Оглавление

[Введение 2](#_Toc484548109)

[1.Постановка задачи 4](#_Toc484548110)

[1.1 Автоматизация распознавания регистрационных номерных знаков автомобилей 4](#_Toc484548111)

[1.2 Задачи распознавания 4](#_Toc484548112)

[Выводы 4](#_Toc484548113)

[2.Существующие решения для определения автомобильных номеров 5](#_Toc484548114)

[2.1 Общая архитектура аппаратного комплекса распознавания 6](#_Toc484548115)

[2.2 Программные продукты 7](#_Toc484548116)

[2.2 1АВТО-Инспектор 7](#_Toc484548117)

[2.2 2SL-Traffic 8](#_Toc484548118)

[Выводы 9](#_Toc484548119)

[3.Системные требования 10](#_Toc484548120)

[4.Архитектура решений 10](#_Toc484548121)

[4.1 Алгоритмы распознавания объектов на изображении 10](#_Toc484548122)

[4.1.1 Бинаризация по порогу 10](#_Toc484548123)

[4.1.2 Фильтрация контуров 11](#_Toc484548124)

[4.1.3 Каскад Хаара 12](#_Toc484548125)

[4.1 K-means …………………………………………………………….12](#_Toc484548126)

[4.2 Способы распознавания символов на изображении 13](#_Toc484548127)

[4.2.1 Общая архитектура 13](#_Toc484548128)

[4.2.2 Алгоритмы нормализации 14](#_Toc484548129)

[4.2.3 Алгоритмы сегментации 16](#_Toc484548130)

[4.2.4 Алгоритмы распознавания 17](#_Toc484548131)

[4.3 Особенности документации библиотеки компьютерного зрения OpenCV 18](#_Toc484548132)

[5. Проектная реализация 20](#_Toc484548133)

[6.Тестирование 20](#_Toc484548134)

[Заключение 20](#_Toc484548135)

[Список использованных источников 21](#_Toc484548136)

# Введение

В настоящее при помощи компьютерных технологий автоматизируется широкий круг дел и процессов, которые в недалеком прошлом возлагались только на человека. Информационные технологии используются повсюду: в образовании, в промышленности, в транспорте и т.п. Программисты всего мира, много работают и разрабатывают совершенно новые и совершенствуют уже существующие решении для автоматизации. Решение сложности фиксирования транспортное средство по регистрационному номерному знаку на данный момент является важным аспектом безопасности, учета и контроля. Использовать такой продукт можно в различных сферах, которые касаются автотранспорта, допустим необходимость идентификация и учета автомобилей. Так же примером могут служить автотранспортная промышленность, контролирование скорости движения, гаражи, автомобильные стоянки, контроль въезда на территорию предприятия и т.п. В настоящее время существует не так много систем определения номерных знаков, и многие из них не являются по настоящему качественной продукцией, по последнему испытанию самый лучший результат это 90 процентов распознавания номеров. Однако, параллельно с созданием алгоритмов, разрабатываются аппаратные средства, которые будут заниматься распознаванием. Системы и комплексы, обладающие высокой точностью и быстрым распознаванием, как правило, очень дороги, либо требуют тщательного изучения инструкции, на которую иногда времени нет. Высокая цена и сложность существующих систем не позволяют осуществить их массовое внедрение в нашей стране. Задачу распознавания и фиксацию автомобиля можно разделить на три подзадачи: нахождение номерного знака на машине, локализация номерной пластины, т.е. самого номера и распознавание символов на нем. Данная работа посвящена разработке и реализации алгоритма распознавания номерного знака для программного аппаратного комплекса для пропускного пункта. В общем случае распознавание реализуется в четыре этапа: предварительная обработка изображения, сегментация, распознавание символов, запись номера в хранилище данных. Предварительная обработка изображения заключается в обработке и в выделении номерного знака на полученном изображении различными фильтрами с целью улучшения его качества. На этапе сегментации выделяются буквы и цифры знака, которые затем распознаются каким- либо методом, далее уже распознанный объект заносится в выбранный источник данных.

