

## 2 СИСТЕМНОЕ ПРОЕКТИРОВАНИЕ

После анализа предметной области задачи дипломного проекта были определены цели разработки программного средства:

- разработать приложение для анализа зерновой культуры на основе предоставляемых изображений;
- предоставить пользователю возможность производить предобработку исходного изображения используя предлагаемые инструменты (цветовые системы, изменение контраста и яркости, морфологические операции, цифровые фильтры);
- обеспечить удобный и понятный интерфейс пользователя;
- предоставить возможность пользователю выбирать данные для обучения классификатора;
- обеспечить полной и понятной документацией об архитектуре и возможностях программного средства;

Разрабатываемое программное средство должно поддерживать следующие функции:

- ввод изображений для анализа;
- предварительная обработка и фильтрация изображений;
- анализ зерна на содержание примесей других культур либо мусорных примесей;
- отображение результатов в виде отчетов;

### 2.1 Структура программного средства

В соответствии с методологией системного подхода в разработке архитектуры, программное средство разбивается на совокупность сущностей, представленных на структурной схеме (см. чертеж ГУИР.400201.035 С1).

На каждый блок в программном средстве возлагается определенная задача. Кроме того каждый блок так или иначе связан с другими блоками, чтобы обеспечить работоспособность всего приложения в целом. Данная связь реализуется посредством обмена данными между блоками.

Блок регистрации изображений – как правило, это аппаратная часть, которая либо производит съемку самих изображений, либо принимает их по сети для дальнейшей передачи в приложение.

База изображений – совокупность изображений для анализа. Изображения могут различаться по качеству, разрешению, нейтральности относительно семян заднего фона. От этих параметров напрямую зависит качество обработки и анализа.

Модуль интерфейса пользователя – разновидность интерфейсов, в котором одна из сторон это человек, а другая компьютер. Модуль предоставляет средства для коммуникации пользователя с программой. Качественная разработка данного блока является важной частью процесса

создания программного средства, так как пользователю должно быть интуитивно понятно как им пользоваться.

Модуль выбора цветовой системы – блок, позволяющий изменить цветовую систему изображения. Используя данный подход можно разложить изображение на каналы с другими параметрами и выбрать тот, который лучше отвечает требуемой задаче.

Модуль улучшения контраста и яркости – блок, отвечающий за предварительную обработку изображения. Данный модуль используется при недостаточном контрасте или яркости объектов изображения для дальнейшей качественной обработки. Используется алгоритм гамма-коррекции.

Модуль бинаризации – модуль, отвечающий за перевод изображения в бинарный вид, где будут присутствовать только два вида цвета: черный и белый. Данная операция является промежуточной, но вместе с тем обязательной для дальнейшего анализа. Для выполнения данной операции используется на выбор либо алгоритм адаптивной бинаризации, либо метод Оцу.

Модуль алгоритмов фильтрации – блок, позволяющий дообработать изображение после бинаризации. Так как на изображении часто присутствует шум, применяя цифровые фильтры в разных комбинациях, можно достичь нужной чистоты снимка, что является залогом правильной кластеризации, так как шум будет определен далее как объект. В данном блоке был реализован медианный фильтр.

Модуль вычисления геометрических признаков – модуль, относящийся непосредственно к анализу зерна. Модуль, вычисляющий признаки объектов, основанных на геометрических аспектах объекта.

Модуль вычисления яркостных характеристик – модуль, относящийся непосредственно к анализу зерна. Отвечает за исследование яркостных и цветовых характеристик зерновой смеси.

Модуль вычисления текстурных характеристик – модуль, относящийся непосредственно к анализу зерна. Отвечает за вычисление текстурных характеристик зерновой культуры.

Модуль принятия решений – модуль, обеспечивающий основную задачу программного средства – произвести анализ изображения. Блок производит процесс выбора объектов, которые являются эталонными, с помощью пользователя. Далее производится принятие решение о принадлежности объекта к одному из кластеров с помощью классификатора, на основе признаков посчитанных в предыдущих блоках. Классификатором был выбран метод опорных векторов.

