

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа программной инженерии

Работа допущена к защите
Директор ВШ ПИ
_____ П.Д. Дробинцев
«__» _____ 2019 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

Тема: Компьютерная диагностика заболеваний растений по
текстурным характеристикам изображений листьев

направление 09.03.04 «Программная
инженерия» образовательная программа
09.03.04_01 «Технология разработки и сопровождения
качественного программного продукта»

Выполнил студент гр. 43504/2 (подпись) Д.О. Лелюхин

Руководитель к.т.н., доцент (подпись) В.С. Тутыгин

Санкт-Петербург
2019 г.

Санкт-Петербургский политехнический
университет Петра Великого Институт
компьютерных наук и технологий Высшая школа
программной инженерии

У Т В Е Р Ж Д А Ю
Директор ВШПИ
П.Д. Дробинцев
« ____ » _____ г.

З А Д А Н И Е **на соискание степени бакалавра**

Студенту группы 43504/2 Лелюхину Дмитрию Олеговичу Тема
проекта (работы) Компьютерная диагностика заболеваний
растений по текстурным характеристикам изображений листьев
(утверждена распоряжением по ИКНТ от _____ № _____)

2. Срок сдачи студентом оконченного проекта (работы) 1 июня 2019г.

3. Исходные данные к проекту (работе)

Имеется описание метода описания текстурных изображений на основе матрицы смежности (матрицы GLCM). Требуется разработать алгоритм и программные средства для нормализации изображений листьев и диагностики болезней растений по ключевым признакам Contrast, Correlation, Energy, Homogeneity, вычисляемым на основе матриц GLCM для основных цветовых компонентов нормализованных изображений листьев, и исследовать эффективность алгоритма распознавания нескольких болезней.

4. Содержание расчетно-пояснительной записки
(перечень подлежащих разработке вопросов)

4.1. Обзор методов и алгоритмов распознавания текстурных изображений.

4.2. Разработка алгоритма и программы нормализации фотоизображений листьев растений.

4.3. Разработка алгоритма и программы диагностики болезней растений по фотоизображениям листьев.

4.4. Программная реализация разработанного алгоритма на языке C++ и экспериментальные результаты.

Реферат

С. 71, Рис. 14, Прилож. 8.

НОРМАЛИЗАЦИЯ ИЗОБРАЖЕНИЙ, МАТРИЦА
GLCM, МАКСИМУМ ФУНКЦИИ
ПРИНАДЛЕЖНОСТИ, НЕЧЕТКАЯ ЛОГИКА,
БИНАРИЗАЦИЯ, МАЖОРИТАРНОЕ
ГОЛОСОВАНИЕ

Объектом исследования является набор из 1000 изображений листьев пшеницы.

Цель работы – разработка и реализация программы для получения надежного распознавания 10-ти основных видов болезней пшеницы на изображениях.

Рассмотрены существующие методы и предложен метод цифровой обработки изображений для диагностики болезней растений по цифровым RGB-фотоизображениям листьев в видимом спектре при использовании матрицы GLCM, сравнения полученных эталонных описаний математических ожиданий по максимуму функции принадлежности, а также последующего распознавания тестовой выборки на основе бинаризации результатов и мажоритарного голосования.

Проведено математическое моделирование предложенного метода, разработан и реализован классификатор для определения вида болезни листьев пшеницы.

Новизна работы заключается в способе компьютерной диагностики видимых симптомов заболеваний листьев растений по фотоизображениям посредством оперирования матрицей GLCM и интерпретировании результатов, используя нечеткую логику. Предложенный подход при группировке по 25-ти изображениям позволяет получить достоверное распознавание в более чем 90% случаях.

В перспективе, разработанное ПО имеет все шансы на внедрение в автономные БПЛА для автоматического распознавания поврежденных патологиями площадей, засеянных различными культурными растениями. Успешная интеграция разработанного ПО в сельское хозяйство позволит снизить физическую нагрузку с фермеров, позволит вовремя диагностировать начинающиеся очаги болезней культурных растений, после устранения которых повысится урожайность.

ABSTRACT

P. 71, Im. 14, Attach. 8.

IMAGE NORMALIZATION, GLCM MATRIX, MEMBERSHIP FUNCTION MAXIMUM, FUZZY LOGIC, BINARIZATION, MAJORITY VOTING

The object of study is a dataset of 1000 images of wheat leaves.

The main goal of the work was the development and implementation of software for obtaining reliable recognition of the 10 main types of wheat diseases in the images.

Existing methods are explored and a digital image processing method is proposed for diagnosing plant diseases using digital RGB photographic images of leaves in visible spectrum using the GLCM matrix, comparing the results of mathematical expectations for the maximum functions inherent in subsequent voting recognition.

A mathematical modeling of the proposed element has been passed successful and a classifier has been developed and implemented to determine the type of wheat disease.

The newest work is that computer diagnostics allows to detect the symptoms of plants disease when

photographing leaves by means of GLCM matrices and interpreting the results using fuzzy logic. The proposed approach when grouped by 25 images allows to obtain reliable recognition in more than 90% of cases.

In the future, the developed software has all the chances of being introduced into autonomous UAVs for automatic recognition of areas damaged by pathologies, sown with various cultivated plants. Successful integration of the developed software into agriculture will reduce the physical burden of farmers, will allow time to diagnose emerging foci of diseases of cultivated plants, after the elimination of which the yield will increase.

Содержание

Обозначения и сокращения.....	9
Введение	10
1. Обзор методов и алгоритмов распознавания болезней растений по изображениям листьев.....	14
1.1. Нейросетевое распознавание.....	14
1.2. Распознавание классическими методами....	23
1.3. Сравнение основных методов диагностики болезней растений.....	25
2. Разработка метода распознавания болезней листьев растений и исследование достижимой точности определения вида болезни	27
2.1. Предобработка изображений.....	27
2.2. Матрица GLCM.....	36
2.3. Построение гистограмм и аппроксимация ..	40
2.4. Классификация болезней	43
3. Реализация классификатора.....	48
3.1. Инструменты для разработки.....	48
3.2. Системный уровень разработки.....	49
3.3. Архитектурный уровень разработки	50
3.4. Программная реализация	51
3.5. Тестирование.....	54
Результаты экспериментального исследования	55
Заключение	57
Библиографический список	59

Приложение 1. Методы классификации болезней растений по изображениям листьев.	61
Приложение 2. Нормализованные изображения для каждого типа болезни листьев растений.	62
Приложение 3. Листинг программы моделирования для расчета доверительных интервалов в среде MatLab R2016a.	63
Приложение 4. Листинг программы моделирования для диагностики 10 болезней по эталонным описаниям в среде MatLab R2016a.....	64
Приложение 5. Таблица эталонных описаний математических ожиданий для всех групп патологий.	66
Приложение 6. Результаты программы в текстовом виде для диагностики 10 болезней по эталонным описаниям в среде MatLab R2016a.....	68
Приложение 7. Пример описания интерфейса класса GlcmBuilder.....	69
Приложение 8. Результаты тестирования методов класса GlcmBuilder.....	71

Обозначения и сокращения

БПЛА - беспилотный летательный аппарат.

ПО – программное обеспечение.

СКО – среднеквадратичное отклонение.

ЦПТ – центральная предельная теорема.

ANN – Artificial Neural Network (рус. искусственная нейронная сеть).

CNN – Convolutional Neural Network (рус. сверточная нейронная сеть).

GLCM - Gray-Level Co-Occurrence Matrix (рус. матрица совместной встречаемости уровней яркости).

HOG – Histogram of Oriented Gradients (рус. гистограмма ориентированных градиентов).

KNN - K-Nearest Neighbor (рус. метод K-ближайших соседей, K-средних).

ML – Machine Learning (рус. машинное обучение).

PNN – Probabilistic Neural Network (рус. вероятностная нейронная сеть).

SIFT – Scale-Invariant Feature Transform (рус. масштабно-инвариантная трансформация признаков).

SVM – Support Vector Machine (рус. метод опорных векторов).

Введение

В настоящее время сельское хозяйство стало намного больше, чем просто средством для кормления постоянно растущих популяций. Растения стали ценным источником энергии, способным решить проблему глобального потепления.

Население Земли стремительно растёт и по прогнозам ООН к 2030 году достигнет восьми с половиной миллиардов человек. Территории, наиболее пригодные для сельского хозяйства, уже обрабатываются, и значительного их прироста не предвидится. Теперь перед человечеством стоит задача вырастить максимальное количество урожая, вовремя предотвращая распространение вредителей и возбудителей заболеваний.

Существует большое количество заболеваний, которые влияют на урожайность растений, что ведет к экономическим и экологическим потерям. Болезни растений приводят к потере 10–16% мирового урожая сельскохозяйственных культур каждый год, что обходится в 220 миллиардов долларов США [6]. В этом контексте диагностика заболеваний растений,

дающая точный и своевременный результат, имеет первостепенное значение. Следовательно, целесообразно было бы выбрать объектом исследования наиболее значимую для человечества зерновую культуру. Такой культурой является пшеница, известная нам с глубокой древности, и, служащая основным поставщиком растительного белка для большей части человечества. Ее мировая площадь уже превысила 235 млн га. Однако, пшеница, как и все зерновые культуры подвержена многим видам заболеваний, в том числе грибками или гнилями, после чего у зараженных растений снижается зимостойкость и засухоустойчивость.



Рисунок 1. Пример аэрофотосъемки засеянных полей с помощью БПЛА.

Общим подходом выявления патологий растений является использование дистанционного метода обнаружения (см. Рис. 1), который исследует фотоизображения, полученные с помощью беспилотных летательных аппаратов (БПЛА).

Беспилотные летательные аппараты оснащаются мультиспектральными камерами, высокая четкость изображения которых позволяет точно выявлять проблемные участки поля (Рис. 2) и применить оборудование для распыления химикатов.