# Постановка задачи

* 1. Автоматизация распознавания регистрационных номерных знаков автомобилей

Автоматизация распознавания номеров автомобилей является одной из часто встречаемых задач для корпоративных систем безопасности. Данные системы позволяют существенно повысить распознавание машин и сформировать отчетность по многим аспектам. Есть проблемы, с которыми можно столкнуться при разработке данной системы, такие как:

* Нахождение расположения номера на машине
* Локализация номера автомобиля
* Сегментация символов номерного знака автомобиля
* Распознавание каждого символа в номере автомобиля
  1. Задачи распознавания
* Нахождения номерного знака на машине
* Локализация знака доступными фильтрами для распознавания
* Сегментация и обработка символов
* Запись данных в БД для дальнейшего анализа

Выводы

Проанализировав вышеперечисленные проблемы и задачи, можно сделать вывод, что для их решения рекомендуется применение современных информационных технологий. Начнем с того, что пропускные пункты нуждаются в автоматизированной системе распознавания. На основе существующих решений и библиотек разработки подобных систем, к данному аппаратному комплексу были сформулированы требования, представленные в техническом задании.

# Существующие решения для определения автомобильных номеров

Существует 20 видов Российских автомобильных номеров, которые содержат как большие, так и маленькие цифры, и буквы, а также их фон бывает позитивным, так и негативными. В зависимости от применяемых регистрационных знаков транспортные средства подразделяют на несколько групп. В данной работе будут рассмотрены номерные знаки первой группы, так как они встречаются чаще всего.

**Транспортные средства 1-й группы**

Транспортные средства первой группы, принадлежат юридическим лицам и гражданам Российской Федерации, и так же гражданам и юридическим лицам иностранных государств, кроме отнесенных к группе 3, а также лицам без гражданства (рисунок 1):



*Рисунок 1 − Типы регистрационных знаков 1-й группы*

* 1 − для легковых, грузовых, грузопассажирских автомобилей и автобусов (кроме отнесенных к типу 1Б), Формат: 3 буквы, 3 цифры, код региона;
* 1А − для легковых автомобилей должностных лиц по перечню, определяемому, Формат: 3 буквы, 3 цифры, ;
* 1Б − для легковых автомобилей, используемых для перевозки пассажиров на коммерческой основе, автобусов, грузовых автомобилей, оборудованных для перевозок более восьми человек. 2 буквы, 3 цифры, код региона;
* 2 − для автомобильных прицепов (включая задние прицепы к мотоциклам и мотороллерам) и полуприцепов; 5 3 − для тракторов, самоходных дорожно-строительных и иных машин и прицепов (полуприцепов) к ним. Формат: 2 буквы, 4 цифры, код региона;
* 4 − для мотоциклов, мотороллеров, мопедов и мотонарт. Формат: 2 буквы, 4 цифры, код региона.

## Общая архитектура аппаратного комплекса распознавания

Система считывания автомобильных номеров состоит нескольких аппаратных и программных модулей (рисунок 2):

* + видеокамера;
  + плата видеозахвата;
  + модуль нахождения номера;
  + модуль обработки изображения номера
  + модуль распознавания;
  + внешняя или внутренняя БД
  + модуль обработки информации



*Рисунок 2 − Общая архитектура*

Изображение с видеокамеры поступает на вход модуля нахождения. Программный детектор находит на изображении движущийся автомобиль. Затем алгоритмом определяется положение или нахождение номерного знака, т.е. область где знак находиться. После этого выделенный номер обрабатывается программой оптического распознавания. Далее идет база данных, которая в зависимости от поставленных перед системой задач, может иметь различную схему, которая обеспечивает хранение данных, потом уже обрабатывается информация из бд для:

* Проверки номера на факт угона
* Анализ проезжих машин и т.п.

Основными задачам являются: захват номера, даты и времени появления автомобиля в угле обзора камеры; запись стоп-кадра автомобиля в нужное время или отрезка видео с машиной; проверка номера на регистрации в бд и т.д.

## Программные продукты

### АВТО-Инспектор

«АВТО-Инспектор» − специальный аппаратно-программный модуль для регистрации и распознавания автомобильных номеров компании «СТБ-Сервис» действует следующим образом. Над контролируемой полосой движения устанавливается ТВ-камера. Камера подключена к системе «АВТО-Инспектор», программное обеспечение которого обнаруживает появление автомобиля в кадре, отбирает кадр с оптимальным размером и четкостью автомобильного номера и распознает номер автомобиля из кадра В базе данных сохранятся этот стоп-кадр и выводится строка с номеров (рисунок 3):



*Рисунок 3− Стоп-кадр*

По данными c сайта Адрон [9]

К преимуществам его внедрения можно отнести:

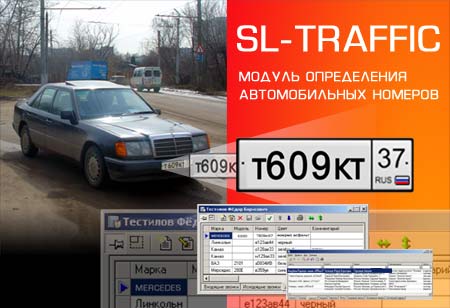
* + Вероятность безусловного распознавания составил 90%;
  + Вероятность ошибки составил 1.03%;
  + Модуль успешно работает в любых погодных условиях;
  + Создание сетевых систем на территориях большой протяженности;
  + Подключение к модулю светофора, шлагбаума, автоматических ворот.