Модуль формирования отчетов – модуль, предоставляющий визуализацию принятого программным средством решения. Так же производится анализ качества классификации и построение ROC-кривой, которая является графическим представлением качества анализа.

В совокупности, все вышеописанные блоки позволяют решить поставленную задачу с наибольшей эффективностью и полнотой.

Обобщенная структурная схема программного средства изображена на рисунке 2.1.

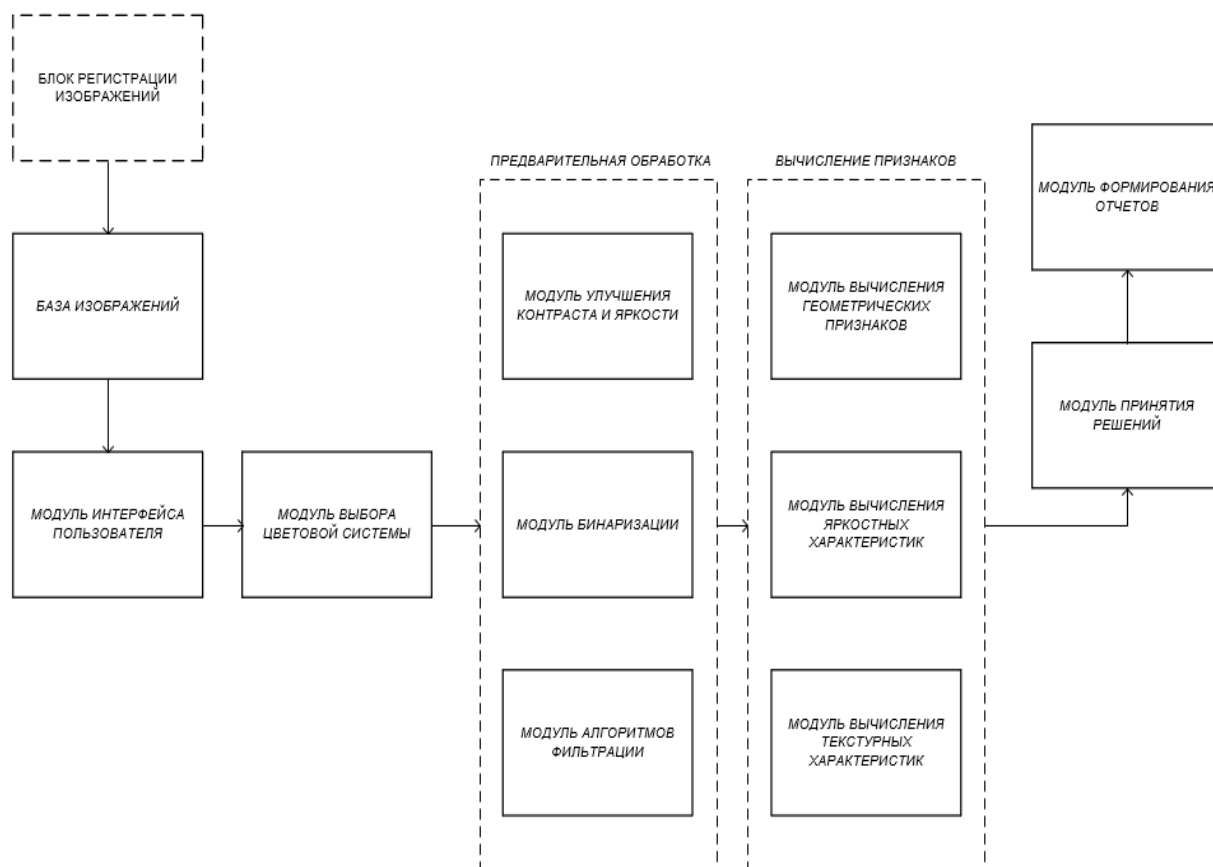


Рисунок 2.1 – Обобщенная структурная схема

## 2.2 Средства разработки

Выбор основных технологий является ключевым моментом при подготовке к созданию программного обеспечения, ведь правильный выбор позволит не только упростить процесс проектирования и разработки, но и облегчить эксплуатацию программного продукта.

