

Содержание

Введение.....	2
1. Техническое задание.....	3
1.1. Назначение разработки и область применения.....	3
1.2. Технические требования	3
2. Анализ технического задания	4
2.1. Выбор операционной системы	4
2.2. Выбор языка программирования.....	6
2.3. Выбор среды разработки	7
2.4. Обзор существующих алгоритмов поиска номерных пластин	8
3. Разработка структуры системы поиска номерных пластин.....	11
3.1. Разработка общей структуры системы	11
3.2. Разработка алгоритма предварительной обработки данных	16
3.3. Разработка алгоритма формирования системы признаков.....	17
3.4. Разработка алгоритма принятия решения	18
4. Разработка программных средств	20
4.1. Разработка интерфейса пользователя системы.....	20
4.2. Программная реализация модулей системы	22
5. Тестирование системы.....	25
5.1. Описание набора данных	25
5.2. Описание методики тестирования.....	32
5.3. Результаты вычислительного эксперимента	34
Заключение	46
Список литературы	47

					ВКР-НГТУ-09.03.01-(13-В-2)-009-17			
Изм.	Лист	№ докум.	Подпись	Дата	Программная система поиска номерной пластины автотранспортного средства ни изображении	Лит.	Лист	Листов
Разраб.		Хапилов Е.М.						
Провер.		Гай В.Е.					1	47
Реценз.								
Н. Контр.								
Утверд.								

Введение

На сегодняшний день системы распознавания номерных пластин транспортных средств получают все большее распространение. Различные компании в целях сокращения расходов, а также повышения безопасности и автоматизации охранных систем. Такие системы могут быть использованы в составе средств, имеющих более расширенный функционал, для решения различных задач, как например: контроль проезда, управление доступом и учет автотранспорта на парковках, КПП и проходных, весовых, автомойках, автомагистралях, а также на территориях ТСЖ и частных домов.

					ВКР-НГТУ-09.03.01-(13-В-2)-009-17	Лист
						2
Изм.	Лист	№ докум.	Подпись	Дата		

1. Техническое задание

1.1. Назначение разработки и область применения

Разрабатываемая система предназначена для поиска номерной пластины автотранспортного средства на изображении. Изображение транспортного средства с номерным знаком загружается в программу поиска номерной пластины, после чего по заданным алгоритмам осуществляется поиск её местоположения на заданном изображении.

Области применения:

- 1) Автоматизация систем контроля транспортных средств на пунктах охраны;
- 2) Автоматизация системы учета автотранспортных средств на предприятии;

1.2. Технические требования

Рассмотрим требования, предъявляемые разрабатываемой системой к ЭВМ:

- 1) Операционная система Microsoft Windows не ниже версии XP;
- 2) Требования к аппаратному обеспечению определяются операционной системой;
- 3) мышь;

					ВКР-НГТУ-09.03.01-(13-В-2)-009-17	Лист
						3
Изм.	Лист	№ докум.	Подпись	Дата		

2. Анализ технического задания

2.1. Выбор операционной системы

Начиная разработку в первую очередь нужно выбрать операционную систему, на которую будут установлены программные средства необходимые для разработки программного продукта.

Беря во внимание опыт использования и ориентацию программного продукта, то выбор остается между ОС Linux и Windows [7].

Остановимся на этих операционных системах более подробно:

1) На данный момент Microsoft Windows – это целое семейство операционных систем, которое можно считать самым распространенным. Windows изначально была надстройкой для MS – DOS и стала так популярна благодаря удобному и дружелюбному графическому интерфейсу. На данный момент основой Windows стал Windows NT. Windows имеет множество версий, в зависимости от цели использования (Домашние, серверные, мобильные решения). Самой распространенной системой от Microsoft является Windows 7, но это ненадолго, так как рынок уверенно занимается Windows 10.

2) Linux – это название Unix – подобных ОС, которые основаны на ядре Unix. Linux имеет открытый исходный код, а также Linux создается и распространяется как свободное и открытое ПО. Основное преимущество UNIX систем от остальных ОС в том, что Unix системы изначально являются многозадачными, т.е. в одно и то же время множество людей могут выполнять множество задач. В Unix системах сразу может работать много людей, при этом они могут выполнять различные вычислительные процессы, которые будут использовать исключительно ресурсы этого компьютера. Второе отличие и несомненное преимущество Unix – это её мультиплатформенность. Ядро Unix написано таким образом, что его можно приспособить почти под любой микропроцессор, т.е. обеспечивается его переносимость.

Основные различия, преимущества и недостатки операционных систем Windows, Linux:

1) Защита

Почти полное отсутствие на платформе Linux, по крайней мере, на сегодняшний день, вредоносных программ, что позволяет избежать расходов на Антивирусное ПО, а также устранение последствий работы вредоносного ПО.

2) Поддерживаемое оборудование

Linux обладает заметно худшей поддержкой компьютерного оборудования, чем Windows, такого, как: принтеры, USB – устройства, сканеры и т.п. Поэтому перед приобретением оборудования необходимо узнать, имеется ли поддержка ОС Linux. На данный момент проблема драйверов на Linux постепенно уходит, пропорционально тому, как Linux получает все большее распространение.

					ВКР-НГТУ-09.03.01-(13-В-2)-009-17	Лист
Изм.	Лист	№ докум.	Подпись	Дата		4

3) Программное обеспечение

Основной недостаток Linux – это значительно меньшее количество ПО, чем для платформы Windows. Более того, если речь идет о каких то известных программных продуктах, которые присутствуют для платформы Windows, то на Linux их может и не быть. Поэтому зачастую приходится находить менее известные аналоги, которые зачастую уступают лидерам.

4) Применение и разработка программного обеспечения

Главной платформой для разработки ПО является платформа Windows. Большая часть программ пишется на таких языках, как Java, C++, C# и т.д. Среди них есть и мультиплатформенные языки, которые позволяют разрабатывать ПО сразу под все платформы, например Java, C++.

Для разработки системы поиска номерной пластины мной была выбрана ОС Windows, так как она отвечает всем предъявляемым требованиям и имеется ранее полученный опыт разработки на данной платформе.

					ВКР-НГТУ-09.03.01-(13-В-2)-009-17	Лист
						5
Изм.	Лист	№ докум.	Подпись	Дата		

2.2. Выбор языка программирования

Важным этапом в разработке ПО – это выбор языка программирования.

На данный момент присутствует обилие языков программирования на ОС Windows, такие, как: Java, C++, C#, JavaScript, Python, R, и т.д.

Для написания дружественного графического интерфейса ввиду знания и простоты написания был выбран Java [9].

Java – этот объектно – ориентированный язык программирования. Java программы транслируются в байт – код, который выполняется виртуальной машиной Java (JVM).

Одним из достоинств такого выполнения программы – это независимость байт – кода от ОС и оборудования, что позволяет выполнять приложения на любом устройстве, для которого существует JVM. Ещё одним достоинством Java является система безопасности, в рамках которой выполнение программы контролируется JVM. При попытке выполнения операций, которые превышают заданные полномочия, вызывается исключение.

Для написания блока обработки графики будет использоваться язык R, в котором реализованы все необходимые для этого функции [5].

R – это язык программирования, который предназначен для статической обработки данных и работы с графикой, обладает хорошей расширяемостью с помощью пакетов. Пакеты представляют из себя библиотеки, предназначенные для специальных областей применения. В базовую поставку включен базовый набор библиотек, а на 2013 год доступно более 4000 пакетов.

					ВКР-НГТУ-09.03.01-(13-В-2)-009-17	Лист
						6
Изм.	Лист	№ докум.	Подпись	Дата		

2.3. Выбор среды разработки

Для Java разработки изначально предоставляется несколько IDE, такие, как Eclipse, IntelliJ IDEA, NetBeans. Каждая оболочка имеет свои особенности, которые делают её уникальной. На данный момент самой распространенной интегрированной средой разработки можно назвать Eclipse, который был выбран для создания графической оболочки системы. Eclipse в первую очередь служит для создания расширений, этим он завоевал популярность: разработчик может расширить Eclipse любым модулем [10]. Уже существуют Java Development Tools (JDT), C/C++ Development Tools (CDT), COBOL, FORTRAN, PHP, X10 (X10DT) и т.д. Расширения дополняют среду Eclipse модулями для работы с базами данных, серверами приложений и т.д. Eclipse написан на Java, поэтому кросс платформенным продуктом.

