**1 ОБЗОР ЛИТЕРАТУРЫ**

## **Обзор существующих аналогов**

На этапе проектирования системы были тщательно изучены существующие аналоги. Среди обнаруженных аналогов преобладают программные средства, распознающие дорожные знаки с помощью видеорегистратора или смартфона. Они выводят на экран обнаруженные дорожные знаки и предупреждают пользователя о превышении скорости.

Наиболее приближенным является программное средство Roadly для ОС Android.

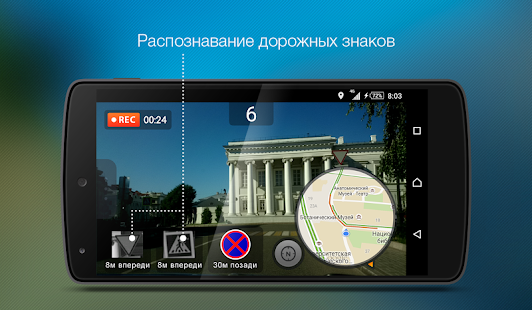


Рисунок. 1.1 Приложение Roadly [2]

Данное приложение является видеорегистратором с возможностью обнаружения на видео дорожных знаков, а также предупреждении пользователя об их присутствии на пути. Обнаруженные на устройстве знаки сохраняются в общую базу, откуда потом они загружаются к другим пользователей. Чем больше пользователей приложения, тем больше и актуальнее база. Приложение может работать в фоне как авторегистратор и выдавать предупреждения поверх любых навигационных приложений, например: Waze, Навител Навигатор (Navitel), Яндекс.Навигатор и др. [2].

В текущей версии приложение способно распознавать следующие дорожные знаки:

* Уступи дорогу
* Ограничения скорости (в момент превышения приложение

показывает сумму штрафа для большей мотивации снизить скорость.

* Обгон запрещен
* Остановка запрещена
* Стоянка запрещена
* Пешеходный переход
* Знаки, предписывающие проезд перекрестков
* Предупреждающие знаки

К минусам данного аналога можно отнести высокое энергопотребление, нестабильная работа и низкое качество видеозаписи в связи с высоким потреблением ресурсов мобильного устройства.

**1.2 Аналитический обзор**

Средство будет являться .NET Desktop приложением для OC Windows, не требующим подключения к сети Интернет при наличии на ПК пользователя всех необходимых библиотек (ffmpeg, EmguCV).

Разрабатываемое средство будет реализовано в виде приложения Windows Forms, что позволит предоставить пользователю удобный интерфейс. Это графические приложения, которые легко развертывать и обновлять, которые могут работать, при подключении к сети Интернет или без него, и могут получать доступ к ресурсам на локальном компьютере более безопасным способом, чем традиционные приложения на основе Windows. В Windows Forms форма – это визуальный компонент, на которой выводится информация для пользователя. Обычно приложение Windows Forms строится путем помещения элементов управления на форму и написания кода для реагирования на действия пользователя, такие как щелчки мыши или нажатия клавиш. Элемент управления – это отдельный элемент пользовательского интерфейса, предназначенный для отображения или ввода данных [3].

FFmpeg – бесплатный набор библиотек (лицензии GPL 2.0 и LGPL 2.1) который можно использовать в своих проектах, как в коммерческих, так и в свободно-распространяемых. С помощью программы ffmpeg, которую можно запускать из командной строки Windows с необходимыми аргументами, можно конвертировать видео из одного формата в другой. Склеивать несколько видеороликов в один. Разбирать видео на отдельные кадры (изображения) с определённой частотой, а потом склеивать обратно. Ускорять-замедлять, масштабировать, добавлять субтитры и несколько звуковых дорожек [4]. Список доступных возможностей приведен на рисунке 1.2.

Emgu CV – это кроссплатформенная оболочка .Net для библиотеки обработки изображений OpenCV.

OpenCV (англ. Open Source Computer Vision Library, библиотека компьютерного зрения с открытым исходным кодом) – библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом. Реализована на C/C++, также разрабатывается для Python, Java, Ruby и других языков. Может свободно использоваться в академических и коммерческих целях – распространяется в условиях лицензии BSD. Фактически, OpenCV – это набор типов данных, функций и классов для обработки изображений алгоритмами компьютерного зрения.

