1. **Разработка программных модулей**
   1. **Преобразование видеозаписей в коллекцию изображений и координат**

Для преобразования базы видеозаписей к единому формату удобному для дальнейшей обработки используется класс FFMPEGConverter. Здесь происходит разбиения видеозаписи на кадры, получения субтитров, содержащих записи координат и времени видеосъемки.

* + 1. **Разбиение видеозаписи на кадры**

Для того чтобы преобразовать видеозапись в папку с изображениями используется функция ConvertVidToImages из класса FFMPEGConverter. Функция выполняет следующие шаги:

1. Создается переменная флага результата операции;
2. Если видеозапись по заданному существует выполняется шаг 3, если не существует – флаг результата устанавливается в нулевое значение;
3. Для каждой видеозаписи создается директория, в которую будут сохранены полученные изображения;
4. Высчитывается частота преобразования, равная отношению одной секунды в миллисекундах к числу отсчетов;
5. Создается переменная, содержащая параметры запуска утилиты ffmpeg, в которую записываются путь к видеозаписи, частота преобразования, путь к папке, в которую необходимо сохранить полученные изображения;
6. Создается и инициализируется переменная процесса для преобразования;
7. Устанавливаются параметры процесса: название запускаемого приложения(ffmpeg), аргументы запуска, режим с отображением консоли на экран для контролирования стадии преобразования;
8. Процесс запускается;
9. Программа ожидает завершения работы процесса;
10. Переменная флага результата устанавливается в единичное значение;
11. Возврат флага результата.
    * 1. **Получение субтитров из видеозаписи**

Для получения субтитров, содержащих географические координаты и время записи, используется функция ConvertVidToSubs из класса FFMPEGConverter. Функция выполняет следующие шаги:

1. Создается переменная, в которой хранится путь к текстовому файлу, в котором будут находится полученные субтитры;
2. Создается переменная, содержащая параметры запуска утилиты ffmpeg, в которую записываются путь к видеозаписи, частота преобразования, путь к папке, в которую необходимо сохранить полученные изображения;
3. Создается и инициализируется переменная процесса для преобразования;
4. Устанавливаются параметры процесса: название запускаемого приложения(ffmpeg), аргументы запуска, режим с отображением консоли на экран для контролирования стадии преобразования;
5. Процесс запускается;
6. Программа ожидает завершения работы процесса;
7. Возврат строки, содержащей путь к файлу с субтитрами.
   * 1. **Считывание файла с субтитрами и его преобразование к единому виду**

После получения субтитров из видеозаписи необходимо преобразовать записи из текстового файла к формату, пригодному для дальнейшей обработки. Для этого используется функция ParseSubtitleFile из класса FFMPEGConverter. Работа функции состоит из следующих шагов:

1. Происходит проверка текстового файла по введенному пути на предмет его существования.
2. Создается переменная строчного типа, в которую происходит считывание всего текстового файла;
3. Создается регулярное выражение для разбиения всей строки на отдельные записи, соответствующие каждому кадру видеозаписи;
4. С помощью регулярного выражения из строки получается коллекция записей;
5. Создается список, в котором будут находится географические координаты распознаваемых знаков;
6. Далее в цикле с помощью функции ParseMovementPoint этого же класса производится преобразование каждой полученной записи в объект класса MovementPoint, который хранит географические координаты;
7. Каждый полученный объект класса MovementPoint добавляется в список координат;
8. Возврат заполненного списка координат.
   * 1. **Функция создания объекта, содержащего географические координаты**

Для преобразования части строки, которая содержит географические координаты к объекту вызывается функция ParseMovementPoint. Данная функция работает следующим образом:

1. Сначала часть текстового файла разбивается на отдельные записи, которые в файле разделены между собой запятыми;
2. После этого для каждой записи производится преобразование даты в удобный для восприятия формат;
3. Затем производится преобразование географической широты в десятичный формат с помощью вызова функции ConvertDegreesAndDecimalMinutesStringToDecimal, логика работы которой описана ниже;
4. То же самое выполняется и для переменной, отражающей географическую долготу;
5. И в конце полученный результат преобразуется в объект класса movementPoint;
6. Возврат полученного объекта из функции.
   * 1. **Функция преобразования географических координат к десятичному формату**

Для того, чтобы привести координаты к единому десятичному формату, удобному для применения, используется функция ConvertDegreesAndDecimalMinutesStringToDecimal. Работает функция следующим образом:

1. Изначально координаты хранятся в дробном виде, разделенном точкой;
2. Вычисляем целые градусы;
3. Затем вычисляем целые минуты;
4. После этого приводим всё к целому количеству градусов, учитывая, что в одном градусе шестьдесят минут.
5. Возвращаем полученное значение.
   1. **Поиск** **дорожных знаков с помощью каскадов Хаара**

После получения необходимых изображений и координат для проведения операции детекции используются методы класса DetectFolder и SignsHaarCascade.

Методы класса DetectFolder позволяют производить поиск дорожных знаков на совокупности фотографий, найденных в определенной директории.

Метод DetectAll класса DetectFolder работает следующим образом:

1. Создается коллекция, содержащая каскады, с помощью которых будет происходить поиск знаков;
2. Создается список, в котором будут находится пути ко всем папкам с изображениями, которые были найдены в введенной директории;
3. Происходит заполнение коллекции каскадов, путем создания объектов класса SignsHaarCascade;
4. В цикле каждое изображение приводится к единому формату Mat;
5. Для каждого каскада из списка вызывается метод DetectAll класса SignsHaarCascade, работа которого описана ниже;
6. При нахождении хотя бы одного дорожного знака, по введенному для сохранения полученных результатов пути производится создание директории с соответствующим названию исходного изображения названием;
7. Производится сохранение по заданному пути результатов поиска;
8. Возврат из функции.
   * 1. **Поиск** **дорожных знаков с помощью каскадов Хаара**

Поиск знаков на определенном изображении с помощью определенного каскада выполняется с помощью метода DetectAll.Работа метода DetectAll класса SignsHaarCascade происходит по следующему алгоритму:

1. Создается список, в котором будут находится части изображений, содержащих дорожные знаки;
2. С помощью каскада и вызова функции DetectMultiScale из библиотеки EmguCV производится поиск регионов на изображении, которые содержат дорожные знаки;
3. Для каждого найденного региона создается изображение и добавляется в список;
4. Функция возвращает полученный список.
   1. **Изменение размеров изображений и преобразование цветового пространства в градации серого**

Для того, чтобы изображения можно было классифицировать с помощью нейронной сети их необходимо привести единому виду:

* Размер 32 на 32 пикселя;
* Единственный цветовой канал (оттенки серого).

Для данных целей используются методы класса ImgOps.

* 1. **Нейронная сеть**

Классификация подготовленных изображений происходит с помощью сверточной нейронной сети, программная реализация которой находится в классе CNN.

* 1. **Сохранение результатов**
  2. **Запись результатов в базу данных**