R использует интерфейс командной строки, однако существуют и графические интерфейсы, такие, как: R Commander, RKWard, RStudio, Weka, Rapid Miner, KNIME .

Для разработки основного блока обработки графики был выбран Rstudio, которые по праву можно считать одним из самых распространенных сред разработки. RStudio – это среда разработки с открытым исходным кодом. RStudio написана на языке C++ и использует фреймворк Qt для графического создания пользователя.

RStudio распространяется в двух версиях:

- 1) RStudio Desktop: программа выполняется непосредственно на локальной машине;
- 2) RStudio Server: программа выполняется в RStudio на удаленном Linux – сервере, доступ к которому осуществляется через браузера;

2.4. Обзор существующих алгоритмов поиска номерных пластин

Существующие системы используют различные алгоритмы для поиска и распознавания номерных пластин на изображении, которые имеют различную сложность реализации. Однако структура алгоритма в большинстве своей одна.

Структура алгоритма:

1) Предварительный поиск номерной пластины на изображении – сужение границ поиска до небольшой области, в которой содержится номерная пластина.

2) Определение границ номера, нормализация контрастности.

Данная структура является базовой. В некоторых случаях она может немного отличаться от базовой [2].

2.4.1 Анализ границ и контуров номерной пластины

Самый просто и очевидный способ – это поиск прямоугольной области – контура номерной пластины. Данный метод применим только в ситуациях с номерной пластиной, у которой имеется четкая, ничем не загороженная и достаточно ровная граница. Изображение фильтруется для поиска границ, далее выделяются и анализируются найденные границы. При использовании данного метода очень велик шанс получения ошибки, в связи с тем, что большинство номеров не имеют идеально выраженного контура.

2.4.2 Анализ части границ номерной пластины

Существует подход, где от рамки номерной пластины анализируется только её часть (см. Рис.2.1). Особенностью данного метода является то, что выделяются контуры и ищутся все вертикальные прямые. Для двух прямых, расположенных близко друг к другу, на небольшом расстоянии по оси Y, рассматривается гипотеза о том, что номерная пластина расположена между этими прямыми.

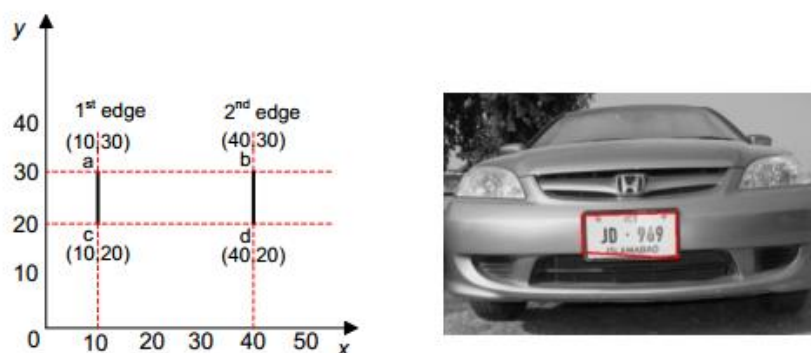


Рис. 2.1 Анализ части границ номерной пластины

2.4.3 Гистограммный анализ регионов

Очень распространенным является метод анализа гистограмм изображения (Рис.2.2). Метод основан на том, что частотная характеристика региона с номерной пластиной отличная от частотной характеристики окрестности номерной пластины. На изображении выделяются границы,

далее выстраивается проекция изображения на ось Y или на ось X. Полученный максимум проекции изображения может совпасть с местом расположения номера.

Данный метод имеет существенный минус – транспортное средство должно быть сопоставимо с размером кадра, т.к. фон изображения может содержать различные детализированные объекты.

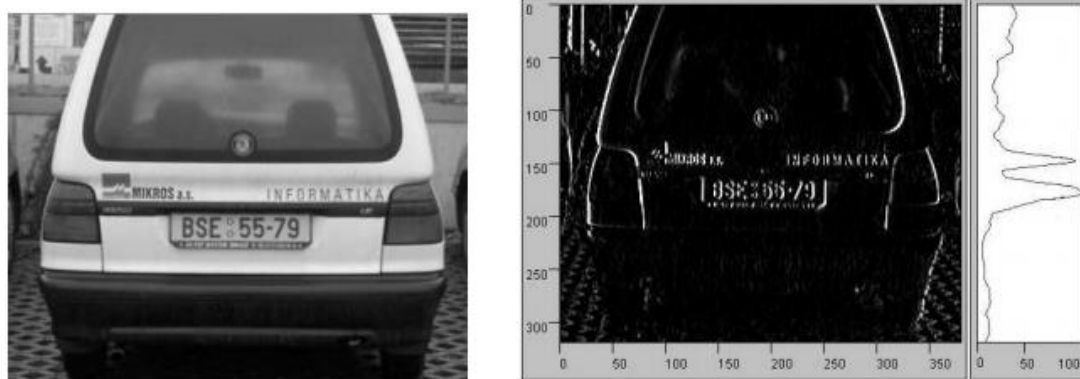


Рис. 2.2 Гистограммный анализ регионов

На данный момент существует множество платных и бесплатных программных систем поиска и распознавания номерных пластин. Бесплатные системы зачастую не уступают платным по функционалу, надежности и скорости работы.

По причине доступности бесплатных программ они обязательны к рассмотрению в рамках данной работы.

2.4.5 «Java ANPR»

Проект Java ANPR относится к области машинного зрения и нейронных сетей при построении системы автоматического поиска и распознавания номерных пластин (ANPR) [1]. Этот проект представляет из себя систему, состоящую из математических принципов и алгоритмов, с помощью которых обеспечивается процесс поиска, нормализации, сегментации символов и распознавания номерных знаков.

Библиотека Java ANPR имеет несколько несомненных преимуществ:

- 1) Кроссплатформенность.
- 2) Все алгоритмы написаны на Java, без использования нативных библиотек.
- 3) Возможность использования под OS Android с небольшими доработками.
- 4) Высокая скорость распознавания (0.2 – 0.8 сек).

Однако Java ANPR плохо работает с зашумленными изображениями, или, с изображениями, сделанными при плохой освещенности.

2.4.6 «Opos»

Написан на C#. Для поиска и распознавания номерной пластины автомобиля используется библиотека компьютерного зрения opencv, распознавание символов же происходит силами библиотеки OCR CuneiForm – Puma.NET. Эта библиотека имеет серьезный недостаток – перед передачей

изображения в Pima.NET оно сначала сохраняется на диск, что является критичным и затруднительным при использовании этой библиотеки в системах реального времени.

2.4.7 «Automatic License Plate Recognition»

Вторая библиотека, написанная на C#. В её основе используются две библиотеки – opencv на C# – Emgu, использующаяся для поиска номерной пластины и tessnet, которая используется для распознавания знаков автомобильного номера. Преимуществами этой библиотеки является возможность работы с кириллическими символами. Недостатком же является затруднительная работа с системами реального времени.

					ВКР-НГТУ-09.03.01-(13-В-2)-009-17	Лист
						10
Изм.	Лист	№ докум.	Подпись	Дата		

3. Разработка структуры системы поиска номерных пластин

3.1. Разработка общей структуры системы

В качестве основы для получения решения выпускной квалификационной работы было решено использовать теорию активного восприятия изображения. Задача распознавания заключается в построении алгоритма, который позволит отнести заданный на системе признаков объект, к одному из классов, который удовлетворяет требованиям сходства. В данном случае требованием сходства является максимум сходства внутриклассовых объектов, при максимуме различий межклассовых объектов [3].

Теория активного восприятия изображения подразумевает реализацию двух этапов системы распознавания образов: разработка алгоритма предварительной обработки данных и формирование системы признаков (рис. 3.1).

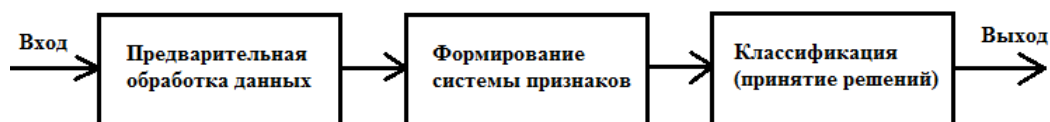


Рис. 3.1. Структура системы распознавания образов.