Недостатками данного решения являются:

* + Некорректное распознавание региона номера;
  + Кроме системы нужно покупать аппаратуру к нему;
  + Системные требования не предусматривают работы системы на операционные системы Windows;
  + Нет возможности внедрения в информационный сервис
  + Нет возможности использования отдельных модулей.

### SL-Traffic

SL-Traffic − программный модуль считывания и распознавания автомобильных номеров в режиме реального времени компании СпецЛаб. Модуль работает в качестве видео фильтра в системе GOALcity или может встраивается в любую систему видеорегистрации, в том числе и сторонних фирм (рисунок 4).



*Рисунок 4− Стоп-кадр*

К преимуществам его внедрения можно отнести:

* + Четыре вида алгоритмов распознавания номера;
  + Регулирование скорости распознавания;
  + Работа как с собственным архивом, так и с другими базами данных;
  + Одновременное распознавание до 20 номеров на одном кадре;

Недостатками данного решения являются:

* + Вероятность ошибочного распознавания 8.30%
  + Алгоритм распознавания выбирается самостоятельно
  + Минимальная высота номера на кадре - 25 пикселей
  + Нет возможности внедрения в информационный сервис

## Выводы

Исходя из анализа преимуществ и недостатков систем, стало ясно, что большинство решений не имеет доступа к информационным сервисам и нет возможности управления модулями отдельно, так же многие системы не могут распознать регион машины, поэтому было решено создать систему подобного класса. Так же данные системы стоят от 1500$ до 3000$. Заявленная точность желает лучшего. При испытании демо-версий надежно распознаются лишь чистые номера высокой контрастности и относительно большого разрешения. В итоге на заявленную точность 90-98%, приходится реальная − 80-87%. Используемые алгоритмы локализации и распознавания номерных знаков не доступны, только лишь некоторые компании дают доступ к ним и то за деньги. Для распознавания обычно используют нейроно-сететевые и шаблонные алгоритмы. Весомым ограничивающим фактором на цель использования системы является максимальная скорость автомобиля, при которой программа способна локализовать и распознать номер на движущемся транспорте. На что влияют, во-первых, способ установки камеры − высота и наклон; а во-вторых быстродействие обработки изображения автомобиля

# Системные требования

## Требуемая аппаратная часть

Компьютер

|  |  |
| --- | --- |
| ОС | Windows 7/8/8.1/10, Ubuntu 16.10 |
| Процессор | Intel Celeron J1900 |
| Память | 2 ГБ |
| Видеокарта | Встроенная в процессор |
| Требования к соединению с Интернетом | скорость в 64 кбит/с или выше(для отправки данных в SAP) |
| Место на жестком диске | 20 гб |

Камера

|  |  |
| --- | --- |
| Матрица | 0.3 млн пикс |
| Разрешение | 640x480 |
| Частота кадров | 30 Гц |
| Подключение | USB 2.0 |
| Фокусировка | автоматическая |
| ОС | Windows 7/8/8.1/10, Ubuntu 16.10 |

## Требование к номерному знаку автомобиля

* Номерной знак должен быть чистым
* Отклонение номера по горизонтальной оси от -30 до 30
* Формат номера типа 1 транспортных средств первой группы

# Архитектура решений

## Алгоритмы распознавания объектов на изображении

### Бинаризация по порогу

Самое просто преобразование — это бинаризация изображения по порогу (image thresholding). Для RGB изображения и изображения в градациях серого порогом является значение цвета. Встречаются идеальные задачи, в которых такого преобразования достаточно. Предположим, нужно автоматически выделить фигуры на картинке с черным фоном (рисунок 4):



*Рисунок 5 - фигуры на черном фоне*

Выбор порога, по которому происходит бинаризация, во многом определяет процесс самой бинаризации. В данном случае, изображение было бинаризовано по среднему цвету (рисунок 5):



*Рисунок 6− Бинаризовtyанный рисунок по среднему цвету*

Обычно бинаризация осуществляется с помощью алгоритма, который адаптивно выбирает порог. Таким алгоритмом может быть выбор матожидания или моды.