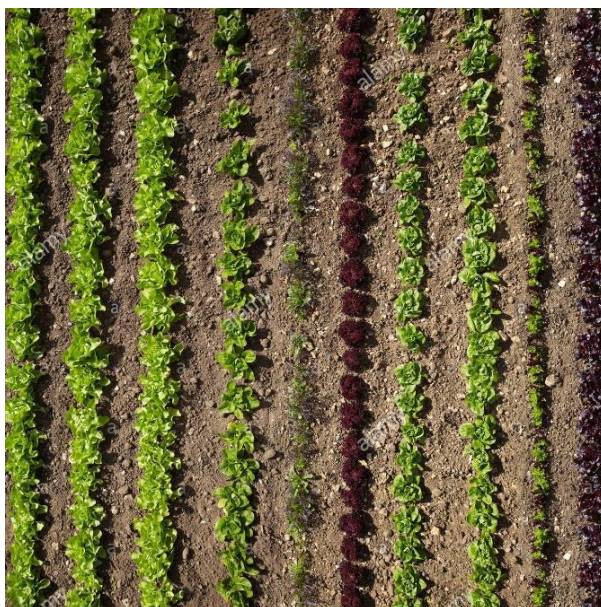


Рисунок 2. Пример полученных с помощью БПЛА изображений листьев растений.

Применение БПЛА в сельском хозяйстве имеет огромный потенциал и с каждым годом интерес к их использованию растет. За последние 5-6 лет по развитию этой отрасли машиностроения сделано во много раз больше, чем за все предыдущие годы. Это позволяет считать эту отрасль бурно развивающейся.

Большинство заболеваний листьев растений порождают какое-то проявление в видимом спектре [2]. В подавляющем большинстве случаев, диагноз или, по крайней мере, первое предположение о заболевании, осуществляется людьми визуально.

Важно учитывать, что сельскохозяйственные культуры занимают чрезвычайно большие площади, делая мониторинг сложной задачей. Проблема может быть решена за счёт использования цифровых фотоизображений, полученных с помощью БПЛА в сочетании с обработкой изображений, распознаванием образов и автоматическими инструментами классификации, один из которых и был разработан.

1. Обзор методов и алгоритмов распознавания болезней растений по изображениям листьев

В настоящей главе будут рассмотрены существующие решения для распознавания болезней листьев растений по изображениям, а также выявлены недостатки существующих методов.

1.1. Нейросетевое распознавание

Одним из подходов диагностики заболеваний растений по изображениям листьев является метод распознавания, использующий нейронные сети [6].

Распознавание нейросетевыми алгоритмами традиционно делится на несколько основных этапов:

- на первом этапе полученные изображения сегментируются с использованием методов K-средних;
- из полученных сегментированных изображений вычисляются вектора признаков, используя HOG или SIFT методы;

- вектора признаков передаются через предварительно обученную нейронную сеть CNN, ANN, PNN или SVM, которая выносит окончательное решение о принадлежности изображения растения к той или иной группе патологий.

Результаты, приводимые авторами изученных статей, показывают, что классификаторы нейронными сетями, основанные на статистической классификации, обеспечивают автоматическое обнаружение заболеваний листьев всевозможных растений с точностью в пределах 83 – 94% [8].

Проведем обзор главных методов, используемых для диагностики заболеваний листьев растений:

1) Метод k -средних - это простой классификатор в методах машинного обучения, где классификация достигается путем определения ближайших соседей к примерам запроса и последующего использования этих соседей для определения класса запроса. Принадлежность точки к классу основывается на расчете минимального расстояния между данной точкой и другими точками.

В качестве классификатора ближайший сосед не включает какой-либо учебный процесс. Он не применим в случае большого количества обучающих примеров, поскольку он не устойчив к шумным данным. Для классификации листьев растений рассчитывается евклидово расстояние между тестовыми и тренировочными образцами. Таким образом, он обнаруживает аналогичные меры и, соответственно, класс для тестовых образцов. Выборка классифицируется на основе наибольшего числа голосов от k -соседей, причем выборка присваивается классу, наиболее распространенному среди его k -ближайших соседей.

Метод ближайшего соседа прост в реализации, а также демонстрирует достаточно хорошие результаты, если функции выбраны тщательно.

Классификатор KNN хорошо справляется с основными проблемами распознавания.

Основной недостаток метода KNN состоит в том, что он является медленным алгоритмом при обучении, то есть он ничего не изучает из обучающих данных и просто использует сами обучающие данные для классификации. Другим недостатком является то, что

этот метод также довольно медленный, если имеется большое количество обучающих примеров, так как алгоритм должен вычислять расстояние и сортировать все обучающие данные при каждом прогнозе. Также недостаточно надежен при оперировании зашумленными данными в случае большого количества обучающих примеров. Наиболее серьезным недостатком KNN является то, что он очень чувствителен к наличию не относящихся к делу параметров.

2) Метод опорных векторов (SVM) представляет собой нелинейный классификатор. Это новая тенденция в ML алгоритмах, которая используется во многих задачах распознавания образов, включая классификацию текстурных показателей изображений. В SVM входные данные нелинейно отображаются на линейно разделенные данные в некотором многомерном пространстве, обеспечивая хорошую производительность классификации. SVM максимизирует предельное расстояние между различными классами. Разделение классов проводится с разными ядрами. SVM предназначен для работы только с двумя классами

путем определения гиперплоскости для разделения двух классов. Это делается путем максимизации разрыва от гиперплоскости до двух классов. Образцы, расположенные ближе всего к краю, которые были выбраны для определения гиперплоскости, известны как опорные векторы.

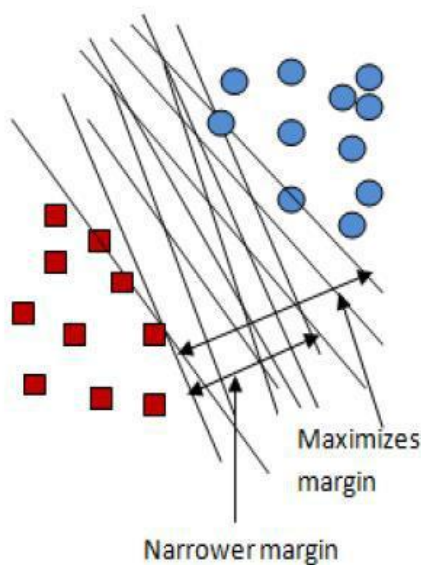


Рисунок 3. Визуализация метода опорных векторов.

На рис. 3 показана концепция машин опорных векторов. Мультиклассовая классификация также применима и в основном строится различными SVM двух классов для решения этой проблемы, используя

один против всех или один против одного. Класс выигрыша затем определяется по наибольшей выходной функции или максимальному количеству голосов соответственно.

Основными преимуществами SVM являются:

- высокая точность прогноза.
- надежная работа, при содержании ошибок примерами обучения.
- простая геометрическая интерпретация и разреженное решение.
- как в нейронных сетях, вычислительная сложность SVM не зависит от размерности входного пространства.

Недостатками SVM являются:

- длительное время обучения.
- трудно интерпретировать изученную функцию (вес).
- большое количество векторов поддержки, используемых в обучающем наборе для выполнения задач классификации.

3) Искусственный нейрон - это в основном инженерный подход биологического нейрона. ANN

состоит из ряда узлов, называемых нейронами. Нейронные сети обычно организованы в слоях. В нейронной сети каждый нейрон в скрытом слое получает сигналы от всех нейронов входного слоя. Сила каждого сигнала и смещения представлены весами и константами, которые рассчитываются на этапе обучения. После того, как входы взвешены и добавлены, результат преобразовывается передаточной функцией в выход. Эта сеть состоит только из входных нейронов и выходных нейронов. Такой подход помогает решать только линейные задачи. Многослойный персептрон является одной из нейронных сетей с прямой связью с одним или несколькими слоями между входным и выходным слоями. Прямая связь означает, что данные передаются в одном направлении от входного к выходному слою (вперед). Несколько слоев нейронов с нелинейными передаточными функциями позволяют сети изучать нелинейные и линейные отношения между входными и выходными векторами. Многослойные персептроны (Рис. 4), которые можно обучать с использованием алгоритма обратного распространения, являются очень популярным 20

выбором для многих исследователей в области распознавания и классификации объектов.

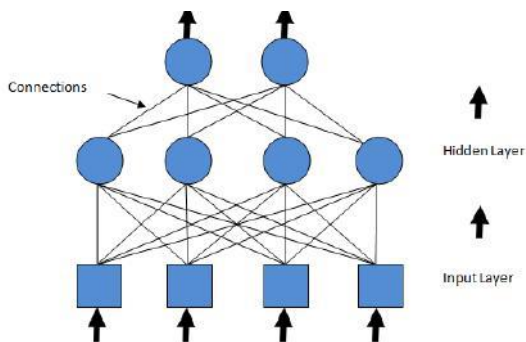


Рисунок 4. Многослойная искусственная нейронная сеть.

4) Операции в вероятностных нейронных сетях (PNN) организованы в многослойную сеть прямой связи с четырьмя уровнями. PNN в основном используется в задачах классификации. Первый слой - это входной слой, который рассчитывает расстояние от входного вектора до обучающих входных векторов. Второй уровень суммирует вклад для каждого класса входных данных и производит свой чистый результат в виде вектора вероятностей. Третий шаблонный слой содержит один нейрон для каждого случая в наборе обучающих данных. Он хранит значения переменных предиктора для случая

вместе с целевым значением. Нейроны шаблона добавляют значения для класса, который они представляют. Выходной слой сравнивает взвешенные голоса для каждой целевой категории, накопленной в слое шаблонов, и использует наибольшее количество голосов для прогнозирования целевой категории. Поскольку PNN намного быстрее, чем многослойные сети персептронов, для их фазы обучения требуется всего один проход через схемы обучения. PNN может быть точнее, чем многослойные сети персептронов, также относительно нечувствительны к выбросам. Для повышения общей производительности выходные данные PNN могут быть позже обработаны другой системой классификации, и, поскольку это происходит очень быстро, PNN используются в онлайн-приложениях, где требуется классификатор в реальном времени. Основным недостатком PNN является то, что он требует много пространства для хранения данных.