Алгоритм предварительной обработки данных, с позиции теории активного восприятия изображения, понимается, как операция интегрирования, а формирование системы признаков понимается, как операция дифференцирования.

Результатом операции интегрирования является матрица «визуальных масс», данное преобразование определено, как Q – преобразование.

Результатом операции дифференцирования является вектор «спектральных коэффициентов» $\mu = (\mu_0, \mu_1, \mu_2, \dots, \mu_{15})$.

Операция дифференцирования реализуется по средствам 16 фильтров (см. рис.3.2), где темный элемент «-1», светлый элемент – «+1».

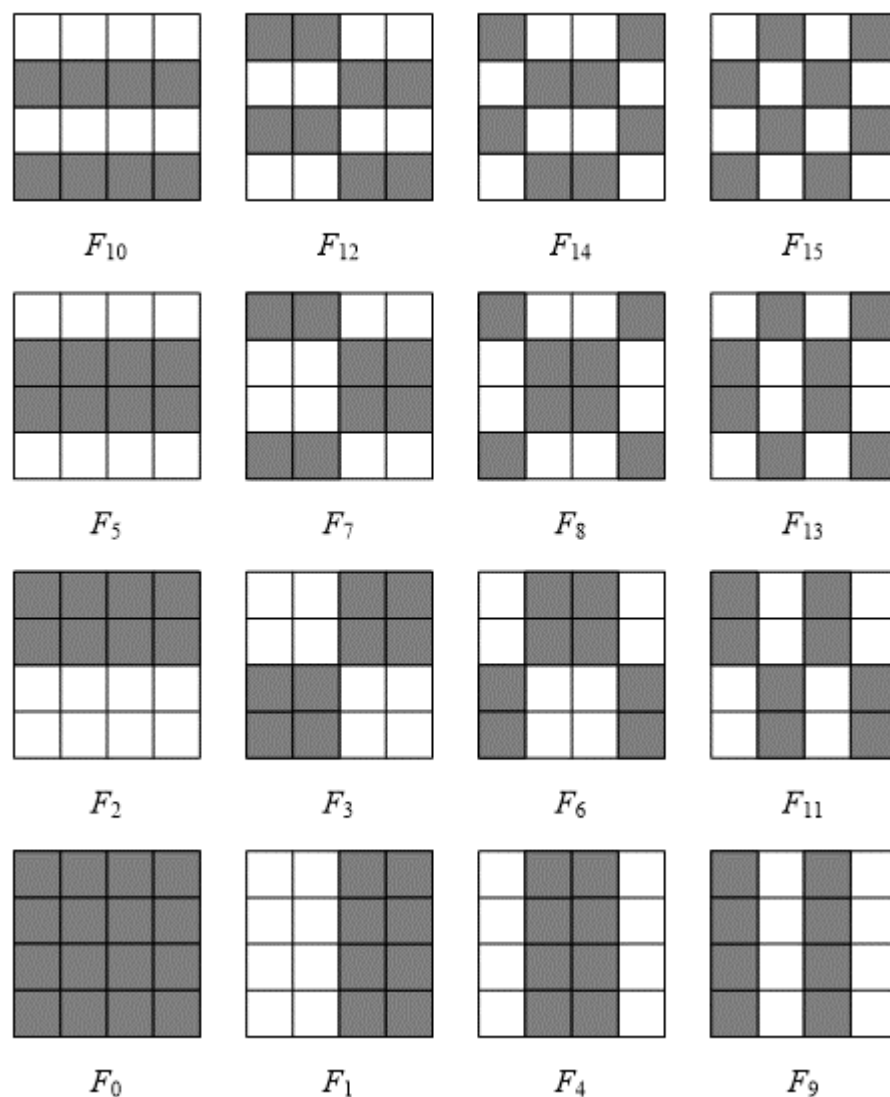


Рис.3.2. Фильтры.

Наблюдается схожесть фильтров теории активного восприятия с фильтрами Уолша, однако они различаются алгоритмом применения к изображению. Фильтры теории активного восприятия изображения используются только после выполнения преобразования интегрирования (Q – преобразования).

Интегральное и дифференциальное преобразования вместе составляют U – преобразование (см. Рис. 3.3).



Рис.3.3. U – преобразование.

Программная часть системы будет состоять из 2 – ух частей:

1) основная программа поиска номерной пластины

Для написания системы поиска номерной пластины использован язык R, ввиду его ориентированности для работы с изображениями. Программа получает на вход изображение формата bmp и преобразовывает его в матрицу коэффициентов каналов изображения. С помощью U – преобразования получаем матрицу спектральных коэффициентов, с помощью которой, вычисляем область, где находится номерная пластина транспортного средства. На выход поступает изображения с выделенной область, отражающей местоположение номерной пластины, которое в дальнейшем будет отображено в клиентской программе.

2) клиентская программа

В клиентской программе пользователь средствами клиентской программы может загрузить изображение, на котором находится транспортное средство для последующего поиска номерной пластины, а также просмотреть результат поиска непосредственно в графической оболочке. Результатом работы является выведенное на экран изображение, на котором выделена прямоугольная область, в границах которой находится номерная пластина автотранспортного средства.

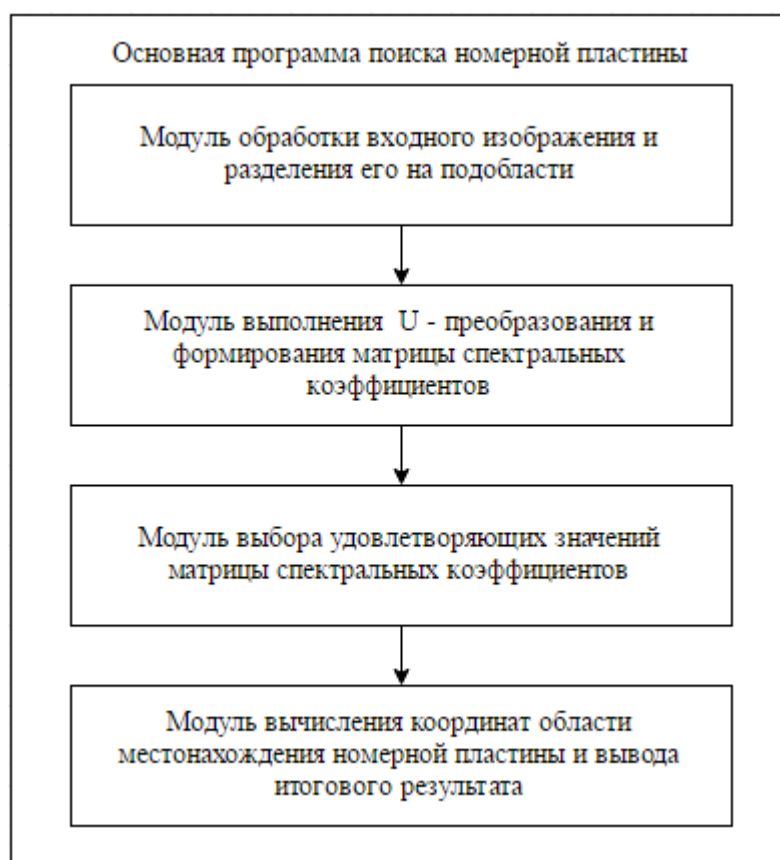


Рис. 3.4. Основная программа поиска номерной пластины.



Рис. 3.5. Клиентская программа.

Результат работы основной программы поиска номерной пластины должен быть отображен в графической оболочке, поэтому решено было сохранять результат работы основной программы на диск для дальнейшей работы с ним в клиентской части. Таким образом, итоговое изображение с выделенной областью, где находится номерная пластина сохранится в директории с программой.

Основная программа поиска номерной пластины состоит из нескольких модулей (см. Рис.3.4):

1)Модуль обработки входного изображения и разделения его на подобласти;

На вход этого модуля подается изображение, которое представляется, как матрица RGB каналов. Далее изображения переводится в градации серого, для дальнейшего нормирования. Нормирование происходит в несколько этапов:

1.1) Определяем минимальный элемент в матрице $\min(I)$, где I – изображение, матрицы $N \times M$.