### Фильтрация контуров

Отдельный класс фильтров — фильтрация границ и контуров. Контуры очень полезны, когда мы хотим перейти от работы с изображением к работе с объектами на этом изображении. Когда объект достаточно сложный, но хорошо выделяемый, то зачастую единственным способом работы с ним является выделение его контуров. Существует целый ряд алгоритмов, решающих задачу фильтрации контуров:

* Оператор Кэнни
* Оператор Собеля
* Оператор Лапласа
* Оператор Прюитт
* Оператор Робертса

Чаще всего используется именно Кэнни, который хорошо работает (рисунок 6):



*Рисунок 7 – использование оператора Кэнни*

### Каскад Хаара

Каскад Хаара используется в методе Виолы—Джонса. Этот алгоритм, позволяет обнаруживать объекты на изображениях в реальном времени. Хотя алгоритм может распознавать различные классы изображений, основной задачей при его создании было обнаружение лиц. Основное преимущество детектора Хаара: скорость. Благодаря быстрой обработки изображения, можно с лёгкостью обрабатывать потоковое видео. Детектор Хаара используется для распознавания большинства классов объектов. К ним относятся лица и другие части тела людей, номера автомобилей, пешеходы, дорожные знаки, животные и.т.д.

Принцип данного способа состоит в том, что классификатор формируется на примитивах Хаара путём расчёта значений признаков. Для обучения на вход классификатора сначала подаётся набор «правильных» изображений с предварительно выделенной областью на изображении, дальше происходит перебор примитивов и расчёт значения признака. Вычисленные значения сохраняются в файле в формате xml.

### K-means

Наиболее простой, но в то же время достаточно неточный метод кластеризации в классической реализации. Он разбивает множество элементов векторного пространства на заранее известное число кластеров k. Действие алгоритма таково, что он стремится минимизировать среднеквадратичное отклонение на точках каждого кластера. Основная идея заключается в том, что на каждой итерации пере вычисляется центр масс для каждого кластера, полученного на предыдущем шаге, затем векторы разбиваются на кластеры вновь в соответствии с тем, какой из новых центров оказался ближе по выбранной метрике. Алгоритм завершается, когда на какой-то итерации не происходит изменения кластеров.

Проблемы алгоритма k-means:

* необходимо заранее знать количество кластеров. Мной было предложено метод определения количества кластеров, который основывался на нахождении кластеров, распределенных по некоему закону (в моем случае все сводилось к нормальному закону). После этого выполнялся классический алгоритм k-means, который давал более точные результаты.
* алгоритм очень чувствителен к выбору начальных центров кластеров. Классический вариант подразумевает случайный выбор кластеров, что очень часто являлось источником погрешности. Как вариант решения, необходимо проводить исследования объекта для более точного определения центров начальных кластеров. В моем случае на начальном этапе предлагается принимать в качестве центов самые отдаленные точки кластеров.
* не справляется с задачей, когда объект принадлежит к разным кластерам в равной степени или не принадлежит ни одному.

## Способы распознавания символов на изображении

### Общая архитектура

Алгоритм распознавания номерного знака состоит из следующих этапов (рисунок 7):

1. Нормализация
2. Сегментация.
3. Распознавание.



*Рисунок 8 – этапы распознавания*

Процедура нормализации используется практически всегда после получения информации, и представляет собой применение операций усреднения и выравнивания гистограмм, различного типа фильтров для исключения помех, а также подавления внешних шумов. Под сегментацией понимается процесс разделения изображения на отдельные символы. Конечный этап обработки - распознавание. Для этого этапа входными данными являются изображения, полученные в результате шумоподавления и процесса сегментации. Распознаванию символов с помощью готовых программ, либо с помощью алгоритмов.

### Алгоритмы нормализации

Так как обнаружение номера происходит не совсем точно и требует дальнейшего уточнения его положения, а также улучшение качества снимка. Например, в случае, изображённом на рисунке 8, требуется поворот и обрезка краёв:

image

*Рисунок 9.- Область содержащая номерной знак выделенная для анализа*

Для начала необходимо стабилизировать изображение по горизонтали. Это делается путем выделения с помощью алгоритма Хафа всех длинных горизонтальных прямых, которые являются границами номера. Классический алгоритм преобразования Хафа связан с идентификацией прямых в изображении, что позволит использовать его в нашей работе. Результат обработки изображения представлен на рисунке 9:

image

*Рисунок 10.- Стабилизированный номер по горизонтали по алгоритму Хафа*

Следующим шагом по обработке номера будет увеличение резкости и контраста. Пример обработки представлен на рисунке 10.



