**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.

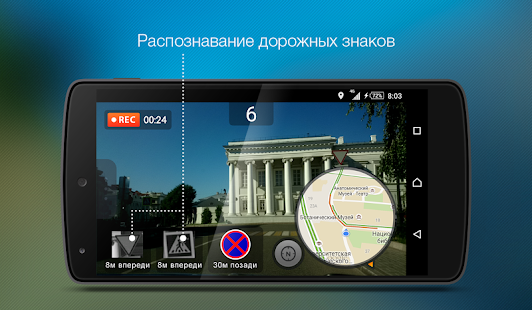


Рис. 1.1 Приложение Roadly [1]

Данное приложение является видеорегистратором с возможностью распознавания на видео дорожных знаков и предупреждении пользователя об их присутствии на пути. А распознанные на устройстве знаки отправляются в общую базу, откуда потом они попадают к другим водителям. Чем больше пользователей приложения, тем больше и актуальнее база. Roadly может работать в фоне как авторегистратор и при этом выдавать предупреждения поверх любых навигационных приложений, например: Waze, Навител Навигатор (Navitel), Яндекс.Навигатор, Sygic, iGO и др. [1].

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. Актуально только для РФ).

* 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 графически насыщенные приложения, которые легко развертывать и обновлять, которые могут работать, когда они подключены к Интернету или отключены от него, и могут получать доступ к ресурсам на локальном компьютере более безопасным способом, чем традиционные приложения на основе Windows. В Windows Forms форма — это визуальная поверхность, 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 [2].

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 параметрами, можно конвертировать видео из одного формата в другой. Склеивать несколько видеороликов в один. Разбирать видео на отдельные кадры (изображения), а потом склеивать обратно. Ускорять/замедлять, масштабировать, добавлять субтитры и несколько звуковых дорожек [3].

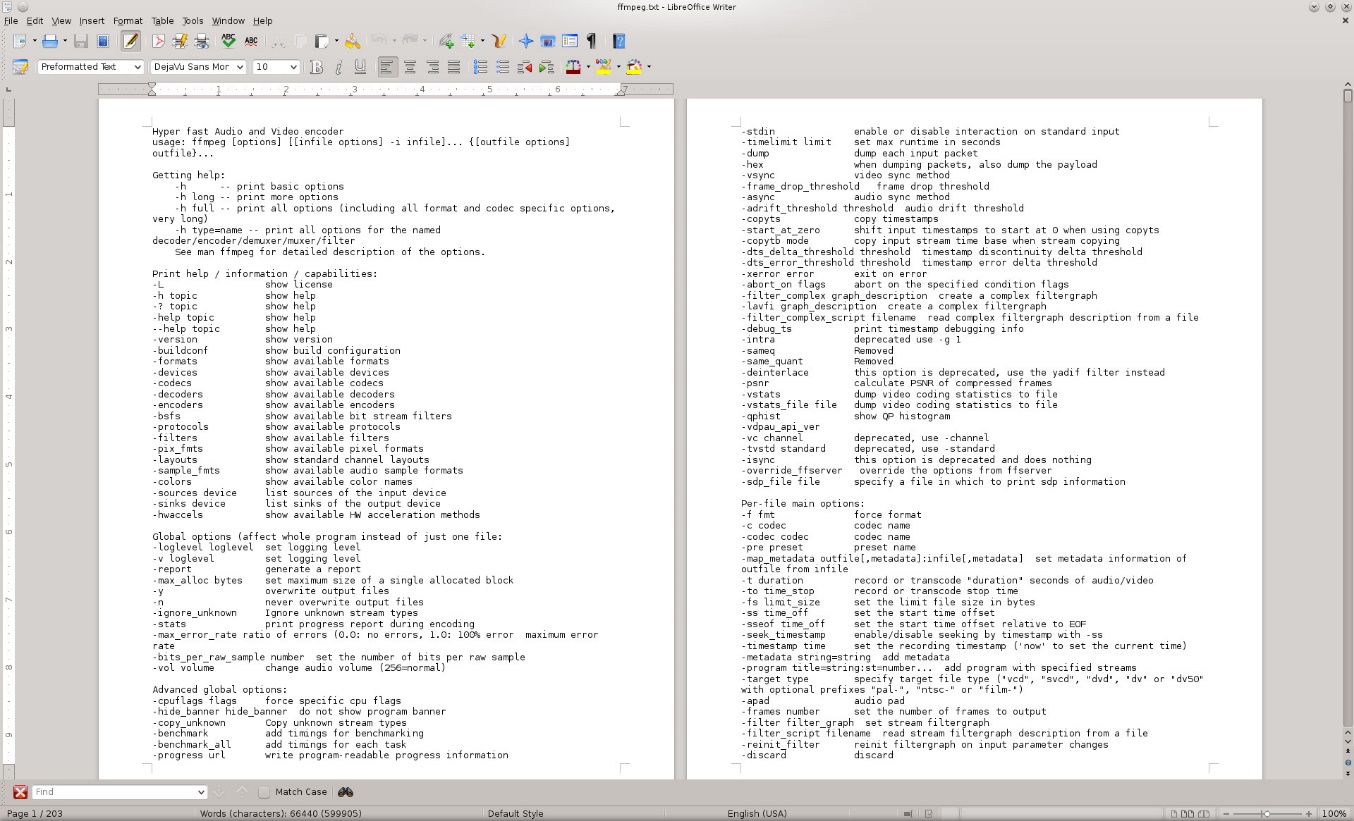


Рис 1.2 Список возможностей команды ffmpeg [3]

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, Matlab, Lua 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.

Основной идеей при разработке .NET Framework является обеспечение свободы разработчика за счёт предоставления ему возможности создавать приложения различных типов, способные выполняться на различных типах устройств и в различных средах. Вторым принципом стала ориентация на системы, работающие под управлением семейства операционных 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) (ранее назывался Microsoft Intermediate Language, MSIL). В терминах .NET получается сборка, англ. assembly. Затем код либо исполняется виртуальной машиной Common Language Runtime (CLR), либо транслируется утилитой NGen.exe 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 «на лету» (just in time) 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. В CLI описаны типы данных .NET, формат метаданных о структуре программы, система исполнения байт-кода и многое другое. Объектные классы .NET, доступные для всех поддерживаемых языков программирования, содержатся в библиотеке Framework Class Library (FCL)[4].

* 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**

В [5] даны исчерпывающие инструкции по созданию приложения Windows Forms и описаны существующие возможности подобных приложений. Показаны примеры использования доступных элементов управления и их возможности.

В [6] описаны существующие алгоритмы детектирования и распознавания дорожных знаков, их достигнутая точность и сравнительная характеристика.

В [7] находится документация и инструкции по использованию

набора библиотек ffmpeg для обработки видеозаписей.

В [8] есть вся необходимая информация для разработки программ с помощью библиотеки EmguCV. Здесь содержится описание всех классов, методов и полей в данной библиотеке.