1. **ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ**

Рассмотрим подробно функционирование программы. Для этого проведем анализ основных блоков программы и рассмотрим их зависимости. А также проанализируем все функциональные компоненты, которые входят в состав кода программы, и рассмотрим назначение всех методов и переменных этих блоков.

В разрабатываемом приложении можно выделить следующие блоки:

* блок преобразования видеозаписи в коллекцию изображений;
* блок пользовательского интерфейса;
* блок детектирования;
* блок классификации;
* блок взаимодействия с базой данных.

Изначально пользователь попадает на главный экран, логика которого находится в классе Form1.cs. Здесь расположен весь основной графический интерфейс. Здесь пользователь может указать на папку, в которой находятся видеозаписи, на которых необходимо произвести распознавание. После этого производится выбор места сохранения полученных изображений, которые будут сохранены во вложенные папки в соответствии с названиями исходных видеозаписей. Также при обработке исходных видеоматериалов есть возможность настройки частоты создания изображений. После этого пользователь выбирает набор необходимых преобразований и фильтров для улучшения качества распознавания. Здесь же есть производится выбор необходимых для распознавания групп дорожных знаков.

* 1. **Классы разрабатываемого программного средства**
     1. **Класс Form1**

В данном классе находятся все элементы, через которые пользователь взаимодействует с приложением. Здесь отображаются все информационные сообщения и выводятся управляющие элементы. Пользователь управляет программой в основном через нажатие на клавиши, расположенные в окне приложения. При нажатии происходит вызов необходимых функций, происходят определенные операции и пользователь получает какое-либо уведомление о завершении произошедшей операции.

* + 1. **Класс ffmpegConverter**

Этот класс является оберткой консольного вызова программы ffmpeg из одноименного набора библиотек для разбиения с определенной частотой видеозаписи на изображения.

Метод convert() создаёт новый процесс, в котором через консоль Windows вызывает программу ffmpeg c определенным набором аргументов. Через аргументы передается расположение папки видеозаписей и папки для сохранения полученных изображений.

* + 1. **Класс ImgOps**

Данный класс содержит методы для преобразования изображений с целью увеличения количественно-качественных характеристик распознавания дорожных знаков. Он использует методы библиотеки EmguCV.

Используя метод RGBtoGrey происходит преобразование изображения из цветного в градации серого. Это позволяет уменьшить количество цветовых каналов с 3 до 1, что позволяет использовать меньше вычислительных мощностей и уменьшает время распознавания.

В методе *InterpolationResize* производится масштабирование изображения с использованием бинарной интерполяции. Данный метод на входе получает исходное изображение и необходимые размеры результата и возвращает полученное изображение.

Суть интерполяции заключается в использовании имеющихся данных для получения ожидаемых значений в неизвестных точках. Интерполяция изображений работает в двух измерениях и пытается достичь наилучшего приближения в цвете и яркости пикселя, основываясь на значениях окружающих пикселей. Бикубическая интерполяция рассматривает массив из 4x4 окружающих пикселей — всего 16. Поскольку они находятся на разных расстояниях от неизвестного пикселя, ближайшие пиксели получают при расчёте больший вес. Бикубическая интерполяция производит значительно более резкие изображения, чем другие методы, и возможно, является оптимальной по соотношению времени обработки и качества на выходе.

Для улучшения различимости элементов классифицируемого фрагмента изображения содержащего дорожный знак необходимо применить метод *ContrastAlignment*. Метод принимает исходное изображение и возвращает полученное изображение. В данном методе используется контрастное выравнивание CLAHE (Contrast-limited adaptive histogram equalization).

Контрастное выравнивание CLAHE используется для изображений, имеющих неоднородное геометрическое распределения яркостей. Оно анализирует небольшие участки изображения и позволяет усилить локальный контраст. Для каждого пикселя рассматривается небольшая окрестность изображения, по которой строится функция преобразования, при этом все изображение, как таковое, не используется. Оно позволяет уменьшить неоднородность освещения дорожных знаков.

Метод RGBtoHSV

* + 1. **Класс SignsHaarCascade**
    2. **Класс CNN**
    3. **Класс OpenHaarCascadeFileDialog**
    4. **Класс OpenVideoFolderFileDialog**
    5. **Класс OpenPictureFolderFileDialog**
    6. **Класс OpenPictureFileDialog**