1.2. Распознавание классическими методами

К классическим методам распознавания болезней растений по изображениям листьев многие авторы статей относят подходы использующий алгоритмы нечеткой логики в совокупности с векторами признаков, получаемым при вычислении текстурных характеристик изображений [9]. Классификаторы нечеткой логики - это системы классификации, которые используют нечеткие множества или нечеткую логику и преобразуют реальные значения данных в степени членства посредством использования функций принадлежности, так что эти правила затем можно использовать для процесса классификации. Это делается путем определения «категорий» для каждого из атрибутов.

Поскольку классификаторы нечеткой логики имеют очень высокую скорость, они предпочтительны в случаях, когда точность значений данных ограничена или, когда требуется классификация в режиме реального времени. Обработка нечетких

изображений - это совокупность всех подходов, которые понимают, представляют и обрабатывают изображения, их сегменты и функции как нечеткие множества. Представление и обработка зависят от выбранного нечеткого метода и от решаемой проблемы. Обработка нечетких изображений делится на три основных этапа: фаззификация изображений, изменение значений принадлежности и, при необходимости, дефаззификация изображений. Из-за неопределенностей, которые существуют во многих аспектах обработки изображений, таких как аддитивный и неаддитивный шум при обработке изображений низкого уровня, неточности в предположениях, лежащих в основе алгоритмов, и неоднозначности в интерпретации во время обработки изображений высокого уровня, нечеткая обработка является желательной. Основным недостатком нечеткой логики как классификатора является размерность, поскольку этот классификатор не подходит для задач, имеющих большое количество функций.

Текстурные характеристики большинство авторов рекомендуют находить, используя матрицы

совместной встречаемости GLCM, которые наиболее точно отражают статистическую составляющую исследуемых изображений [10].

1.3. Сравнение основных методов диагностики болезней растений

После изучения вышеупомянутых методов классификации мы приходим к следующему выводу (Приложение 1). Метод k-ближайшего соседа, пожалуй, самый простой из всех алгоритмов для предсказания класса тестового примера. Очевидным недостатком метода k-NN является временная сложность прогнозирования. Кроме того, нейронные сети терпимы к зашумленным входным данным в виде изображений. Но в нейронной сети сложно понять структуру алгоритма. SVM был признан конкурентоспособным с лучшими доступными алгоритмами машинного обучения в классификации многомерных наборов данных. В SVM вычислительная сложность сводится к задаче квадратичной оптимизации, и легко контролировать сложность правила принятия решений и частоту ошибок. Недостаток SVM состоит в том, что трудно

определить оптимальные параметры, когда данные обучения не являются линейно разделимыми. Также SVM более сложен для понимания и реализации. Исходя из недостатков нейросетевых алгоритмов в качестве основы предложенного метода компьютерной диагностики болезней растений были взяты две методики: нечеткая логика и матрица смежности GLCM.

2. Разработка метода распознавания болезней листьев растений и исследование достижимой точности определения вида болезни

В данном разделе будет рассмотрен предлагаемый новый способ определения вида болезни листьев пшеницы, а также его реализация в среде Matlab R2016a.

2.1. Предобработка изображений

Удалось собрать выборку в размере 1000 изображений для здоровых 10-ти видов больных листьев пшеницы. Традиционно, выборка разделилась на обучающую (80 %) и тестовую (20 %), в отношении 4 к 1. Однако, изображения, полученные с БПЛА (Рис. 5) содержат слишком много неинформативного фона на заднем плане, для устранения которого необходимо проводить предварительную обработку изображений (Приложение 2).

(А)

(Б)



Рисунок 5. Примеры полученных с помощью БПЛА фотоизображений листьев здорового растения (А), заболевшего (Б).

Нормализацию изображения, как один из важных методов обработки изображений, стоит рассматривать в виде преобразования исходного изображения в другое для улучшения интерпретируемости или восприятия информации людьми или обеспечения благоприятных условий для других методов автоматической обработки

изображений. Оперирование исходным изображением в цветовой палитре RGB рассматривается, как один из фундаментальных процессов цифровой обработки изображений с целью повышения контрастности изображений на уровне серого, что облегчает последующие операции более высокого уровня, такие как обнаружение, идентификацию объектов или применение различного рода морфологических операций. Цветные изображения можно улучшить при наличии шумов, применив фильтрацию. Большинство задач по улучшению изображения обычно манипулируют гистограммой изображения, используя некоторую функцию преобразования, чтобы получить требуемое улучшение контрастности. Эволюционные алгоритмы и ранее использовались для улучшения изображения. В исследуемой работе предложен метод бинаризации изображений с помощью метода Отсу, основанного на анализе гистограммы тонального изображения, полученного в результате расчета вегетативного индекса исходного изображения. По сравнению с другими методами не требует операций постоянного подбора параметров. В то же время метод Отсу затрачивает меньше времени, чтобы достичь

оптимума. Согласно [7], полученные сегментированные изображения с помощью метода Отсу оказались лучше по сравнению с другими методами сегментации.

Использование текстурных характеристик изображений растений при дистанционном зондировании сельскохозяйственных растений не ново. Преимущество использования цветовых индексов состоит в том, что они подчеркивают определенный цвет, такой как зелень растений, который должен быть интуитивно понятным для сравнения человеком.

Индексы Вобекке без индексов строк и столбцов каждого пикселя рассчитываются, как:

$$(1.1)$$

$$(1.2)$$

$$(1.3)$$

$$(1.4)$$

где r , g и b - хроматические координаты.

Далее изображения представлялись в цветовой палитре RGB (Рис. 6), на основании которой удалось

получить

соответствующие

полутоновые

30

изображения. Полутоновые изображения подвергались фильтрации, используя фильтр Гаусса, затем повышалась контрастность изображения для выделения краев.

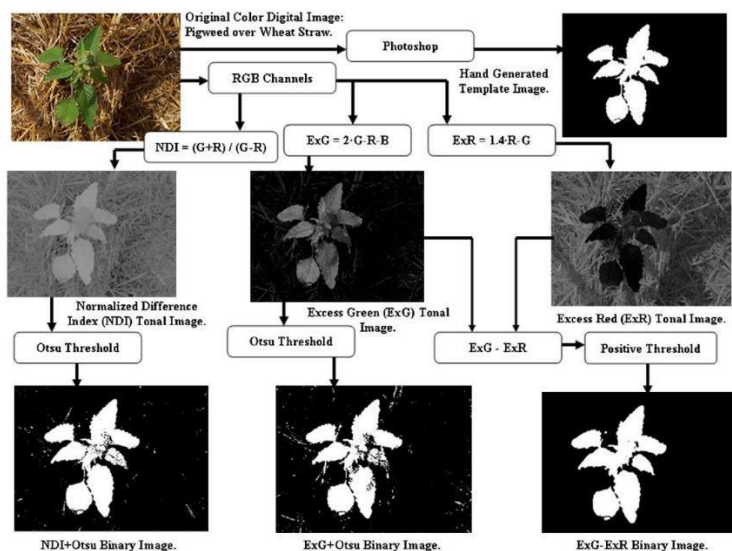


Рисунок 6. Блок-схема алгоритма бинаризации изображения.

Тональные изображения для индексов NDI и E^*G в некоторых случаях подвергались фильтрации, затем, используя пороговый метод Отсу, получались бинарные изображения. Дополнительное тональное изображение получалось в результате разности тональных изображений E^*G и E^*R . Применяя к

тональному изображению, на основе формул 1.1-1.4, $E \cdot G$ и $E \cdot R$ позитивный порог, установленный на отметке 0, можно получить бинарное изображение.

Для вычисления наклона изображения листа растения использовались функции `regionprops()` и `bwboundaries()` [3], позволяющие найти крайние точки регионов интереса, по которым можно судить о величине угла наклона растения на изображении. Для изображений листьев пшеницы производилась нормализация (Рис.7) изображений посредством поворота с последующей установкой изображения листа растения в вертикальное положение. Затем изображение масштабировалось и приводилось к размеру 100*300 пикселей. Этот процесс можно было бы выполнить и вручную, используя инструменты Paint или Gimp 2.0, но выборка состоит из 1000 необработанных изображений, поэтому написание программы автоматической нормализации носило первостепенное значение во избежание лишней потери времени на ручную сегментацию изображений.

Наборы изображений использовались для оценки и сравнения точности вегетативных 32

показателей: NDI, ExG и ExG-ExR. ExR был вычтен из ExG с нулевым порогом для создания двоичного изображения ExG-ExR. Предполагается, что яркие тональные области представляют потенциальные области зеленых растений.

В отличие от ExG или NDI с пороговым методом Отсу, индекс растительности ExG-ExR не требует специального расчета порога. Значения растительных пикселей все положительные, а оставшиеся фоновые пиксели все отрицательные.

Таким образом, индекс ExG-ExR способен самостоятельно генерировать двоичное изображение с постоянным пороговым значением, равным нулю.

Были получены полутоновые изображения из цветных, затем, производились необходимые трансформации и вычисления и на выходе получалось сегментированное (Рис. 7).

