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 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 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

Метод RGBtoHSV