*Рисунок 11.- Номерной знак без фильтра(слева) и номерной знак, в котором увеличен контраст и резкость(справа)*

Помимо описанного способа, можно выделять границы при помощи градиентных фильтров первого и второго порядка. Применим фильтр второго порядка – LoG-фильтр (Marr-Hildreth), который работает путем сверки изображения с лапласианом функции Гаусса. Он сочетает в себе обнаружение границ со сглаживанием. На рисунке 11 представлен результат работы данного фильтра с использованием маски фильтра G:



*Рисунок 12. - Номерной знак без фильтра(слева) и номерной знак обработанный LoG-фильтром(справа)*

При обработке изображения имеются определенные трудности, такие как:

1. Низкая эффективность распознавания при загрязнении номера.
2. Горизонтальные границы номера не всегда являются хорошим ориентиром для разбиения него по сегментам.
3. Автомобильные номера различных стран имеют разные шрифты и разный формат.
4. Автомобильные номера могут быть установлены в места, не предусмотренные конструкцией транспортного средства, либо отсутствовать.

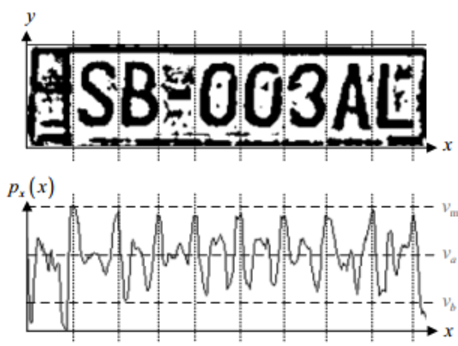
### Алгоритмы сегментации

Если связанная область (контур) имеет высоту в пикселях от H1 до H2 а ширина и высота связана отношением от K1 до K2, то оставляем в кадре и отмечаем, что в этой области может быть знак. Почти наверняка на этом этапе останутся лишь цифры и буквы, остальной мусор из кадра уйдет. Возьмем ограничивающие контуры прямоугольники, приведем их к одному масштабу и дальше поработаем с каждой буквой/цифрой отдельно (рисунок 12).



*Рисунок 13. – сегментация по алгоритму контуров*

Так же можно провести операцию бинаризации, т. е. использовать какое-то правило разделения пикселей на два класса. При разделении номера на символы эта операция вовсе не обязательна. После этого достаточно найти максимумы горизонтальной диаграммы, это и будут промежутки буквами. Особенно если мы ожидаем определенное количество знаков и расстояние между знаками будет примерно одинаковое, то разбиение на буквы по гистограмме сработает отлично (рисунок 13).



*Рисунок 14. – сегментация с помощью максимумов*

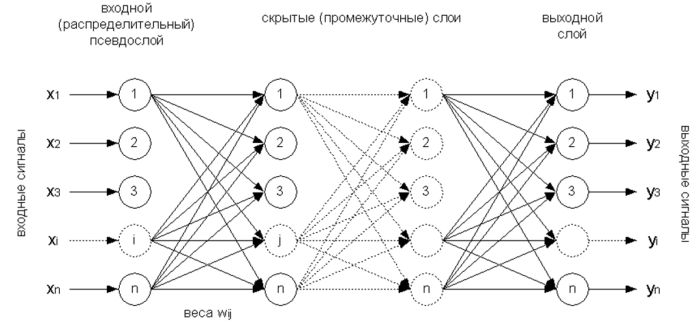
При значительном загрязнении номера периодические максимумы при разбиении на символы могут просто не проявиться, хотя сами символы могут быть визуально вполне читаемы.

Горизонтальные граница номера — не всегда хороший ориентир. Номера могут быть изогнуты штатно (Mercedes C-класса), могут быть бережно утоплены в неподходящее почти квадратное углубление для номера на американских машинах. А верхняя граница заднего номера просто часто прикрыта элементами кузова. Так же иногда из-за не правильной обработки могут появится лишние элементы, которые не являются символами.