1.2) Вычитаем из каждого элемента матрицы I найденный минимальный элемент:

$$I = I - \min(I)$$

1.3) Определяем максимальный элемент в матрице I .

1.4) Делим каждый элемент матрицы I на найденный максимальный элемент:

$$I = \frac{I}{\max(I)}$$

Последним этапом является разделение нормированной матрицы на небольшие подобласти.

2) Модуль выполнения U – преобразований и формирования матриц спектральных коэффициентов

В данном модуле выполняется операции интегрирования и дифференцирования. Далее из вектора спектральных коэффициентов составляется матрица размером 10×10 элементов.

3) Модуль выбора удовлетворяющих значений матрицы спектральных коэффициентов

Данный модуль отбирает необходимые значения, удовлетворяющие условиям возможного расположения номерной пластины.

4) Модуль вычисления координат области местонахождения номерной пластины и вывода итогового результата

По полученным номерам выбранных ячеек матрицы спектральных коэффициентов вычисляются координаты положения номерной пластины.

					ВКР-НГТУ-09.03.01-(13-B-2)-009-17	Лист
						15
Изм.	Лист	№ докум.	Подпись	Дата		

3.2. Разработка алгоритма предварительной обработки данных

Q – преобразование

Все изображение и любая его подобласть определения допускает преобразование проектирования, так как объект исследования состоит из множества положительных действительных чисел и ограничен полем зрения. Данное преобразование в «Теории активного восприятия изображения» определено, как Q – преобразование.

Для выполнений Q – преобразования необходимо разбить матрицу на блоки:

$$\frac{N}{4} * \frac{M}{4} \text{ (см. Рис. 3.6).}$$

0.1	0.1	0	0.3	0.1	0.1	0	0
0.2	0.2	0	0	0.2	0.2	0	0.7
0	0	0.8	0.7	0	0.2	0.8	0.7
0.1	0	0.9	1	0	0	0.9	1
0.1	0.1	0	0	0.1	0.1	0	0
0.2	0.2	0	0	0.2	0.2	0	0
0	0	0.8	0.7	0	0	0.8	0.7
0	0.9	0.9	1	0	0	0.9	1

→

0.6	0.3	0.6	0.7
0.1	3.4	0.2	3.4
0.6	0	0.6	0
0.9	3.4	0	3.4

Рис.3.6. Q – преобразование.

Q – преобразование выполняется по формуле:

$$B_{kl} = \sum_{i=1}^2 \sum_{j=1}^2 A_{kl}[i, j]$$

$$k = \overline{1,4} \quad l = \overline{1,4}$$

Q – преобразование относится к модулю «Обработки входного изображения» основной программы поиска номерной пластины. В результате выполнения операции интегрирования получаем матрицу визуальных масс, необходимую для формирования системы признаков (операция дифференцирования).

3.3. Разработка алгоритма формирования системы признаков

Одним из основных этапов в системе распознавания образов является формирование системы признаков. Системой признаков для данной задачи является вектор спектральных коэффициентов. Получение вектора спектральных коэффициентов является одной из основных задач разработанного алгоритма, на основе которого формируется дальнейшая система принятия решений.

После получения матрицы «визуальных масс» в результате выполнения Q – преобразования выполняется формирование вектора спектральных коэффициентов (см. Рис.3.8), методом наложения выбранного фильтра (см. Рис. 3.9), повторяющего форму номерной пластины.

$$\Sigma \left[\begin{array}{c} \text{Матрица} \\ \text{«визуальных масс»} \end{array} \times \begin{array}{c} \text{Фильтр } F_1 \end{array} \right] = -0.4$$

0.6	0.3	0.6	0.7
0.1	3.4	0.2	3.4
0.6	0	0.6	0
0.9	3.4	0	3.4

-1	-1	+1	+1
-1	-1	+1	+1
-1	-1	+1	+1
-1	-1	+1	+1

 $\mu =$

F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}
18.2	-0.4	0.4	1.4	-1.2	-1.6	0.6	0.6	1.4	-11	11.4	-1.6	-1.2	0.4	-0.4	-13.8

Рис. 3.8. Получения вектора спектральных коэффициентов.

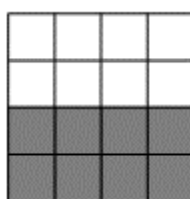


Рис. 3.9. Используемый фильтр.

3.4. Разработка алгоритма принятия решения

Для выбора подходящих значений матрицы спектральных коэффициентов необходимо разработать алгоритм принятия решений (см. Рис. 3.10).

Примерный размер номерной пластины равен 2, иногда 3 подобластям, поэтому основным критерием для поиска будет нахождение 2, иногда 3-х положительных значений, находящихся в одном ряду.

После выбора подходящей пары значений необходимо рассчитать координаты области, в которой находится номерная пластина. Координаты области высчитываются исходя из положения пары коэффициентов в матрице спектральных коэффициентов и из размера одной подобласти.

Если пара найденных коэффициентов имеют, например, номера 67 и 77 в матрице спектральных коэффициентов, а размер одной подобласти имеет размеры 50x50 пикселей, тогда область имеет координаты:

- 1) Верхнее левое: $x = 250, y = 300$;
- 2) Нижнее левое: $x = 250, y = 350$;
- 3) Верхнее правое: $x = 350, y = 300$;
- 4) Нижнее правое: $x = 350, y = 350$;

Номерная пластина, в некоторых ситуациях, может занимать 3 ячейки матрицы спектральных коэффициентов, для этого разработан алгоритм обработки данной нетипичной ситуации:

1) Если коэффициент слева, от найденной пары значений, является положительным значением и больше чем коэффициент справа (тоже положительное значение) более, чем в 3 раза, тогда область увеличивается влево на половину ячейки.

2) Если коэффициент справа, от найденной пары значений, является положительным значением и больше чем коэффициент слева (тоже положительное значение) более, чем в 3 раза, тогда область увеличивается вправо на половину ячейки.

3) Если коэффициенты слева и справа от выбранной пары значений положительные и разница их значений отличается менее, чем в 2 раза, тогда область увеличивается и слева и справа на 1/6 ячейки.

Пример пары значений матрицы спектральных коэффициентов приведен на рис. 3.10. Значение, которое находится слева от выбранной пары спектральных коэффициентов больше, значения справа более, чем в 3 раза, поэтому область будет сдвинута на половину ячейки влево.

-0.359504132231405	-0.210743801652893	-0.557851239669422	3.78099173553719
0.983471074380165	2.57851239669421	2.51652892561983	0.0165289256198347
-1.99586776859504	-1.78925619834711	-1.62396694214876	-1.64876033057851

Рис. 3.10. Выбранная пара спектральных коэффициентов.