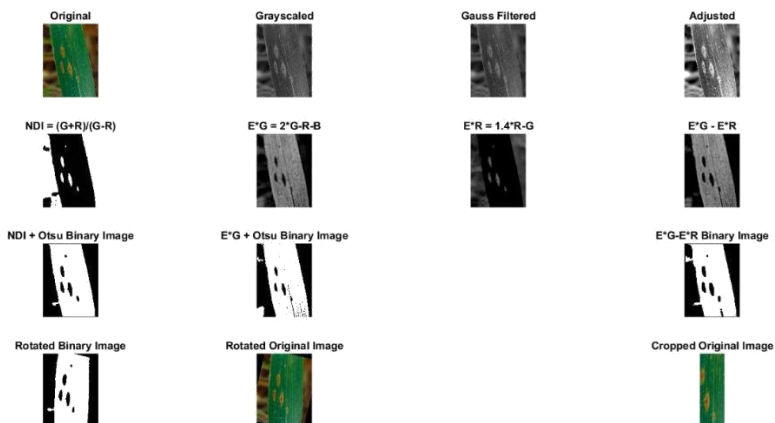


Рисунок 7. Этапы предобработки изображения.

Индекс растительности ($E^*G - E^*R$) был успешно протестирован для разделения растений и фона для наборов изображений. Этот индекс работал исключительно хорошо для отдельных растений и различных фоновых остатков почвы. $E^*G - E^*R$ также отлично работал с цифровыми изображениями с естественным освещением. Индекс $E^*G - E^*R$ показал более высокую точность выделения контуров объектов ближнего плана по сравнению с индексами $E^*G + Otsu$ и $NDI + Otsu$.

Тем не менее, как $E^*G - E^*R$, так и $E^*G - Otsu$ хорошо работали на изображениях, содержащих

неинформативный в рамках нашей задачи фон. Стоит отметить, что вегетативный индекс $NDI + Otsu$ является ненадежным для наборов изображений. Ручное извлечение шаблона изображения растения для сравнения с вегетативным индексом для определения точности является утомительным, поэтому необходимо использовать инструменты автоматической сегментации областей интереса на изображениях. Дополнительные исследования могут помочь определить факторы окружающей среды и освещения, которые влияют на точность вегетативного индекса. Этот подход может быть применен к вегетативным показателям, близким к инфракрасному [7], с соответствующим визуальным изображением.

Изображение вегетативного индекса является важным критическим этапом для машинного зрения идентификации листьев растений у полевых культур. Его основная цель заключается в предоставлении границ областей растений, представляющих интерес для последующего подробного анализа свойств растений. Полученный двоичный шаблон может быть использован для помощи в дальнейшем извлечении

отдельных цельных листьев для анализа особенностей формы и текстурной оценки обитания для характеристик вида. Этот метод полезен для изображений с преимущественно темными областями и областями низкой четкости из-за серьезных ограничений в условиях освещенности. Как плюс алгоритма - это его понятность и возможность быстрого преобразования алгоритма под другую задачу.

2.2. Матрица GLCM

Текстурные характеристики изображений могут быть вычислены на основе матрицы смежности (матрицы GLCM). Один из аспектов текстуры связан с пространственным распределением и пространственной взаимозависимостью значений яркости локальной области изображения [2].

Статистики пространственной взаимозависимости значений яркости вычисляются по GLCM-матрицам переходов значений яркости между ближайшими соседними точками. Матрица смежности GLCM [3] (или матрица совместной

встречаемости) уровней яркости представляет собой оценку плотности распределения вероятностей второго порядка, полученную по изображению в предположении, что плотность вероятности зависит лишь от расположения двух пикселей.

Перед вычислением матрицы GLCM и подсчётом показателей, исходные нормализованные цветные изображения растений, представленные в цветовой модели RGB, были разделены на шесть цветовых компонентов: R (красная), G (зеленая), B (синяя), RG (разностная составляющая R и G), RB (R и B) и GB (G и B). Далее вычислялись показатели GLCM-матриц: Contrast (Контрастность) – показатель степени локальных изменений в матрице GLCM, Correlation (Корреляция) - показатель вероятности совместного появления указанных пар пикселей, Energy (Энергия) - представляет сумму квадратов элементов в GLCM, также известен как равномерность или второй угловой момент, Homogeneity (Однородность) - степень близости распределения элементов в GLCM к диагонали GLCM для каждого изображения по формулам:

1) Contrast:

$$= \frac{1}{c-0} \sum_{i=0}^{c-1} \sum_{j=0}^{c-1} |i-j|^2 \cdot n_{ij} \quad (2.1)$$

2) Correlation:

$$= \frac{1}{c^2} \sum_{i=0}^{c-1} \sum_{j=0}^{c-1} \frac{\sum_{k=0}^{c-1} n_{ijk}}{n_{ij} \cdot n_{ik}} \cdot (i-j)^2 \quad (2.2)$$

3) Energy:

$$= \sum_{i=0}^{c-1} \sum_{j=0}^{c-1} n_{ij}^2 \quad (2.3)$$

4) Homogeneity:

$$= \sum_{i=0}^{c-1} \sum_{j=0}^{c-1} \frac{n_{ij}}{c^2} \quad (2.4)$$

Введение дополнительных цветовых каналов позволяет увеличить вектор признаков в 2 раза, с 12-ти до 24-х, что только благоприятно скажется на качестве распознавания болезней растений.

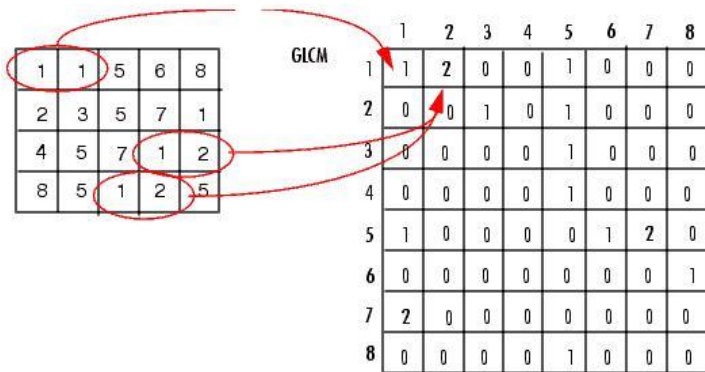


Рисунок 8. Пример построения матрицы GLCM из исходного изображения.

После взаимодействия с пикселями изображения при обработке изображения получаем матрицу чисел со значениями яркостей пикселей (см. рис. 8). Затем вычисляется значение элемента матрицы смежности $GLCM(1,1)$. Этот элемент содержит значение 1, потому что есть только один случай совместного появления в горизонтальном расположении двух пикселей со значением 1. Как показано на рис.4, элемент $GLCM(1,2)$ принимает значение 2, поскольку в данном примере два раза встречается совместное появление пикселей со значениями 1 и 2 в горизонтальном размещении. Аналогично заполняются остальные элементы матрицы $GLCM$.

Совершенно очевидно, что такие матрицы содержат информацию, которая может быть использована для последующего распознавания. Вычисляя признаки для различных изображений, можно получить многомерный вектор признаков текстур.

2.3. Построение гистограмм и аппроксимация

После вычисления матрицы GLCM по формулам 2.1 – 2.4 для всех изображений тестовой выборки мы все еще остаемся далеки от успешного распознавания различных видов болезней пшеницы. У нас имеется структура, содержащая вектора значений матриц GLCM всех статистических показателей, цветовых компонент и видов болезней пшеницы. Все значения находятся в промежутке $[0, 1]$. После чего, мы имеем полное право построить гистограммы, представляющие собой распределение плотности вероятности из элементов векторов признаков (Рис. 9).

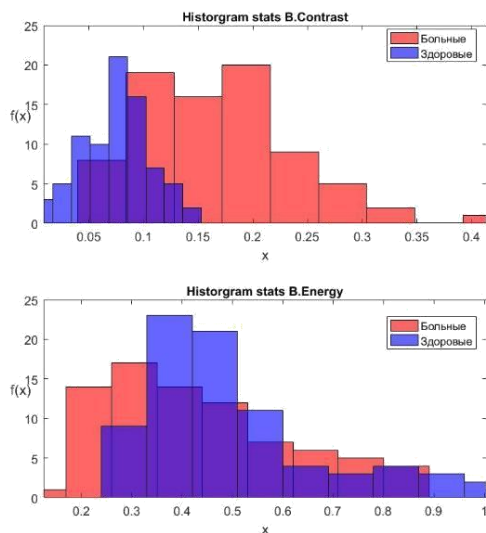


Рисунок 9. Пример построения гистограмм из элементов векторов признаков для двух болезней.

Однако, основываясь исходя из ЦПТ, какова бы ни была форма гистограммы, выборочное распределение всегда будет стремиться к нормальному. Это значит, что мы можем улучшить точность классификации предварительно аппроксимировав полученные гистограммы функцией Гаусса 1-го порядка, после чего плотность распределения примет вид нормального распределения (Рис. 10).

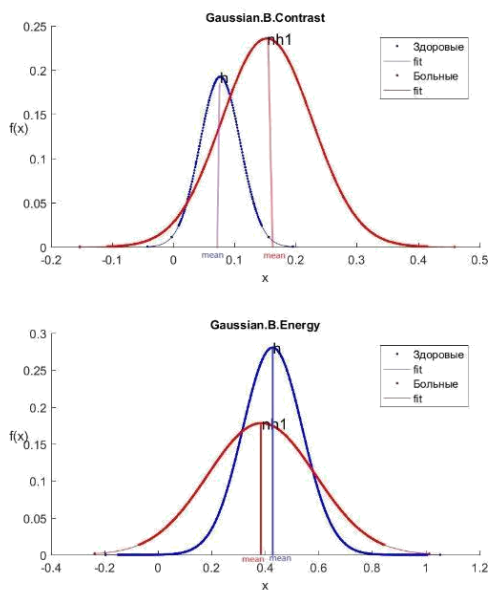


Рисунок 10. Пример аппроксимации гистограмм функцией Гаусса 1-го порядка.