### Алгоритмы распознавания

Для решения задачи распознавания символов используются нейронные сети. Например, самоорганизующиеся нейронные сети Кохонена, обеспечивающие топологическое упорядочивание входного пространства образов. Они позволяют типологически непрерывно отображать входное n-мерное пространство в выходное m-мерное.

Ключевым аспектом нейронной сети является ее обучение, которое сводится к определению связей между нейронами и установлению силы этих связей (весовых коэффициентов). Алгоритмы обучения нейронной сети упрощенно сводятся к определению зависимости весового коэффициента связи двух нейронов от числа примеров, подтверждающих эту зависимость. Наиболее распространенным алгоритмом обучения нейронной сети является алгоритм обратного распространения ошибки, представленный на рисунке 14:



*Рисунок 15 – Многослойная нейронная сеть с обратным распространением ошибки.*

Сущность алгоритма обратного распространения ошибки сводится к заданию произвольных небольших начальных значений весов связей нейронов. Затем с помощью рекурсивного алгоритма, начиная с выходных узлов по направлению к первому скрытому слою для всех обучающих пар «значения входных признаков – значение выходного признака» (примеров из обучающей выборки) вычисляется выход сети (Y). Данный процесс продолжается до тех пор, пока не будет достигнут минимальный уровень ошибки

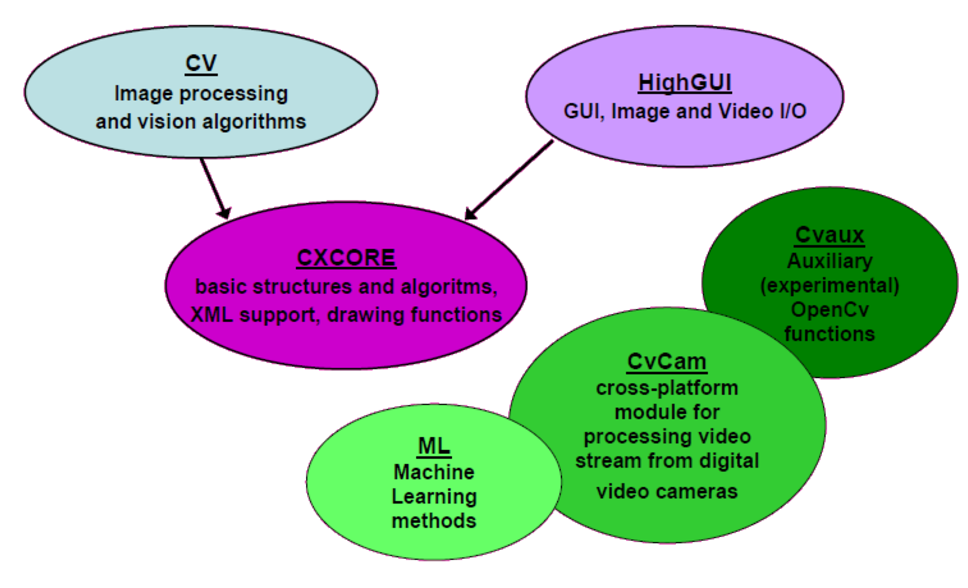
Так же можно использовать открытое программное обеспечение, выполняющее автоматическое распознавание как единичной буквы, так и сразу текста, как Tessaract ORC. Эта программа удобна тем, что есть для любых ОС, стабильно работает и легко обучаем. Но он очень плохо работает с замыленным, битым, грязным и деформированным текстом. Когда я попробовал сделать на нём распознавание номеров – от силы лишь 20-30% номеров из базы распознались правильно. Самые чистые и прямые. Хотя, конечно, и при использовании готовых библиотек, стоит обучать их на нужные шрифты.

## Особенности документации библиотеки компьютерного зрения OpenCV

OpenCV — 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. Данная библиотека может предоставить реализацию алгоритмов, которые мы перечислили в предыдущих разделах. Фактически, OpenCV – это набор типов данных, функций и классов для обработки изображений алгоритмами компьютерного зрения.