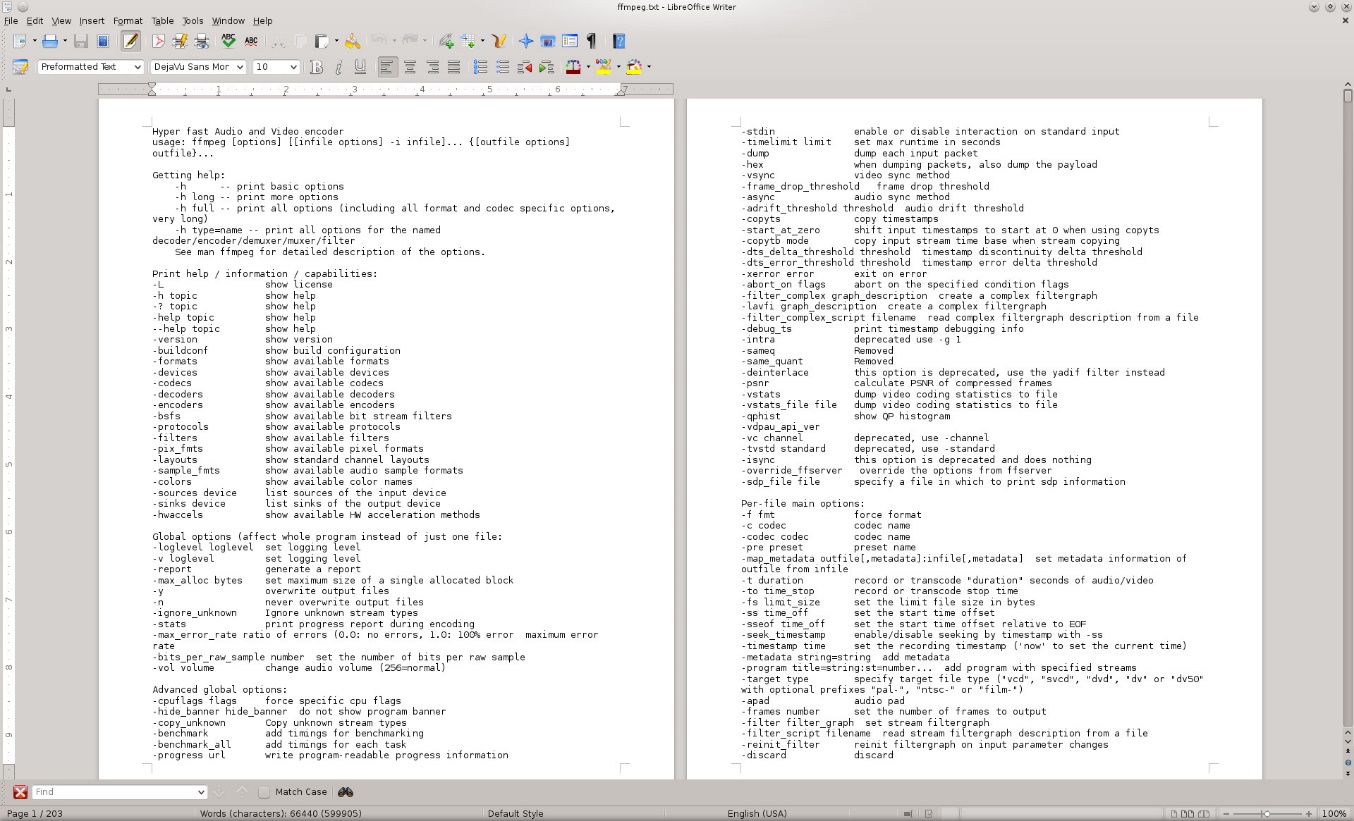


Рисунок. 1.2 - Список возможностей команды ffmpeg [3]

Основной идеей при разработке .NET Framework является обеспечение разработчику удобство и гибкость разработки за счёт предоставления ему возможности создавать приложения различных типов, способные запускаться на различных типах устройств и в различных средах. Вторым принципом стала ориентация на семейство систем Microsoft Windows. Программа для .NET Framework, написанная на любом поддерживаемом языке программирования, сначала переводится компилятором в единый для .NET промежуточный байт-код Common Intermediate Language (CIL). Затем код либо исполняется виртуальной машиной Common Language Runtime (CLR), либо транслируется в исполняемый код для конкретного целевого процессора. Использование виртуальной машины предпочтительно, так как избавляет разработчиков от необходимости заботиться об особенностях аппаратной части. В случае использования виртуальной машины CLR встроенный в неё JIT-компилятор преобразует промежуточный байт-код в машинные коды нужного процессора. Современная технология динамической компиляции позволяет достигнуть высокого уровня быстродействия. Виртуальная машина CLR также сама заботится о базовой безопасности, управлении памятью и системе исключений, избавляя разработчика от части работы. Архитектура .NET Framework описана и опубликована в спецификации Common Language Infrastructure (CLI), разработанной Microsoft и утверждённой ISO и ECMA[5].

* 1. **Событийно-ориентированное программирование**

Важным вопросом при разработке приложения является его архитектура. В данном проекте при разработке пользовательского интерфейса был сделан выбор в сторону событийно-ориентированного программирования (СОП) как наиболее подходящего для поставленных задач. СОП можно также определить как способ построения компьютерной программы, при котором в коде явным образом выделяется главный цикл приложения, тело которого состоит из двух частей: выборки события и обработки события.

**1.4 Постановка задачи**

Разрабатываемое программное средство должно отвечать следующим требованиям:

1. Иметь удобный и интуитивно-понятный интерфейс, который в полной мере дает пользователю контроль над работой приложения.
2. В полной мере выполнять необходимый функционал, не допуская ошибок, приводящих к остановке работы приложения.

**1.5 Анализ источников**

Исчерпывающие инструкции по созданию приложения Windows Forms и существующие возможности подобных приложений описаны в [6]. Так же там показаны примеры использования доступных элементов управления и их возможности.

Существующие алгоритмы детектирования и распознавания дорожных знаков, такие как каскады Хаара и нейронные сети, их достигнутая точность и сравнительная характеристика были рассмотрены в [7].

Для использования ffmpeg изучена документация и инструкции по использованию, которые подробно расписаны в [8].

В [9] есть вся необходимая информация для разработки программ с помощью библиотеки EmguCV. Здесь содержится описание всех классов, методов и полей в данной библиотеке.