При выборе языка программирования, на котором будет реализован программное средство, внимание было обращено на такие факторы, как удобство работы, целесообразность использования для решения поставленной задачи, наличия стандарта и документации, поддержка основных операционных систем. На основе этих признаков был выбран C++. Данный язык является компилируемым и статически типизированным. C++ поддерживает такие парадигмы программирования, как процедурное и объектно-ориентированное программирование. Как объектно-ориентированный язык, он оперирует понятиями наследование,

инкапсуляция и полиморфизм [12]. Так же обеспечивается модульность, раздельная компиляция, обработка исключений, абстракция данных, объявление типов объектов, виртуальные функции. К плюсам данного языка можно отнести наличие стандартной библиотеки, которая включает, в том числе, общеупотребительные контейнеры и алгоритмы. Так же C++ является в достаточной степени низкоуровневым, для объектно-ориентированных языков, это позволяет с помощью него решать задачи в которых требуется высокое быстродействие (обработка цифровых изображений и видео, сложные поисковые алгоритмы, сетевые технологии). Спектр задач для него очень широкий, что обеспечило ему огромную популярность среди программистов, на протяжении вот уже нескольких десятков лет [13].

Для разработки интерфейса была выбрана кроссплатформенная свободная IDE – Qt Creator. Она поставляется с фреймворком Qt. Qt Creator позволяет использовать все преимущества объектно-ориентированного подхода, позволяет осуществлять повторное использование кода с большей эффективностью, нежели при применении так называемого модульного программирования. Немаловажным также является использование высокопроизводительных компиляторов и стремление использовать скомпилированный код. Известно, что последний обладает существенно более высокой производительностью, чем код интерпретируемый. Так же Qt Creator обеспечивает визуальными средствами проектирования пользовательских интерфейсов, что позволяет ускорить работу над проектами, облегчить повторное использование кода и в определенной степени привлечь к созданию приложений начинающих программистов. Так же к достоинствам выбранной технологии можно отнести ее кроссплатформенность, это облегчает разработку программных средств, которые планируется использовать на разных операционных системах. В Qt Creator реализовано автодополнение, подсветка кода, задание стиля выравнивания, отступов и постановка скобок, все это облегчает разработку и снимает с программиста часть рутинной работы [14]. Логотип технологии представлен на рисунке 2.2.



Рисунок 2.2 – Логотип технологии Qt

В качестве сторонней библиотеки была выбрана библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом – OpenCV. Данное решение было принято после анализа имеющихся аналогов на предмет соответствия требованиям: высокое быстродействие, большое количество готовых решений алгоритмов, кроссплатформенность и доступная документация. OpenCV поддерживается для таких языков программирования, как C++, Python, Java, это делает библиотеку доступным средством разработки для большего количества программистов. Также библиотека имеет ускорение на уровне аппаратуры [15]. Логотип библиотеки представлен на рисунке 2.3.

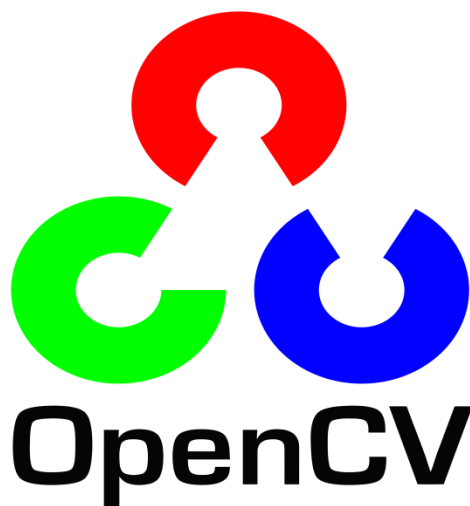


Рисунок 2.3 – Логотип библиотеки OpenCV.

Выбранные инструменты позволяют в полной мере и с высокой эффективностью решить все поставленные задачи данной дипломной работы.