В качестве функции аппроксимации была выбрана Гауссова 1-го порядка, так как является функцией нормального распределения. Также в процессе аппроксимации гистограммы продлеваются на расстояние в 2σ в обоих концах, даже если гистограммы имеют вид распределения с асимметрией.

Для построения эталонных описаний необходимо воспользоваться статистическими

данными аппроксимированных функций распределения. Так как распределение у нас нормальное, следовательно, математическое ожидание будет совпадать с центром распределения [1]. Именно матожидание будет отражать статистическую особенность для определенного вида болезни, в виду того, что в вычислении матожидания задействованы все элементы выборки.

Теперь необходимо и достаточно лишь найти максимальную точку кривой аппроксимации, после чего вычислить центр распределения. Дополнительными характеристиками распределений плотности также являются СКО и диапазон доверительного интервала.

2.4. Классификация болезней

После получения эталонных описаний обучающей выборки мы готовы к тому, чтобы начать распознавать изображения из тестовой выборки. Эталонными описаниями нормализованных изображений листьев являются не только математические ожидания значений ключевых параметров: Contrast, Correlation,

Energy, Homogeneity, но и доверительные интервалы функций распределения этих параметров. Они действительно получаются большими и значительно перекрываются друг с другом (Рис. 11).

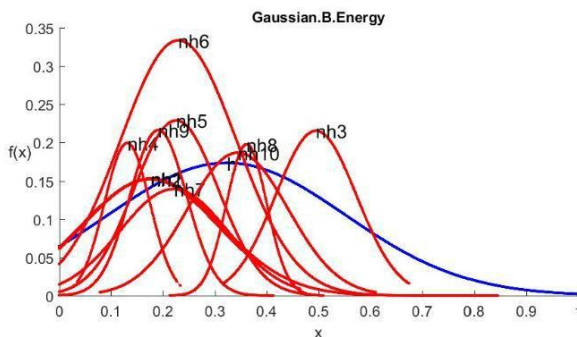


Рисунок 11. Пример плотно стоящих функций распределения для каждой из болезней.

Это, как кажется, делает невозможным процесс распознавания при большом количестве болезней. Для опровержения этой гипотезы была разработана отдельная программа (см. Приложение 3), моделирующая процесс распознавания посредством искусственного уменьшения доверительных интервалов эталонных описаний и возможного разброса этих параметров у анализируемых изображений до тех пор, пока не получилось надежное распознавание. Оказалось, что доверительные 44

интервалы эталонных описаний и возможный разброс этих параметров у анализируемых изображений нужно уменьшить в 5 раз. Это можно сделать, если анализировать не параметры Contrast, Correlation, Energy, Homogeneity у одного образца, а усредненные значения этих параметров у 25 образцов. Только таким способом можно распознать n-ное количество болезней.

Тестовая выборка также представлена нормализованными изображениями группами из 25-ти изображений листьев пшеницы конкретной болезни. Однако для однозначного правильного распознавания N типов болезней растений доверительные интервалы ключевых значений эталонных описаний должны не превышать величину $1/N$ [4]. Необходимое количество усредняемых значений было выбрано равным 25-ти исходя из результатов программы моделирования. Дополнительная программа моделирования (Приложение 4) показала, что при усреднении значений текстурных характеристик для групп из 25-ти изображений дают высшую вероятность, того, что значения, вычисленные с помощью матриц GLCM,

гарантированно попадут в интервалы эталонных описаний.

Итак, для каждой группы вычисляются усреднённые значения показателей GLCM-матриц. Каждое усредненное значение 24-х признаков группы тестовой выборки сравнивается с эталонными значениями обучающей выборки для здоровых и 10-ти больных групп растений. Результаты сравнений бинаризируются (значение результата сравнения полагается равным 1, если среднее значение параметров входит в доверительный интервал эталонного описания (Приложение 5) для конкретной текстурной характеристики, а также находится ближе к эталонному значению для изображения определенной группы растений, и 0 – если не входит в доверительный интервал или находится дальше, чем остальные).

Используемый подход по диагностике заболевания растения, основанный на определении максимальной близости значений ключевых параметров к эталонным описаниям, в литературе по нечеткой логике [5], имеет название: "по максимуму функции принадлежности".

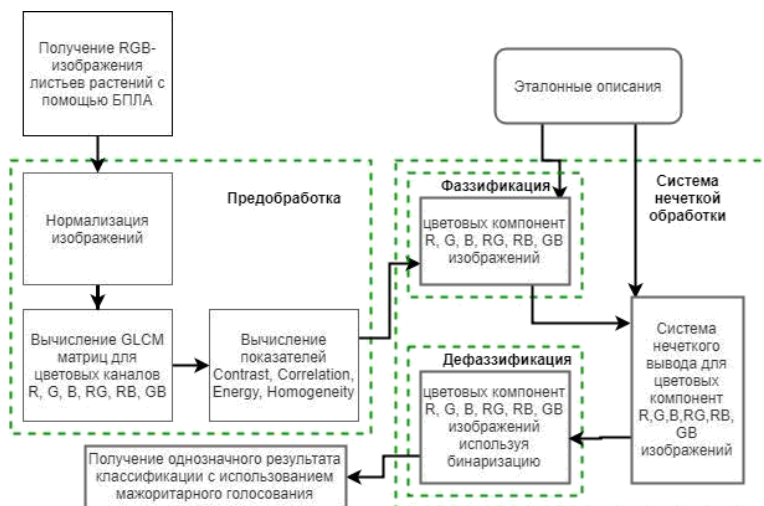


Рисунок 12. Структура системы диагностики заболеваний растений по изображениям листьев.

После бинаризации результатов необходимо принять окончательное решение к какой патологии соотносится исследуемая группа из 25-ти изображений растений. Было решено использовать мажоритарное голосование, позволяющее производить суммирование бинаризованных результатов и принять решение о принадлежности изображения растения к классу здоровых или одного из 10-ти видов больных (см. Рис. 12).

3. Реализация классификатора

Текущий раздел содержит рассмотрение на всех уровнях разработки, а также описание программной реализации классификатора. Дается подробное описание системы и всех ее функций.

3.1. Инструменты для разработки

Для портирования и моделирования работы системы была выбрана ОС Windows 10 и среда MatLab R2016a с использованием системы автодокументации кода MatLab Publish. В целях улучшения функциональности и повышения производительности система была переписана на язык C++, используя библиотеку OpenCV, системы документирования кода Doxygen и контроля версий GitHub (Рис. 13),

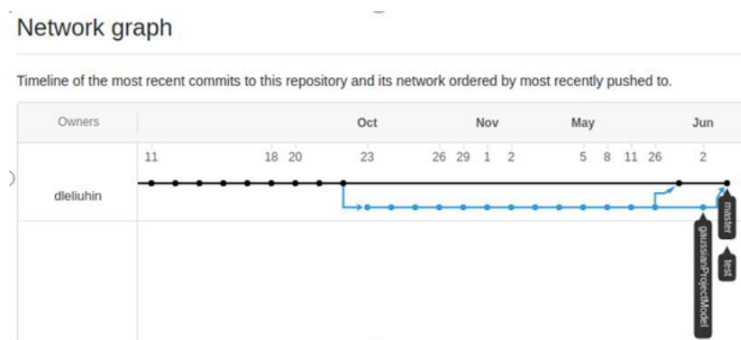


Рисунок 13. Пример истории коммитов в системе контроля версий Git.

ОС Linux Mint 64-bit Mate. В качестве IDE была выбрана QtCreator версии 4.9.2. Для отладки – GDB Debugger. Также была использована система визуализации покрытия кода тестами Lcov вместе с Gcov.

3.2. Системный уровень разработки

Вся нужная информация для системы - в изображениях, иерархически хранящихся в подпапках, названия которых совпадают с именованиями кодов болезней:

- 0 - здоровые растения
- 1 - Бурая ржавчина
- 2 - Желтая ржавчина
- 3 - Корневые гнили
- 4 - Мучнистая роса
- 5 - Пиренофороз
- 6 - Полосатая мозайка
- 7 - Септориоз-septoria
- 8 - Снежная плесень
- 9 - Стеблевая головня
- 10 - Темно-бурая пятнистость

Назначение информации в виде изображений – вычисление ключевых показателей эталонных описаний для последующего распознавания, которое осуществляется после преобразования изображений в модуле нормализации, а затем и в графическом модуле, который, в свою очередь, принимает на вход текстурные характеристики всей обучающей выборки и, затем, с помощью блока обработки, выдает эталонные описания, на основании которых модуль классификации может распознавать входные изображения тестовой выборки. После всех операций блок логирования записывает промежуточные и конечные результаты в файлы-логи.

3.3. Архитектурный уровень разработки

Каждый блок системы на архитектурном уровне имеет свое уникальное назначение:

- Блок нормализации – отвечает за преобразование изображений с лишним фоном и неинформативной частью в обрезанные и улучшенные изображения.

- Графический блок – необходим для визуализации построения гистограмм и функций аппроксимации Гаусса 1-го порядка.
- Блок обработки – несет целевое значение для вычисления эталонных описаний, основываясь на построенных функциях аппроксимации Гаусса 1-го порядка.
- Блок классификации – имеет в своем составе уже вычисленные эталонные описаний и принимает на вход изображения тестовой выборки, также прошедшие через блок нормализации, и выносит вердикт о результатах распознавания.
- Блок логирования - выдает результаты распознавания в текстовом или табличном виде в конкретные файлы.

3.4. Программная реализация

Программная реализация заключалась в разделении составляющих системы на классы и подклассы (Рис. 14):

- Класс DataParser – отвечает за рекурсивное вытягивание всех изображений из подпапок строго определенном формате и иерархии. Содержит методы для получения имен файлов с изображениями в подпапках и выдаче векторов имен следующему классу ImageNormalizer.

