>
19. Н.Ю. Мартюшева. Разработка информационной системы для автоматизированной диагностики заболеваний растений: выпускная квалификационная работа магистра: 09.04.01 - Информатика и вычислительная

техника ; 09.04.01_15 - Технологии проектирования системного и прикладного программного обеспечения [электронный ресурс]:<https://elib.spbstu.ru/dl/3/2019/vr/vr19-719.pdf/info>

ПРИЛОЖЕНИЕ ССЫЛКА НА ПРОЕКТ НА GITHUB

<https://github.com/DanilaEremenko/PlantDisease>