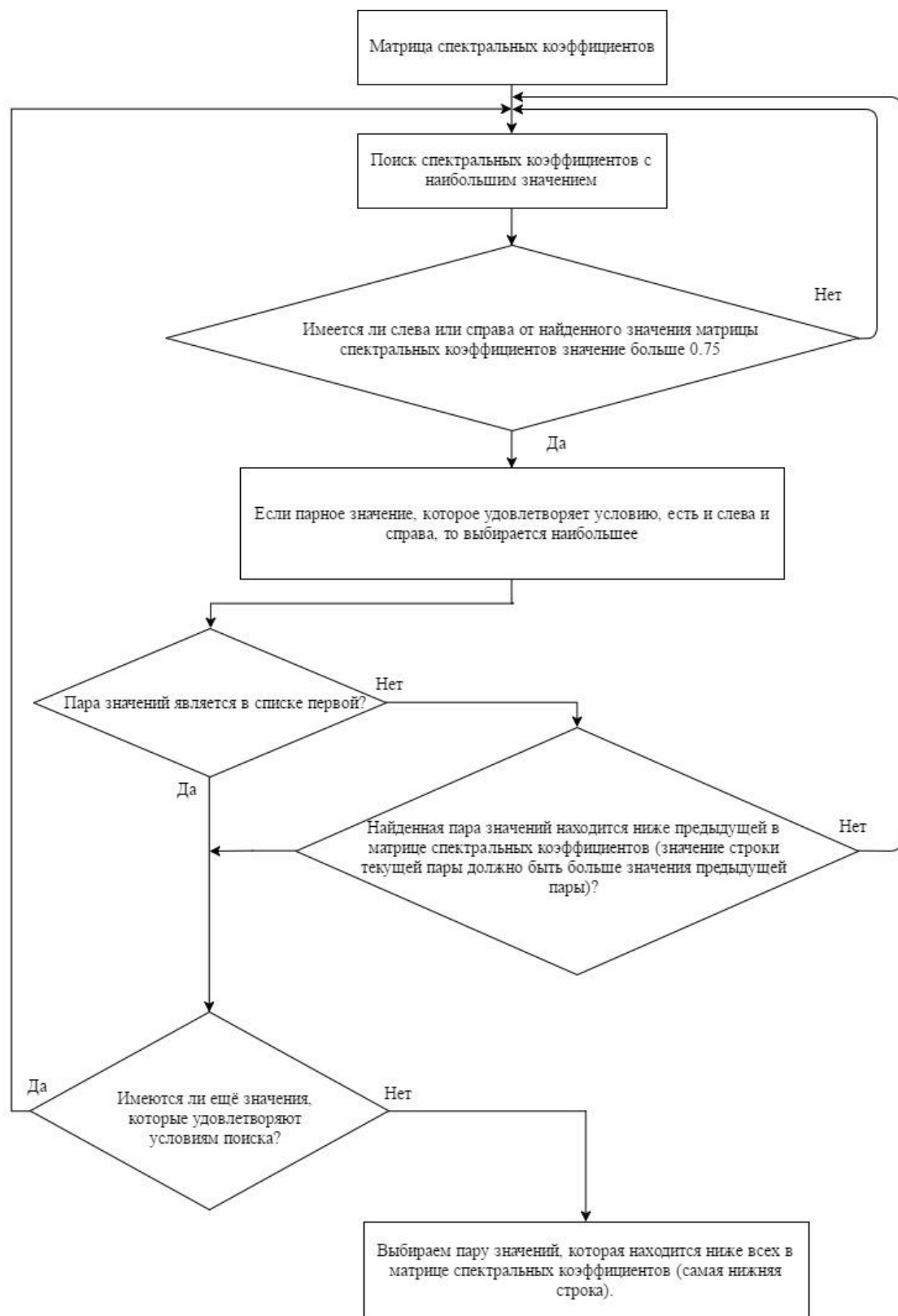


Рис. 3.11. Алгоритм получения пары спектральных коэффициентов.

4. Разработка программных средств

4.1. Разработка интерфейса пользователя системы

Для разработки интерфейса пользователя была использована библиотека Java Swing [6]. Для создания графического интерфейса были применены следующие классы и методы библиотеки Swing:

- 1) «JFrame» – пустое окно приложения.
- 2) «JButton» – кнопка. Для установки размеров и позиции кнопки внутри «JFrame» использовались методы «setSize» и «setLocation».
- 3) «JLabel» – текст. Необходим для вывода текстовой информации внутри «JFrame». Содержит методы аналогичные «JButton».
- 4) «JFileChooser» - необходим для создания диалога для выбора файлов. Позволяет производить навигацию по файловой системе (см. Рис. 5.2).

Программная реализация представлена на рис. 4.1.

```
chooseButton.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        JFileChooser fileopen = new JFileChooser();  
        int ret = fileopen.showDialog(null, "Открыть файл");  
        if (ret == JFileChooser.APPROVE_OPTION) {  
            File file = fileopen.getSelectedFile();  
            getFile = file.getAbsolutePath();  
            label.setText(file.getName());  
            try(FileWriter writer = new FileWriter("path.txt", false))  
            {  
                writer.write(getFile);  
            }  
            catch(IOException ex){  
                System.out.println(ex.getMessage());  
            }  
        }  
    }  
});
```

Рис.4.1. Программная реализация «JFileChooser».

В первой строке создается новый экземпляр класса «JFileChooser», а при помощи второй строки отображается диалог пользователю. После отображения диалога пользователь выбирает необходимый файл, путь к которому с помощью функции «getAbsolutePath» присваивается строковой переменной «getFile», для последующей записи в промежуточный текстовый документ «path.txt», который будет использован основной программой поиска.

Для обработки нажатий на кнопки использован метод «addActionListener». Пример обработчика нажатий для кнопки выхода из программы:

```

exitButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        System.exit(0); //Выход из программы
    }
});

```

Рис. 4.2. Обработчик нажатий.

Для вывода результата работы программы в виде изображения используется метод «showMessageDialog» класса «JOptionPane» (см. Рис. 4.3). Данный метод предназначен для вывода в графической интерфейс различной информации.

```

JOptionPane.showMessageDialog(null, icon, "Найденный номер", JOptionPane.PLAIN_MESSAGE);

```

Рис. 4.3. Вывод результата работы.

4.2. Программная реализация модулей системы

4.2.1. Модуль обработки входного изображения и разделения его на подобласти

Первым этапом является создание начальной матрицы каналов с помощью функции «read.bmp(location)», где «location» – это местоположение bmp файла в файловой системе, а «read.bmp()» – функция, которая переводит изображение в матрицу каналов.

Далее изображению переводится в градации серого путем сложения RGB каналов:

```
if (length(dim(img)) == 3)
{
  img <- img[, ,1] + img[, ,2] + img[, ,3];
  img <- img / 3;
}
```

Для выполнения Q – преобразований необходимо выполнить нормирование и создать фильтр для дальнейшего получения вектора спектральных коэффициентов:

```
img <- img - min(img)
img <- img / max(img)
```

Создание фильтра для дальнейшего получения вектора спектральных коэффициентов:

```
filt1<-matrix(c(1,1,-1,-1), nrow = 4, ncol=4)
```

Следующим этапом является разделение нормированной матрицы на подобласти. Результатом является массив подобластей «listMass», содержащий в себе подобласти, готовые к выполнению Q – преобразования.

```
for (j in seq(1,20,2))
{
  for (i in seq(1,20,2))
  {
    dtFirst <- img[swAll[i]:swAll[i + 1],
shAll[j]:shAll[j + 1]]
    listMass[[j1]] <- list(dtFirst)
    j1 = j1 + 1
  }
}
```

4.2.2 Модуль выполнения U – преобразования и формирования матрицы спектральных коэффициентов

Результатом работы данного модуля является матрица спектральных коэффициентов, с помощью которой, определяется положение номерной пластины на изображении.

Перед выполнением Q – преобразования матрица записывается в промежуточную переменную «listProm», после чего выполняется Q – преобразование поочередно каждой подобласти. Результатом является массив, состоящий из матриц визуальных масс «listms».

```
for (j in seq(1,8,2))
{
```

```

for (k in seq(1,8,2))
{
  dt <- listProm[sw[j]:sw[j], sh[k]:sh[k]]
  mass[k1,j1] = sum(dt)
  listms[[i]] <- list(mass)
  j1 = j1 + 1
}
j1 = 1
k1 = k1 + 1
}
}

```

Итогом работы модуля является получение матрицы спектральных коэффициентов «spectrFin» с помощью ранее полученного массива матриц визуальных масс «listms» и наложения фильтра «filt1».

```

for(j in 1:100)
{
  spectr[j]<-sum(filt1*unlist(listms[[j]]))
}
spectrFin<-matrix(spectr,nrow=10,ncol=10)

```

4.2.3 Модуль выбора удовлетворяющих значений матрицы спектральных коэффициентов

В данном блоке основной задачей является поиск максимальных значений, и затем подбор пары для этих значений. Максимальное значение не должно находиться в 1, 2, 9 и 10 столбцах, а также выше 2 и ниже 9 строки матрицы спектральных коэффициентов, значение коэффициента должно быть больше 0.75. Парное значение должно удовлетворять тем же условиям и находиться рядом с первым найденным значением.

```

for(j in 2:5){
  for(i in 1:100){
    if((spectrFin2[[i]] > listNumber[[1]]) & i > 20 &
i < 90 & (i %% 10) > 2 & (i %% 10) < 10 &
spectrFin2[[i]] > 0.75)
    {
      if((spectrFin2[[i + 10]] > 0.75 |
(spectrFin2[[i - 10]] > 0.75)))
      {
        if(j > 2 & firstCoord[[j - 1]] != 0 & ((i %%
10) > (firstCoord[[j - 1]] %% 10)))
        {
          listNumber[[1]] <- spectrFin2[[i]]
          firstCoord[[j]] <- i
        }
        if(j == 2){
          listNumber[[1]] <- spectrFin2[[i]]
          firstCoord[[j]] <- i
        }
      }
    }
  }
}