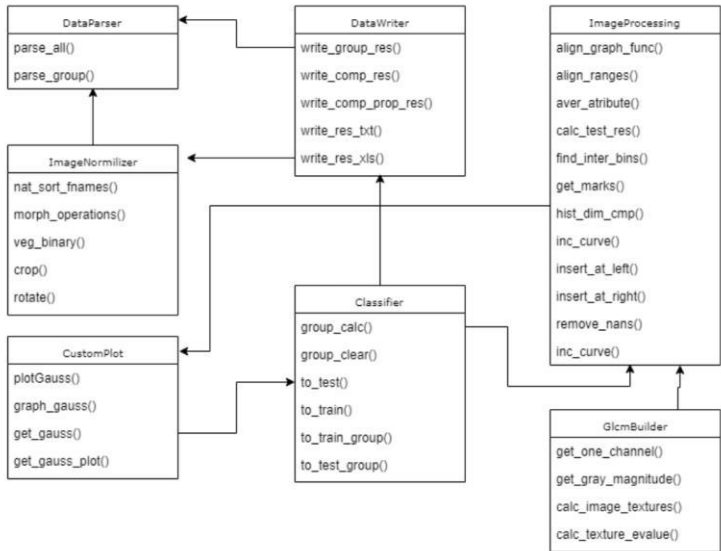


Рисунок 14. Диаграмма классов разработанной системы.

- Класс ImageNormalizer – нормализует изображения, применяя методы для проведения морфологических операций, бинаризации, а также поворота, смещения и

масштабирования вырезанных областей-блобов, после чего отдает команду на исполнение методов классов `DataWriter` и `ImageProcessing`.

- Класс `DataWriter` – записывает информацию, содержащую промежуточные и результирующие логи распознавания, а также ключевые значения эталонных описаний.
- Класс `ImageProcessing` – основное ядро системы, содержащий все методы и алгоритмы для вычисления эталонных описаний, матриц GLCM и операций для фильтрации данных.
- Подкласс `Glcmbuilder` – вычисляет текстурные характеристики изображений, используя матрицы GLCM (Приложение 7).
- Класс `CustomPlot` – выводит функции аппроксимации Гаусса 1-го порядка на графики, что позволяет контролировать корректность исполнения программы.
- Класс `Classifier` – классифицирует усредненные тестовые изображения и отдает команду классу `DataWriter` для записи логов в файлы.

3.5. Тестирование

Тестирование является важной частью процесса разработки ПО и позволяет выявить дефекты (баги) в различных частях программы. Так как, кроме вывода графиков плотностей распределения, других взаимодействий с программой, используя UI не было, следовательно, было покрыть весь код исключительно модульными тестами. Так как средой разработки являлся QtCreator, он имеет в своем составе встроенные средства для разработки модульных тестов.

В Qt за юнит-тестирование отвечает модуль QTestLib (testlib). Он предоставляет разработчику набор макросов для тестирования. Всего было реализовано свыше 143 Unit-теста (Приложение 8) в целом для каждого модуля программы. Благодаря использованию встроенной библиотеки тестирования QTest и Gcov с Lcov было обнаружено незначительное количество багов, которые впоследствии были исправлены. Также, тестирование системы показало, что все требования спецификации выполняются верно.

Результаты экспериментального исследования

В первых версиях программы предложенный подход к распознаванию болезней растений по изображениям листьев был проверен отдельно на 40 изображениях листьев здоровых и 40 больных эвкалиптов заболеванием Gall wasp и показал положительный результат в 95% случаях. На тот момент изображения разбивались на группы из 5-ти случайных изображений (всего 16 групп), которые при проведении 1000 опытов сортировались в случайном порядке и опять собирались в новые группы.

Ввиду того, что размер выборки был ограничен, было решено для подтверждения достоверности работы программы провести моделирование, используя полученные эталонные описания обучающей выборки.

Конечная версия программы содержится в 5000 строках кода на MatLab и 12000 на C++. Моделирование и тестирование программы позволило

определить некоторые ошибки в классификаторе, которые впоследствии были устранены.

Результаты моделирования и реальной работы программы совпали. Удалось добиться высокого уровня распознавания в около 90% случаях (Приложение 6). Сравнивая полученные результаты с результатами авторов, использующих современные нейросетевые алгоритмы, стоит отметить, что предложенный метод ничем не уступает по качеству нейросетевым, а некоторые из них даже опережает.

В будущем планируется улучшить уровень распознавания предложенным методом введя предварительное улучшение качества изображений. Также планируется расширить датасеты не только изображениями листьев пшеницы, но и листьями других видов растений, что позволит подтвердить гипотезу о том, что классификатор способен распознавать любые типы болезней разных видов растений. В течение следующих двух лет, в рамках работ над выпускной квалификационной работой магистра планируется оснастить беспилотный квадрокоптер реализованным ПО, чтобы проверить работоспособность системы в реальных условиях.

Заключение

1. Проанализированы известные методы диагностики болезней растений по изображениям листьев.
2. Предложен метод диагностики болезней растений по нормализованным изображениям листьев, основанный на вычислении 4-х текстурных характеристик изображений, вычисленных с использованием матриц GLCM для 6-ти цветовых компонент изображений: R, G, B, RG, RB, GB.
3. Предложен и реализован алгоритм нормализации изображений при помощи вегетативных индексов Вобекке с применением морфологических операций, поворота и масштабирования.
4. Доказано путем моделирования, что можно достигнуть достоверности распознавания около 90% за счет бинаризации результатов и мажоритарного голосования.
5. Произведена экспериментальная проверка метода диагностики на 1000 изображений листьев пшеницы 10-ти видов болезней, подтвердившая результаты, полученные при моделировании.

6. Разработано ПО для компьютерной диагностики заболеваний растений по текстурным характеристикам изображений листьев.

Библиографический список

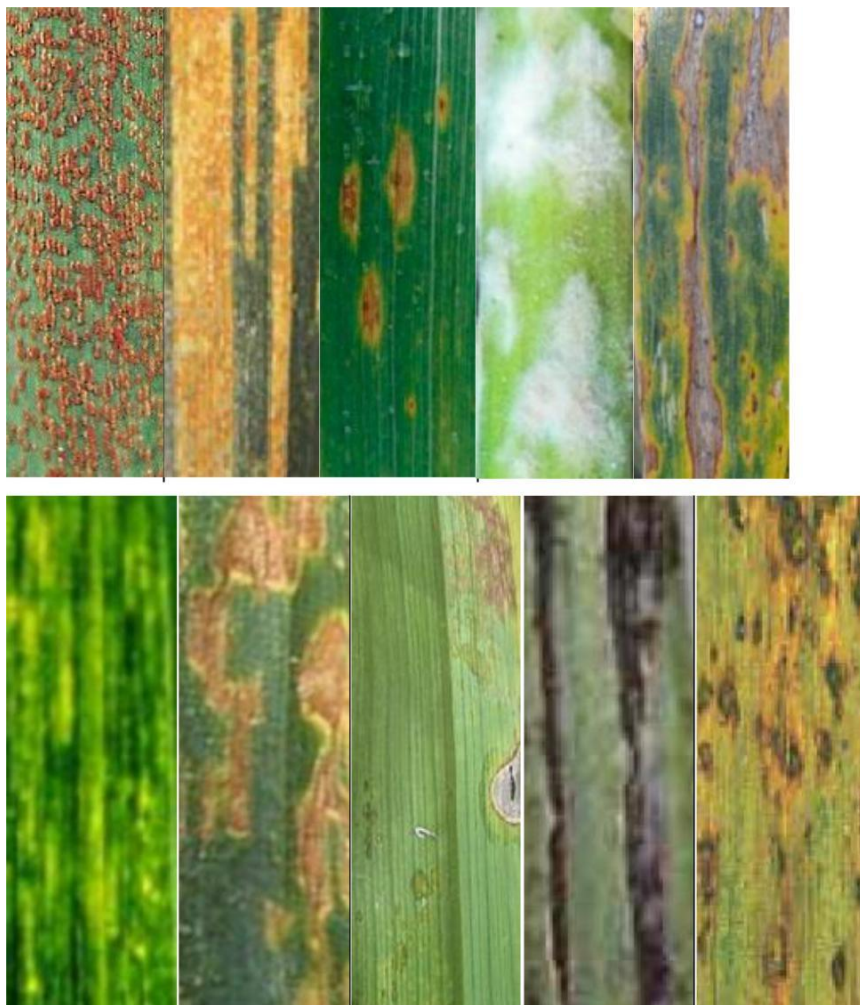
1. Битюков С.И. Сравнение гистограмм в физических исследованиях / С.И. Битюков, А.В. Максимушкина, В.В. Смирнова. – Обнинск.: Изд-во Национального исследовательского ядерного университета "МИФИ", 2016. - №1. – с. 81-90.
2. Денисюк В.С. Алгоритмы выделения особенностей на изображениях с целью классификации заболеваний растений // Конструирование и оптимизация параллельных программ. - Новосибирск, 2008. - №16. - с.171-182.
3. Сергиенко А.Б. Цифровая обработка сигналов. - СПб.: Изд-во Питер, 2013г. – 608 с.
4. Тутьгин В.С. Автоматизация физического эксперимента: учеб. пособие / В.С. Тутьгин. – СПб.: Изд-во Политехн. ун-та, 2013. – 200 с.
5. Штовба С. Д. Введение в теорию нечетких множеств и нечеткую логику.–URL:
<http://matlab.exponenta.ru/fuzzylogic/book1/>. (дата обращения: 26.05.2019).
6. Diptesh Majumdar, Dipak Kumar Kole, Aruna Chakraborty, and Dwijesh Dutta Majumder. "An

- Integrated Digital Image Analysis System for Detection, Recognition and Diagnosis of Disease in Wheat Leaves." In Proceedings of the Third International Symposium on Women in Computing and Informatics, pp. 400- 405. ACM, 2015.
7. George E. Meyer, Joao Camargo Neto. Verification of color vegetation indices for automated crop imaging applications / Computers and electronics in agriculture, Vol.63., 2008, 282-293 p.
 8. Jayme Garcia Arnal Barbedo. Digital image processing techniques for detecting, quantifying and classifying plant diseases / Barbedo SpringerPlus 2013, 2:660.
 9. Jyotismita Chaki, Ranjan Parekh. Designing an Automated System for Plant Leaf Recognition / IJAET, Vol.2, Issue. 1, pp: 149-158, 2012.
 10. Liu, T., Wu W., Chen W., Sun C., Zhu X., Guo W., Automated image-processing for counting seedlings in a wheat field. *Precis. Agric.* 2016, 17, 392–406.

