**2 СИСТЕМНОЕ ПРОЕКТИРОВАНИЕ**

Изучив теоретические аспекты разрабатываемого средства и выработав список требований необходимых для разработки, разбиваем программу на функциональные блоки (модули).

В разрабатываемом приложении можно выделить следующие блоки:

* блок пользовательского интерфейса;
* блок разделения видеозаписей на кадры;
* блок обработки изображения;
* блок детектирования;
* блок классификации;
* блок базы данных.

Структурная схема, иллюстрирующая перечисленные блоки и связи между ними приведена на чертеже ГУИР.400201.067 C1.

Рассмотрим функциональные блоки приложения.

*Блок пользовательского интерфейса* представляет собой совокупность средств, при помощи которых пользователь взаимодействует с приложением. Для построения интерфейса используется технология Windows Forms - интерфейс программирования приложений, отвечающий за графический интерфейс и представление данных, получение и обработку ввода пользователя и представляет собой событийно-ориентированное приложение, поддерживаемое Microsoft .NET Framework. Здесь пользователь взаимодействует с программой.

*Блок разделения на кадры* выполняет задачу кадрирования полученных видеозаписей для последующей работы с отдельными изображениями. Он является обёрткой ffmpeg - набора свободных библиотек с открытым исходным кодом, которые позволяют записывать, конвертировать и передавать цифровые аудио- и видеозаписи в различных форматах. В данной программе из набора библиотек используется только одноименный модуль ffmpeg - утилита командной строки для конвертирования видеофайла из одного формата в другой.

*Блок обработки изображения* необходим для улучшения количественных и качественных характеристик модулей детектирования и классификации. Здесь происходит преобразования изображений в наиболее удобные для работы форматы и цветовые форматы. Функционал основан на использовании библиотеки EmguCV.

*Блок детектирования* работает на основе алгоритма каскадов Хаара, используя Виолы-Джонса для нахождения знаков на изображении. В методе Виолы-Джонса основу составляют примитивы Хаара, представляющие собой разбивку заданной прямоугольной области на наборы разнотипных прямоугольных подобластей. В данном блоке используется класс CascadeClassifier библиотеки EmguCV. Для работы алгоритма необходимы файлы обученных каскадов. Для обучения необходимо не менее 1000 положительных и столько же отрицательных примеров и около суток для обучения. Результат обучения хранится в xml файле, который и используется данным блоком.

*Блок классификации* представляет собой нейронную сеть.

*Блок базы данных* содержит программный код для стандартизации и сохранения полученных результатов в базу данных.