```

Из всех найденных пар значений выбирается та пара, значение строки которой самое большое, то есть выбирается самая нижняя пара:

```
coordA<-firstCoord[[2]]
coordB<-secondCoord[[2]]
for(i in 2:5){
  if(firstCoord[[i]] %% 10 > coordA %% 10)
  {
    coordA<-firstCoord[[i]]
  }
  if(secondCoord[[i]] %% 10 > coordB %% 10)
  {
    coordB<-secondCoord[[i]]
  }
}
```

4.2.4 Модуль вычисления координат области местонахождения номерной пластины и вывода итогового результата

После нахождения номеров удовлетворяющих спектральных коэффициентов в матрице необходимо вычислить 4 точки, из которых формируется прямоугольная область, содержащая номерную пластину.

Вычисление координат прямоугольной области на примере верхней левой точки:

```
LeftTopX <- saveSh * (coordA %/% 10)
LeftTopY <- saveSw * (coordA %% 10 - 1) - (saveSw / 3)
```

Если же рядом есть положительные значения удовлетворяющее условиям алгоритма (на случай если рамка в 3х ячейках, а не двух), то область необходимо усреднить, то есть, либо сдвинуть влево, вправо, или расширить с левого и правого концов:

Найденная область вырисовывается в начальной матрице каналов «img» и далее сохраняется на диск в формате JPG с помощью функции «writeJPEG» из библиотеки «JPEG» языка R [8]:

```
writeJPEG(img, target = "NewImage.jpg")
```


5. Тестирование системы

5.1. Описание набора данных

Для тестирования использовались фотографии автомобилей, имеющих различную степень загрязнения и поврежденности номерной пластины. Также для проверки работы системы на нетипичных случаях использовались изображения, сделанные в солнечную погоду с использованием автомобилей, имеющих яркое лакокрасочное покрытие (белое), а также изображения с несколькими автомобилями.

Для тестирования использовались изображения, размер которых примерно равен 1000x1000 пикселей, где транспортное средство находится приблизительно в центре изображения.

Ниже приведены тестовые изображения:



Рис. 5.1. Тестовое изображение 1



Рис. 5.2. Тестовое изображение 2



Рис. 5.3. Тестовое изображение 3

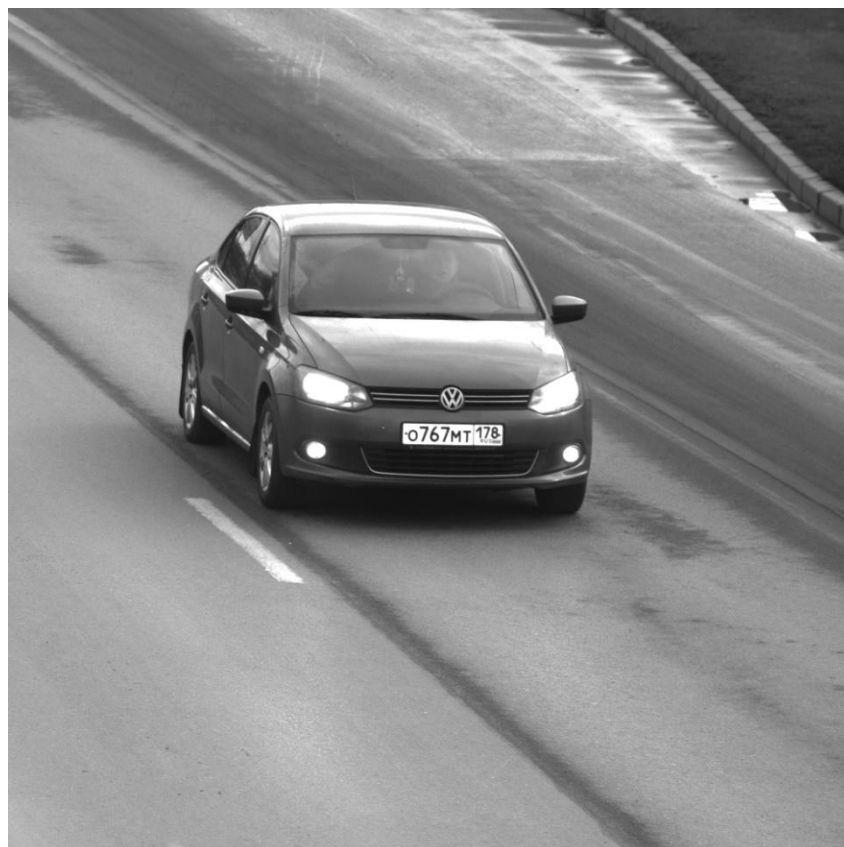


Рис. 5.4. Тестовое изображение 4



Рис. 5.5. Тестовое изображение 5



Рис. 5.6. Тестовое изображение 6

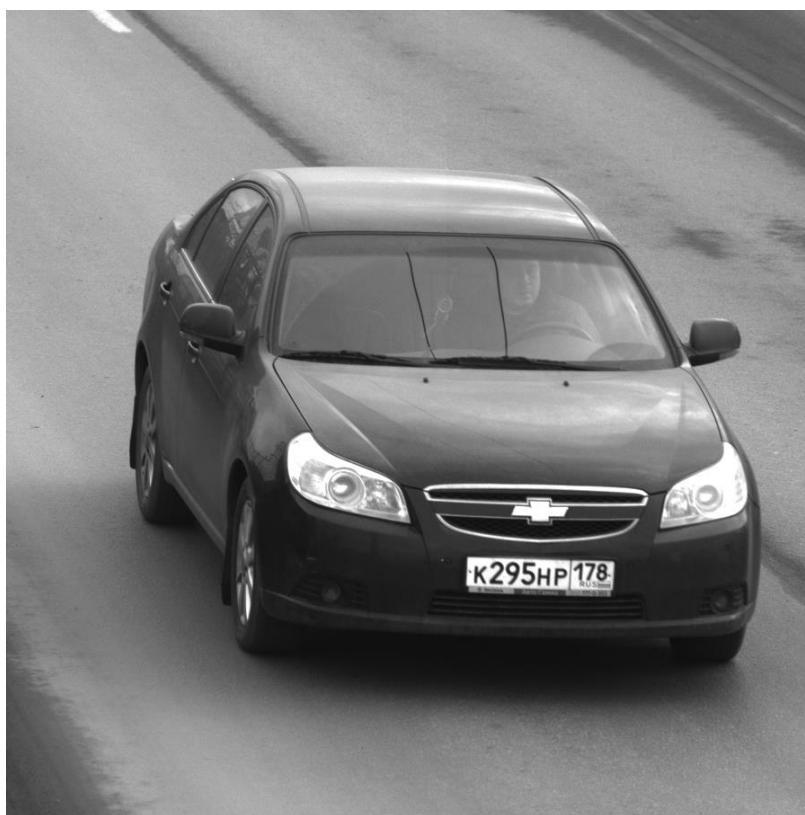


Рис. 5.7. Тестовое изображение 7

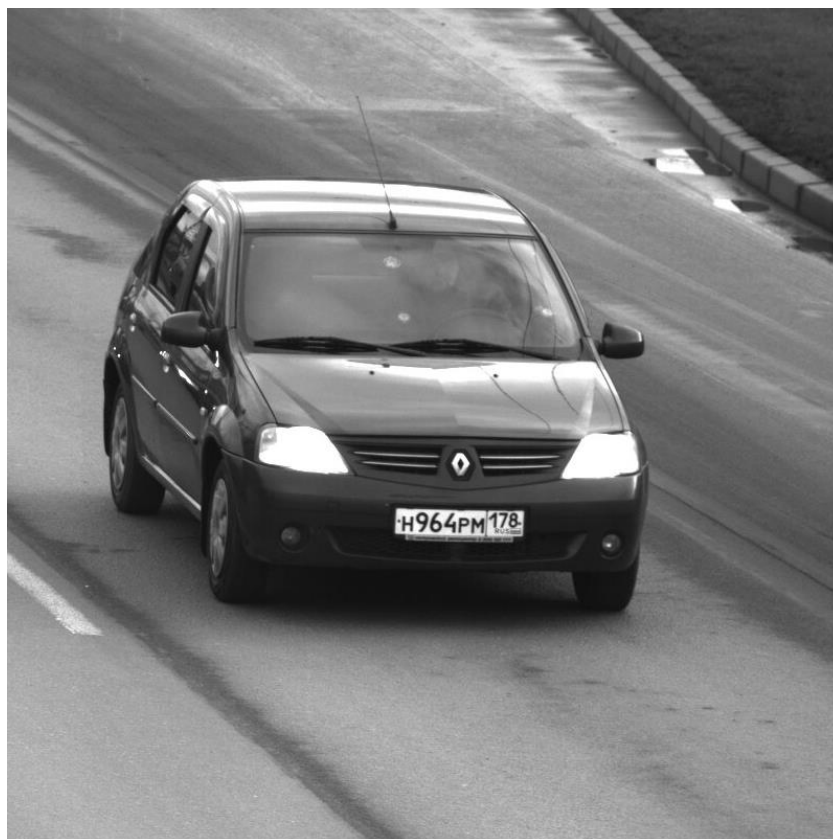


Рис. 5.8. Тестовое изображение 8



Рис. 5.9. Тестовое изображение 9



Рис. 5.10. Тестовое изображение 10



Рис. 5.11. Тестовое изображение 11

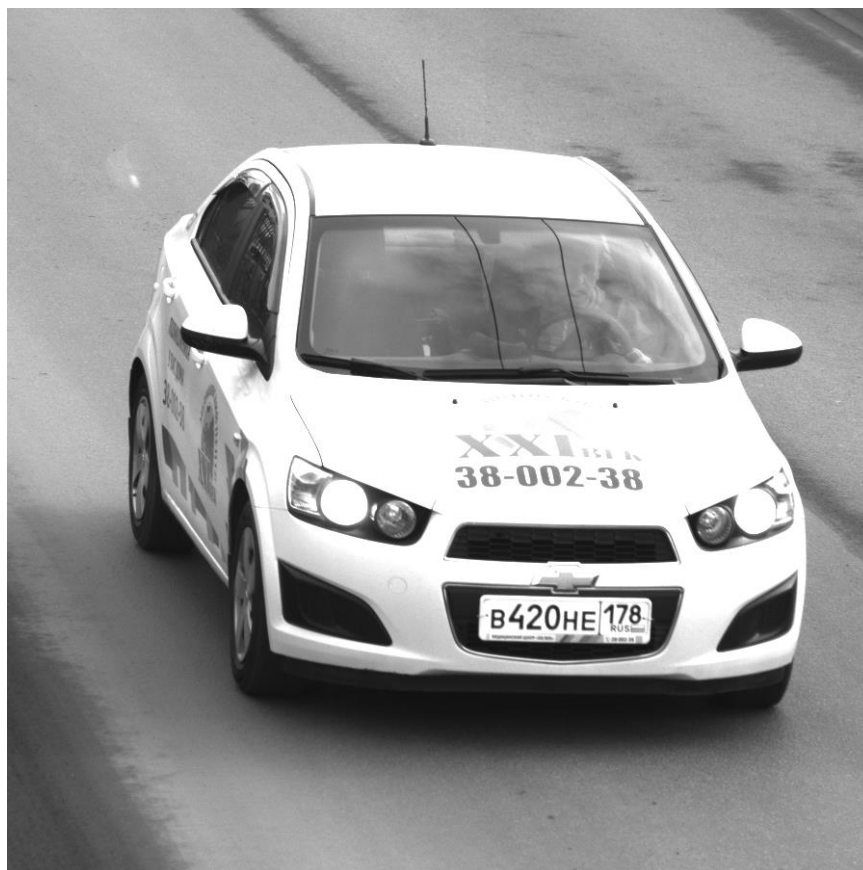


Рис. 5.12. Тестовое изображение 12

					ВКР-НГТУ-09.03.01-(13-В-2)-009-17	Лист
						31
Изм.	Лист	№ докум.	Подпись	Дата		

5.2. Описание методики тестирования

Для тестирования системы необходимо открыть пользовательский интерфейс клиентской части приложения и с его помощью, загружая изображения, параметры которых описаны в пункте 5.1, проверить работоспособность системы. Для проверки работоспособности была создана подборка из различных изображений.

Рассмотрим пользовательский интерфейс:

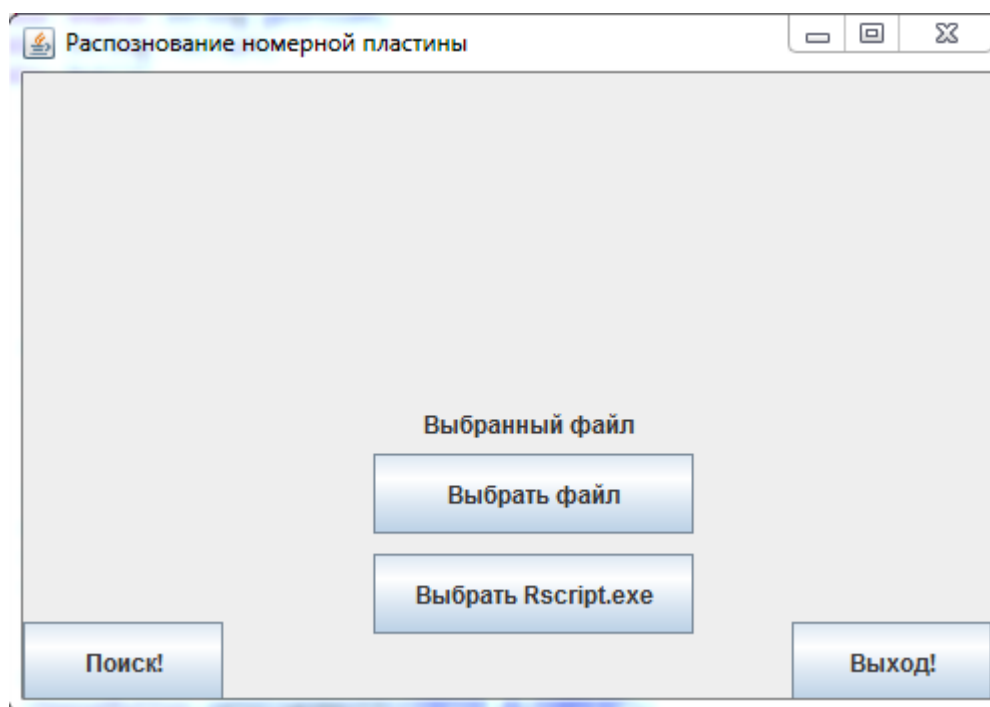


Рис.5.13. Пользовательский интерфейс.

Порядок работы системы поиска номерной пластины:

1) Выбрать изображение формата bmp, на котором осуществляется поиск и выбрать исполняемый файл Rscript.exe, необходимый для запуска основной части программы, написанной на R (см. Рис. 5.14).

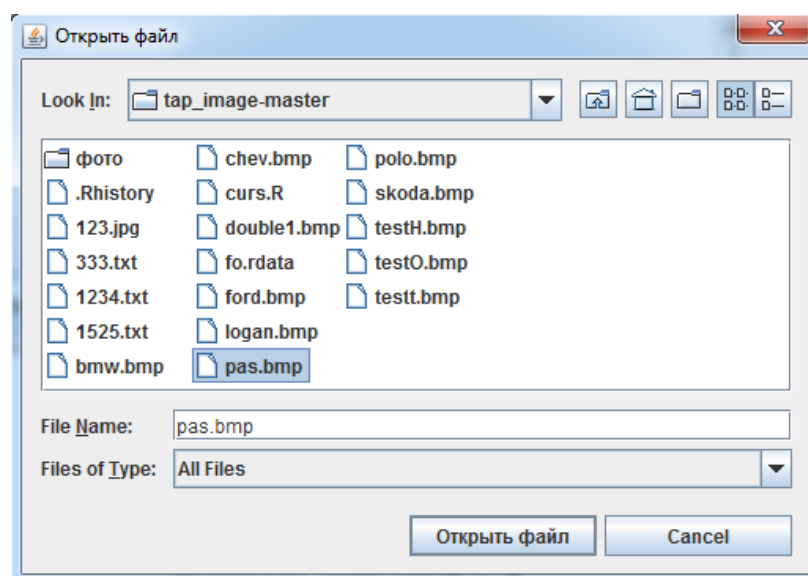


Рис. 5.14 Выбор входных компонентов.