Приложение 1. Методы классификации болезней растений по изображениям листьев.

№	Методы	Описание	Преимущества	Недостатки
1	KNN	Вычисляет минимальное расстояние между точками.	Легкая реализация и качественный уровень распознавания.	Медленно обучается. Не устойчив к зашумленным данным при большом размере выборки.
2	SVM	Переводит исходные вектора в пространство более высокой размерности и ищет разделяющие гиперплоскости с максимальным зазором в этом пространстве.	Точность прогноза высока. Надежно работает, когда пример обучения содержит ошибки.	Длительное время обучения, трудно понять изученную функцию. Большое количество опорных векторов для выполнения задачи классификации.
3	PNN	Работает с четырехслойной структурой, которая включает в себя расчет расстояния, хранение переменной предиктора и сравнение голосов.	Быстрый и точный.	Требуется большое место для хранения данных.
4	ANN	Базовая форма представлена в виде многослойного перцептрона, который обновляет вес посредством базового распространения.	Хороший потенциал для обнаружения болезни листьев растений.	Требует много времени на обучение.
5	Нечеткая логика	Представляет собой нечёткое множество, определяемое при помощи обобщенного понятия характеристической функции.	Высокая скорость, предпочтительно с ограниченной точностью.	Большая размерность вектора признаков. Плохая производительность.
6	GLCM	Позволяет получить статистическую информацию в виде текстурных характеристик изображения.	Высокая скорость работы и точность прогноза. Легко реализуется и моделируется.	Медленно обучается.

Приложение 2. Нормализованные изображения для каждого типа болезни листьев растений.



Приложение 3. Листинг программы моделирования для расчета доверительных интервалов в среде MatLab R2016a.

```
clc;
clear all;
close all;

N=100000; %> Количество точек расчета.
kmax=25; %> Количество точек усреднения (размер окна усреднения).
N1=N/kmax; %> Количество усредненных сигналов.

k=0;
for i=1:N
    q(i)=(randn(size(N))); %> Генерация шума.
end

for k=1:N1
    sum_q1(k)=0;
    for j=1:kmax
        m=k*j;
        q1(j)=q(m);
        sum_q1(k)=sum_q1(k)+q1(j);
    end
    sum_q1(k)=sum_q1(k)/kmax;
end

%> Доверительный интервал ЗСКО случайного сигнала q.
dov_int_1=3*std(q)

%> Доверительный интервал ЗСКО после усреднения блоками по kmax отсчетов.
dov_int_2=3*std(sum_q1)

figure
i=1:N;
plot(i,q(i));
axis tight; %> Диапазон X и Y по осям точно соответствует Xmax и Kmax.
title('Случайный шум до усреднения');

figure
k=1:kmax;
plot(k,sum_q1(k));
axis tight; %> Диапазон X и Y по осям точно соответствует Xmax и Kmax.
title('Случайный шум после усреднения');

figure;
hist(q,7);
title('Гистограмма шума до усреднения');
figure;
hist(sum_q1,7);
title('Гистограмма шума после усреднения');

pause;
close all;
clear;
```


Приложение 4. Листинг программы моделирования для диагностики 10 болезней по эталонным описаниям в среде MatLab R2016a.

Моделирование функций принадлежности и диагностики болезней с использованием параметров функций GLCM. Вычисляется процент правильных результатов диагностики при многократном повторении диагностики исходные значения параметров задаются с различными случайными отклонениями от эталонных значений.

```
clc;
clear;

m=1000;%количество статистических испытаний.
k=7;%номер заболевания на входе
kb_in=2^k;%входной код заболевания (нужен для определения правильности диагностики)
S=0.03;%СКО случайных отклонений от эталонных значений
BT =0.9;%коэффициент для порога бинаризации
x = 0:14;%номера возможных болезней

% Эталонные описания 10 болезней для цветовой компоненты R.
Rcon = [0.32 0.25 0.12 0.25 0.144 0.19 0.186 0.156 0.184 0.129];
Rcor = [0.84 0.93 0.87 0.94 0.93 0.966 0.942 0.91 0.93 0.915];
Ren = [0.12 0.11 0.355 0.14 0.18 0.144 0.17 0.22 0.18 0.275];
Rhom = [0.85 0.89 0.93 0.94 0.907 0.91 0.91 0.92 0.918 0.936];

result=0;
global_result=0;
for j1=1:m
    for i=1:10
        Voting_global(i)=0;
    end

    noise1=randn(1,10); noise2=randn(1,10); noise3=randn(1,10); noise4=randn(1,10);
    for i=1:10
        Sum_res_vot(i)=0;
        Rcon_fp(j)=1-abs(Rcon(j)-Rcon(k)+S*noise1(j));%функция принадлежности 1
        Rcor_fp(j)=1-abs(Rcor(j)-Rcor(k)+S*noise2(j));%функция принадлежности 2
        Ren_fp(j)=1-abs(Ren(j)-Ren(k)+S*noise3(j));%функция принадлежности 3
        Rhom_fp(j)=1-abs(Rhom(j)-Rhom(k)+S*noise4(j));%функция принадлежности 4
    end;

    %Бинаризация R_con
    for i=1:10
        Res_con_R(i)=0;
        Res_cor_R(i)=0;
        Res_en_R(i)=0;
        Res_hom_R(i)=0;

        if(Rcon_fp(i)>BT) %BT(binartization threshold)-порог бинаризации для функции принадлежности
            Res_con_R(i)=1;
        end;
```

```

if (Rcor_fp(i)>BT)
    Res_cor_R(i)=1;
end;

if (Ren_fp(i)>BT)
    Res_en_R(i)=1;
end;

if (Rhom_fp(i)>BT)
    Res_hom_R(i)=1;
end;
end

ResCon_R=Res_con_R;%%%
ResCor_R=Res_cor_R;%%%
ResEn_R=Res_en_R;%%%
ResHom_R=Res_hom_R;%%%

for i=1:10
    Sum_res_vot_R(i)=0;
end;
for i=1:10
    Sum_res_vot_R(i)=ResCon_R(i)+ResCor_R(i)+ResEn_R(i)+ResHom_R(i);
end

Sum1R=Sum_res_vot_R;%%%
Sum1Rmax=max(Sum1R);
for i=1:10
    Voting_res_R(i)=0;
    if (Sum1R(i)==Sum1Rmax)
        Voting_res_R(i)=1;
    end
end
Voting_result_R=Voting_res_R;%%%
% Voting_global
for i=1:10
    Sum1global(i)=Voting_result_R(i)+Voting_result_G(i)+Voting_result_B(i)+Voting_result_RG(i)+Voting_result_RB(i)+Voting_
    result_GB(i);
end
Sum1globalRes=Sum1global;%%%
Sum1globalmax=max(Sum1global);

for i=1:10
    Voting_global(i)=0;
    if (Sum1global(i)==max(Sum1global))
        Voting_global(i)=1;
    end
end
VotingGlobalRes=Voting_global;%%%
kb_=0;
for i=1:10
    if (Voting_global(i)==1)
        kb_ = kb_+2^i ;
    end
end

kb_in=kb_in;
kb_out=kb_ ;
if (kb_in==kb_out)
    global_result=global_result+1;
end
end

GR=global_result
GR_percent=GR/m*100

close all;
clear

```

Приложение 5. Таблица эталонных описаний математических ожиданий для всех групп патологий.

		h	nh1	nh2	nh3	nh4	nh5	nh6	nh7	nh8	nh9	nh10
R	Contrast	0.1165	0.32	0.25	0.12	0.25	0.144	0.19	0.186	0.156	0.184	0.129
	Correlation	0.9200	0.84	0.93	0.87	0.94	0.93	0.966	0.942	0.91	0.93	0.915
	Energy	0.3955	0.12	0.11	0.355	0.14	0.18	0.144	0.17	0.22	0.18	0.275
	Homogeneity	0.9395	0.85	0.89	0.93	0.94	0.907	0.91	0.91	0.92	0.918	0.936
G	Contrast	0.0990	0.32	0.233	0.127	0.222	0.177	0.1995	0.159	0.1805	0.169	0.1445
	Correlation	0.9355	0.7875	0.946	0.8265	0.929	0.86	0.928	0.8635	0.806	0.9445	0.957
	Energy	0.3545	0.132	0.1935	0.3115	0.155	0.2565	0.1385	0.2155	0.318	0.1915	0.278
	Homogeneity	0.9499	0.886	0.892	0.9318	0.9265	0.921	0.9075	0.918	0.92	0.9365	0.9325
B	Contrast	0.1140	0.309	0.233	0.1345	0.186	0.184	0.2105	0.1705	0.148	0.139	0.1255
	Correlation	0.8710	0.8955	0.8	0.723	0.9385	0.8585	0.892	0.888	0.766	0.9035	0.895
	Energy	0.3230	0.177	0.1755	0.4955	0.131	0.2255	0.23	0.221	0.3615	0.191	0.3445
	Homogeneity	0.9445	0.8495	0.897	0.944	0.9155	0.9035	0.867	0.9185	0.922	0.9195	0.954