Основные модули библиотеки (рисунок 15):

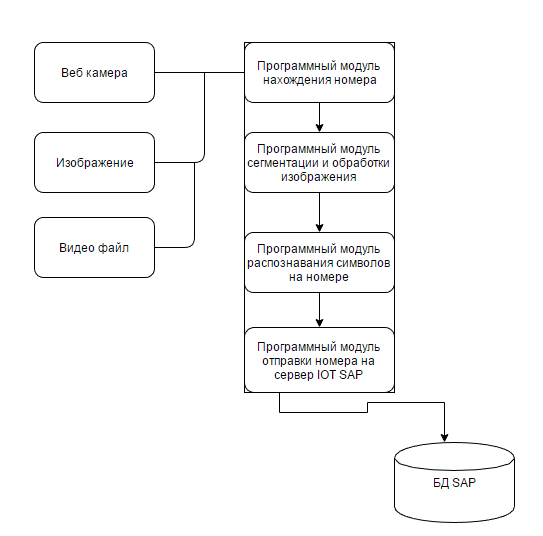
1. cxcore — ядро содержит базовые структуры данных и алгоритмы:
2. Базовые операции над многомерными числовыми массивами
3. Матричная алгебра, математические функции, генераторы случайных чисел
4. Запись/восстановление структур данных в/из XML
5. Базовые функции 2D графики
6. CV — модуль обработки изображений и компьютерного зрения:
7. Базовые операции над изображениями (фильтрация, геометрические преобразования, преобразование цветовых пространств и т. д.)
8. Анализ изображений (выбор отличительных признаков, морфология, поиск контуров, гистограммы)
9. Анализ движения, слежение за объектами
10. Обнаружение объектов, в частности лиц
11. Калибровка камер, элементы восстановления пространственной структуры
12. Highgui — модуль для ввода/вывода изображений и видео, создания пользовательского интерфейса:
13. захват видео с камер и из видео файлов, чтение/запись статических изображений.
14. функции для организации простого UI (все демо приложения используют HighGUI)
15. Cvaux — экспериментальные и устаревшие функции:
16. пространств. зрение: стерео калибрация само калибрация
17. поиск стерео-соответствия, клики в графах
18. нахождение и описание черт лица
19. CvCam — захват видео:
20. позволяет осуществлять захват с видео с цифровых видеокамер (поддержка прекращена и в последних версиях этот модуль отсутствует)



*Рисунок 16– схема основных модулей*

# Проектная реализация

Разработанная система считывания автомобильных номеров состоит нескольких программных модулей (рисунок 17)



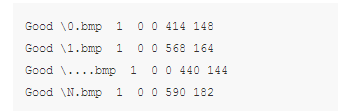
*Рисунок 17– схема основных модулей программно-аппаратного комплекса*

## Программный модуль нахождения номера

Ранее уже был рассмотрен алгоритм нахождения номера с помощью каскада Хаара, в данном разделе будут описаны этапы обучения классификатора для модуля нахождения.

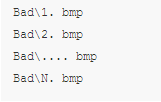
Создание классификатора состоят из нескольких этапов:

* Создание папки Good с выборкой из 500 картинок, на которых есть машина и номер, так же в ней должен быть создан файл, названный так же как директория, т.е. «Good.bat», в нем данные должны быть записаны как относительный пути к изображениям и так же нужно указать положение рассматриваемого объекта и его размер, рисунок 18:



*Рисунок 18– структура файла Good.dat*

* Создание папки Bad с выборкой из 600 картинок, на который есть все что угодно, но нет машин, так же в ней должен лежать файл, названный так же как директория, т.е. «Bad.dat», в нем данные должны быть записаны как относительные пути к изображениям, рисунок 19:



*Рисунок 19– структура файла Bad.dat*

* Преобразование положительных картинок к общему размеру с помощью программы opencv\_createsamples.exe
* Подсчет итогового каскада с помощью программы opencv\_traincascade.exe

После выполнения данных этапов, мы создали файл в формате xml, под названием haarcascade\_russian\_plate\_number.xml структура, которого представлена на рисунке 20:

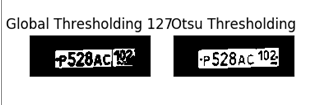


*Рисунок 20– структура файла haarcascade\_russian\_plate\_number.xml*

## Программный модуль сегментации и обработки изображения

После получения локализированного изображения номера, нужно его нормализовать, для дальнейшей обработки. Для этого сначала выполняется бинаризация по алгоритму Отсу, который позволяет разделить пиксели двух классов («полезные» и «фоновые»), рассчитывая такой порог, чтобы внутриклассовая дисперсия была минимальной.