2)Нажать кнопку «Старт».

Результатом работы программы является, появившееся в пользовательском интерфейсе изображение, которое содержит выделенную область, отражающую местоположение номерной пластины (см. Рис. 5.15).

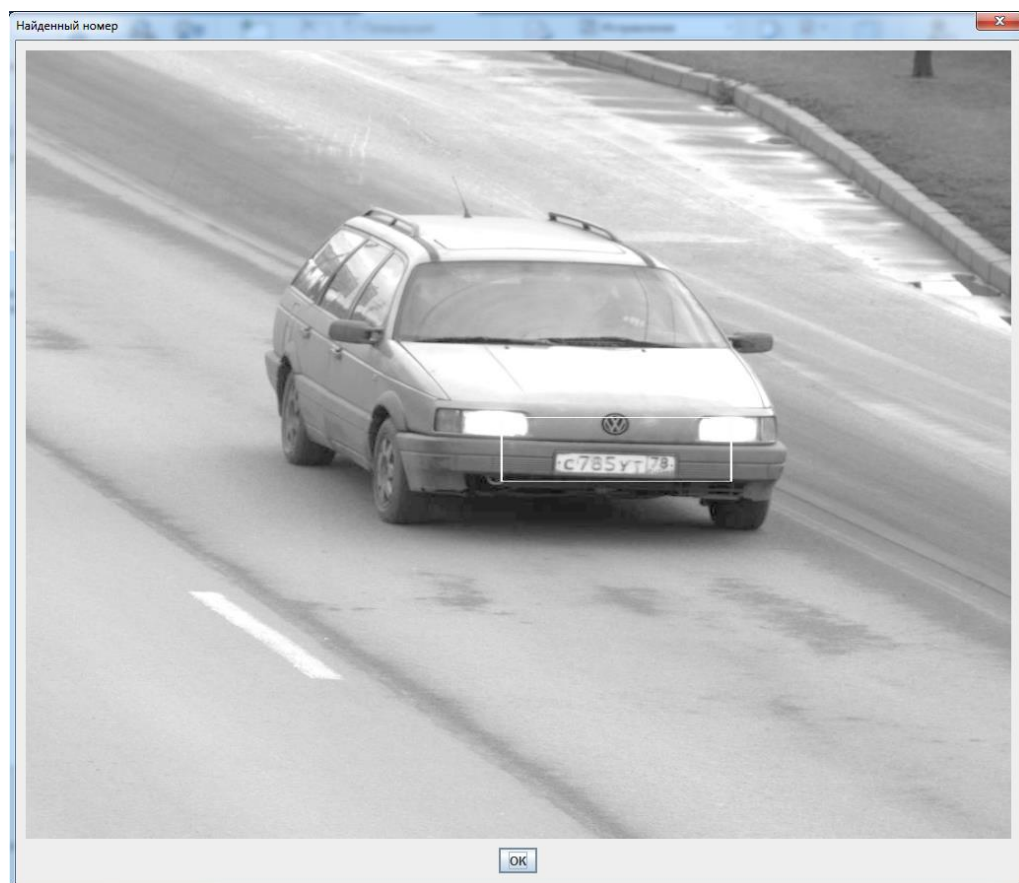


Рис.5.15. Результат работы программы.

5.3. Результаты вычислительного эксперимента

Тестовое изображение 1 показано на рис. 5.15.



Рис. 5.15. Результат поиска 1

Тестовое изображение 2 показано на рис. 5.16.



Рис. 5.16. Результат поиска 2

На тестовом изображении 3, показанном на рис. 5.17, номерная пластина достаточно блеклая и изношенная.



Рис. 5.17. Результат поиска 3

Тестовое изображение 4 показано на рис. 5.18.



Рис. 5.18. Результат поиска 4

На тестовом изображении номер 5, показанном на рис. 5.19, номерная пластина достаточно блеклая и загрязненная.



Рис. 5.19. Результат поиска 5

Тестовое изображение 6 показано на рис. 5.20.



Рис. 5.20. Результат поиска 6

					ВКР-НГТУ-09.03.01-(13-В-2)-009-17	Лист
						39
Изм.	Лист	№ докум.	Подпись	Дата		

Тестовое изображение 7 показано на рис. 5.21.



Рис. 5.21. Результат поиска 7

					ВКР-НГТУ-09.03.01-(13-В-2)-009-17	Лист
						40
Изм.	Лист	№ докум.	Подпись	Дата		

Тестовое изображение 8 показано на рис. 5.22.



Рис. 5.22. Результат поиска 8

На тестовом изображении 9, показанном на рис. 5.23, присутствуют несколько транспортных средств, однако система построена таким образом, чтобы находить номерную пластину на ближайшем транспортном средстве.



Рис. 5.23. Результат поиска 9.

На тестовом изображении 10, показанном на рис. 5.24, изображен автомобиль с изношенной номерной пластиной, и имеющий яркий окрас, что дополнительно усложняет поиск.



Рис. 5.24. Результат поиска 10.

На тестовом изображении 11, показанном на рис. 5.25, номерная пластина загрязнена очень сильно и не выделяется, поэтому итоговая область, отражающая положение номерной пластины съехала вниз и влево. В данной ситуации программные средства бессильны.



Рис. 5.25. Результат поиска 11.

На тестовом изображении 12, показанном на рис. 5.26, изображено транспортное средство с хорошо читаемыми регистрационными знаками, однако имеющее белое лакокрасочное покрытие, что усложняет поиск номерной пластины.



Рис. 5.26. Результат поиска 12.

Заключение

В процессе выполнения выпускной работы была спроектирована и реализована программная система поиска номерной пластины автотранспортного средства на изображении.

Система предназначена для поиска на изображении, содержащем автотранспортное средство, местоположения номерной пластины и выделения области, в которой она расположена. В результате тестирования данной системы была доказана её работоспособность и возможность решать поставленную задачу.

В дальнейшем планируется добавить возможность работы данной системы с уличными IP – камерами.

					ВКР-НГТУ-09.03.01-(13-В-2)-009-17	Лист
Изм.	Лист	№ докум.	Подпись	Дата		46

Список литературы

1. Библиотеки для автоматического распознавания номеров авто и их практическое использование – <https://www.ibm.com/developerworks/ru/library/os-avto/>
2. Распознавание номеров: от А до 9 – <https://habrahabr.ru/company/recognitor/blog/221891/>
3. Утробин, Владимир Александрович “Компьютерная обработка изображений. Основы специальной теории восприятия” – Нижний Новгород : Нижегородский гос. технический ун – т им. Р. Е. Алексеева, 2015
4. Методика построения систем распознавания автомобильного номера – <https://cyberleninka.ru/article/n/metodika-postroeniya-sistem-raspoznavaniya-avtomobilnogo-nomera>
5. «Наглядная статистика. Используем R!» – Алексей Шипунов, Е. Балдин, Полина Волкова, А. Коробейников, С. Назарова, С. Петров, В. Суфиянов. – «ДМК Пресс», Москва, 2014 г.
6. Java Swing – <https://javaswing.wordpress.com/>
7. Сравнение Windows и Linux – https://ru.wikipedia.org/wiki/Сравнение_Microsoft_Windows_NT_и_Linux
8. Package «jpeg» – <https://cran.r-project.org/web/packages/jpeg/jpeg.pdf>
9. Java – <https://ru.wikipedia.org/wiki/Java>
10. Eclipse – [https://ru.wikipedia.org/wiki/Eclipse_\(среда_разработки\)](https://ru.wikipedia.org/wiki/Eclipse_(среда_разработки))