RG	Contrast	0.0024	0.0155	0.039	0.02255	0.001	0.00145	0.00075	0.0193	0.00725	0	0.00375
	Correlation	0.8743	0.8935	0.808	0.904	0.686	0.925	0.95589	0.92	0.856	0.0024	0.852
	Energy	0.9975	0.482	0.396	0.9825	0.9977	0.99	0.9525	0.977	0.9515	0.99974	0.96
	Homogeneity	0.9975	0.996	0.9796	0.98835	0.999	0.99975	0.99967	0.99	0.9967	0.99993	0.9979
RB	Contrast	0.0077	0.1865	0.07	0.049	0.0195	0.007	0.0485	0.0645	0.0428	0.0127	0.0415
	Correlation	0.9170	0.961	0.979	0.899	0.86	0.9405	0.9745	0.969	0.977	0.9125	0.9645
	Energy	0.9720	0.182	0.208	0.3325	0.396	0.21	0.4295	0.299	0.2875	0.6675	0.4725
	Homogeneity	0.9964	0.947	0.965	0.974	0.98625	0.99445	0.97495	0.9666	0.97885	0.9884	0.978
GB	Contrast	0.0181	0.105	0.078	0.0397	0.05	0.006	0.0435	0.0625	0.0419	0.03195	0.052
	Correlation	0.9600	0.9255	0.941	0.889	0.8735	0.9135	0.884	0.9475	0.9375	0.9616	0.964
	Energy	0.6360	0.382	0.4095	0.4725	0.609	0.285	0.53	0.2285	0.5205	0.495	0.4985
	Homogeneity	0.9917	0.948	0.96035	0.97675	0.9759	0.996	0.97825	0.96945	0.98295	0.9845	0.9746

Приложение 6. Результаты программы в текстовом виде для диагностики 10 болезней по эталонным описаниям в среде MatLab R2016a.

1	imagenames	Channel	Stats	Test	h	in	d	r	nh1	in	d1	r	nh2	in	d2	r	nh3	in	d3	r
2																				
3	Remained																			
4		R	Contrast	0.180	0.12	1	0.064	0	0.32	1	0.140	0	0.24	1	0.060	0	0.12	1	0.062	0
5		R	Correlation	0.928	0.92	1	0.008	0	0.84	1	0.086	0	0.93	1	0.001	1	0.87	1	0.057	0
6		R	Energy	0.273	0.40	1	0.122	0	0.12	0	0.156	0	0.11	0	0.164	0	0.35	1	0.082	0
7		R	Homogeneity	0.911	0.94	1	0.028	0	0.85	1	0.064	0	0.89	1	0.023	0	0.94	1	0.025	0
8		G	Contrast	0.188	0.10	1	0.089	0	0.32	1	0.132	0	0.23	1	0.045	0	0.13	1	0.061	0
9		G	Correlation	0.953	0.94	1	0.018	0	0.79	1	0.166	0	0.95	1	0.007	0	0.83	0	0.127	0
10		G	Energy	0.199	0.35	1	0.155	0	0.13	1	0.067	0	0.19	1	0.006	1	0.31	0	0.112	0
11		G	Homogeneity	0.908	0.95	1	0.042	0	0.89	1	0.021	0	0.89	1	0.016	0	0.93	1	0.024	0
12		B	Contrast	0.158	0.11	1	0.044	0	0.31	1	0.151	0	0.23	1	0.075	0	0.13	1	0.024	0
13		B	Correlation	0.912	0.87	1	0.041	0	0.90	1	0.017	0	0.80	1	0.112	0	0.72	0	0.189	0
14		B	Energy	0.309	0.32	1	0.014	1	0.18	1	0.132	0	0.18	1	0.133	0	0.50	0	0.187	0
15		B	Homogeneity	0.922	0.94	1	0.023	0	0.85	1	0.072	0	0.90	1	0.025	0	0.94	1	0.022	0
16		RG	Contrast	0.014	0.00	1	0.016	0	0.02	1	0.002	1	0.04	1	0.025	0	0.02	1	0.009	0
17		RG	Correlation	0.824	NaN	0	1.000	0	0.89	1	0.069	0	0.81	1	0.016	1	0.90	1	0.080	0
18		RG	Energy	0.873	1.00	1	0.124	0	0.48	1	0.391	0	0.40	1	0.477	0	0.98	1	0.109	0
19		RG	Homogeneity	0.993	1.00	1	0.004	0	1.00	1	0.003	1	0.98	1	0.014	0	0.99	1	0.005	0
20		RB	Contrast	0.027	0.01	1	0.020	0	0.19	1	0.159	0	0.07	1	0.043	0	0.05	1	0.022	0
21		RB	Correlation	0.873	0.92	1	0.044	0	0.96	1	0.088	0	0.98	1	0.106	0	0.90	1	0.026	0
22		RB	Energy	0.766	0.97	1	0.206	0	0.18	0	0.584	0	0.21	0	0.558	0	0.33	1	0.434	0
23		RB	Homogeneity	0.986	1.00	1	0.010	0	0.95	1	0.039	0	0.96	1	0.021	0	0.97	1	0.012	0
24		GB	Contrast	0.042	0.02	1	0.024	0	0.11	1	0.063	0	0.08	1	0.036	0	0.04	1	0.002	0
25		GB	Correlation	0.931	0.96	1	0.029	0	0.93	1	0.006	1	0.94	1	0.010	0	0.89	1	0.042	0
26		GB	Energy	0.544	0.64	1	0.092	0	0.38	1	0.162	0	0.41	1	0.134	0	0.47	1	0.071	0
27		GB	Homogeneity	0.979	0.99	1	0.013	0	0.95	1	0.031	0	0.96	1	0.019	0	0.98	1	0.002	0
28																				
29	Result:	Count in ranges	= 24 ;	Count bin result	= 5 ;	Diseased plant number	= nh8.													
30																				

Приложение 7. Пример описания интерфейса класса `Glcmbuilder`.

```
#ifndef GLCMBUILDER_H
#define GLCMBUILDER_H
/**
 * \file modules/glcmbuilder.h
 *
 * \brief Interface core class Glcmbuilder
 *
 * \author Leliuhin Dmitrii
 *
 * \version 1.0
 *
 * \date $Date: 23/05/2019 02:47:20 $
 *
 * Contact: dleliuhin@gmail.com
 *
 * Created on:
 * 23/05/2019 */
#include "container.h"

/*! \class Glcmbuilder
 * \brief A core class.
 * A class that have methods for calc statistic textural image features.
 */
class Glcmbuilder
{
public:
    /*! \fn static cv::Mat get_one_channel(cv::Mat &src,
     *                                     EnumChannels channel);
     * \brief
     * Extract a channel from RGB Image.
     * \param src - input image.
     * \param channel - needed channel of input image color space.
     * \return a extracted channel from multichannel image.
     */
    static cv::Mat get_one_channel(cv::Mat &src, EnumChannels channel);

    /*! \fn static cv::Mat get_gray_magnitude(cv::Mat &src,
     *                                     GrayLevel level = GRAY_8);
     * \brief
     * Magnitude all pixels of Gray Image, and
```

```

*      Magnitude Level can be chosen in 4/8/16;
*      \param src - input matrix.
*      \param level - needed level of GLCM matrix magnitude.
*      \return a GLCM magnitude.
*/
static cv::Mat get_gray_magnitude(cv::Mat &src, GrayLevel level = GRAY_8);

/*!\fn static void calc_image_textures(cv::Mat &src,
* GrayLevel level = GRAY_8, * bool to_adjust = false);

* \brief
* Calculate image textures of src Mat,
* which is including Contrast, Correlation, Energy, Homogeneity.
* \param src - input matrix.
* \param level - needed level of GLCM matrix magnitude.
* \param to_adjust - flag, if need to adjust image.
* \return void.
*/
static void calc_image_textures(cv::Mat &src,
                                GrayLevel level = GRAY_8,
                                bool to_adjust = false);

/*!\fn static void calc_image_textures(cv::Mat &src,
* GrayLevel level = GRAY_8,
* bool to_adjust = false);
* \brief
* Calculate image texture eigen values of src Mat,
* which is including Contrast, Correlation, Energy, Homogeneity.
* \param src - input matrix.
* \param evaluate - texture value need to calc.
* \param level - needed level of GLCM matrix magnitude.
* \return void.
*/
static void calc_texture_evalue(cv::Mat &src,
                                TextureValues &evaluate,
                                GrayLevel level = GRAY_8);
};
#endif // GLCMBUILDER_H

```

Приложение 8. Результаты тестирования методов класса `Glcmbuilder`.

***** Start testing of `Test_Glcmbuilder` *****

Config: Using `QtTest` library 5.9.2, `Qt` 5.9.2 (x86_64-little_endian-lp64 shared (dynamic) release build; by `GCC` 5.3.1 20160406 (Red Hat 5.3.1-6))

PASS : `Test_Glcmbuilder::initTestCase()`

PASS : `Test_Glcmbuilder::test_get_one_channel()`

PASS : `Test_Glcmbuilder::test_get_gray_magnitude()`

PASS : `Test_Glcmbuilder::test_calc_image_textures()`

PASS : `Test_Glcmbuilder::test_calc_texture_evaluate()`

PASS : `Test_Glcmbuilder::test_calc_glcmbuilder()`

PASS : `Test_Glcmbuilder::test_normalize_mat()`

PASS : `Test_Glcmbuilder::test_calc_evaluate()`

PASS : `Test_Glcmbuilder::cleanupTestCase()`

Totals: 9 passed, 0 failed, 0 skipped, 0 blacklisted, 14ms

***** Finished testing of `Test_Glcmbuilder` *****