Экспериментальным способом стало ясно, что использование алгоритма Отсу увеличивает распознавание символов на 21.52 процента, чем если бы мы использовали постоянную константу 127 для порога, на рисунке 21 показано сравнение двух алгоритмов:



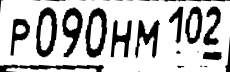
*Рисунок 21–результаты разных методов бинаризации (слева порог 127, справа порог меняется по алгоритму Отсу)*

После бинаризации идет этап обработки. Сначала нужно повернуть номер, чтобы тот был расположен горизонтально. Для этого на номере были найдены все горизонтальные линии и относительно их рассчитывались углы отклонения. После этого нужно рассчитать среднее арифметическое углов , и поворачивать уже данный угол(рисунок 22).



*Рисунок 22–результат алгоритма поворота на нужный угол (слева до поворота, справа после поворота)*

После стабилизированные номера в горизонтальное положение, обрезается картинка по алгоритму контуров. На картинке находиться все элементы изображения, которые можно поместить в контур, вырезаем по самому первому контуру, который найдется на изображении, обычно это контур номерного знака, а не его символ, работа алгоритма представлена на рисунке 23:



*Рисунок 23–результат алгоритма контуров*

На этом этапе обработка изображения заканчивается, и можно начать сегментацию символов на нем.

Алгоритм сегментации был описан в пункте 4.2.3, но чтобы в буквах О , P, цифрах 9,0,8 и т.п. не выделялось ничего лишнего, нужно было проверять не пересекается ли данные цифры и буквы с какой-нибудь областью, которая внутри их. Если данной операции не делать, то сегментация выделяет лишние элементы, как показано на рисунке 24:



*Рисунок 24–результат алгоритма без удаления ненужных областей*

При использовании алгоритма удаления областей получается такой результат, рисунок 25:



*Рисунок 25–результат алгоритма c удалением ненужных областей*

В результате данного модуля, мы получает список символов, которые потом будут распознаваться. Данный модуль имеет свои ошибки и иногда сегментация выделает области, которые убрать данными алгоритмами не получиться (рисунок 26)



*Рисунок 26–результат алгоритма сегментации*

Или же при плохом освещении и грязи, алгоритм бинаризации преобразует картинку так, что символы будут немного с друг другом пересекаться и отдельно символ не получиться выделить, либо он просто не выделится, рисунок 27 и 28:



*Рисунок 27–результат алгоритма сегментации*



*Рисунок 28–результат алгоритма сегментации без удаления областей внутри*

## Программный модуль распознавания символов на номере

## Программный модуль отправки номера на сервер IOT SAP

# тестирование

# Заключение

В данной работы была рассмотрена реализация и возможность применения системы распознавания автомобильных номеров. Так же было рассмотрено, что по сравнение с уже с существующими системами эта система предполагает более высокую точность распознавания за счет применения нейросетевых алгоритмов.

# Список использованных источников

1. ГОСТ 50577-93. Знаки государственные регистрационные транспортных средств типы и основные размеры. Технические требования. Введен 01.01.94. – М.: Изда- тельство стандартов. – 33 стр.
2. Арлазаров В. Л., Троянкер В.В., Котович Н.В. Адаптивное распознавание символов. [Электронный ресурс]. – <http://www.ocrai.narod.ru/adaptive.html>.
3. Лукошенко Г. Н. Распознавание скелетных образов. [Электронный ресурс]. – <http://www.ocrai.narod.ru/skeletrecognize.html>
4. Саймон Хайкин. Нейронные сети. Полный курс. – М.: Вильямс, 2005. – 1104 стр.
5. OpenCV шаг за шагом. Введение. [Электронный ресурс]. – <http://robocraft.ru/blog/computervision/264.html>
6. Распознавание номеров: от А до Я [Электронный ресурс]. — Режим доступа:https://habrahabr.ru/company/recognitor/blog/221891/.html (дата обращения: 12.11.2016).
7. Прикладное использование алгоритма распознавания государственных автомобильных номеров /Силаева А.Э., Юрчев В.А.//Cовременные тенденции развития науки и технологий. –2017.—№103.- c.150
8. Программно-аппаратный комплекс распознавания регистрационных знаков автотранспортных средств /Силаева А.Э., Тетерин Д.М., Миронов А.Н.// Сборник трудов 7-ой международной конференции “ИТ – Стандарт 2016”. -2016. -.№7- с.630
9. РАСПОЗНАВАНИЕ АВТОМОБИЛЬНЫХ НОМЕРОВ: «АВТО-ИНСПЕКТОР» — ЛИДЕР ТЕСТИРОВАНИЯ [Электронный ресурс]. — Режим доступа: http://www.adron-perm.ru/test\_leader/(дата обращения: 06.06.2017).