

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт Космических и Информационных Технологий

институт

Информационные системы

кафедра

УТВЕРЖДАЮ

Заведующий кафедрой ИС

\_\_\_\_\_ Л.С. Троценко

подпись

инициалы, фамилия

« 13 » 06 2018 г.

**БАКАЛАВРСКАЯ РАБОТА**

09.03.02 Информационные системы и технологии

Разработка программного обеспечения для распознавания объектов на  
графическом изображении

Руководитель

\_\_\_\_\_  
подпись, дата

П. П. Дьячук

инициалы, фамилия

Выпускник

\_\_\_\_\_  
подпись, дата

А. Г. Рыкова

инициалы, фамилия

Нормоконтролер

\_\_\_\_\_  
подпись, дата

Ю.В. Шмагрис

инициалы, фамилия

Красноярск 2018

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт Космических и Информационных Технологий  
институт  
Информационные системы  
кафедра

УТВЕРЖДАЮ  
Заведующий кафедрой ИС  
\_\_\_\_\_ С.А. Виденин  
подпись инициалы, фамилия  
« 05 » \_\_\_\_\_ 03 \_\_\_\_\_ 2018г.

**ЗАДАНИЕ**  
**НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ**  
**в форме бакалаврской работы**

Студенту Рыковой Анне Геннадьевне

фамилия, имя, отчество

Группа КИ14-13Б Направление 09.03.02

номер

код

Информационные системы и технологии

наименование

Тема выпускной квалификационной работы: Разработка программного обеспечения для распознавания объектов на графическом изображении

Утверждена приказом по университету № 4896/с от 05.04.2018г.

Руководитель ВКР П.П. Дьячук, кандидат педагогических наук, доцент кафедры «Информационные системы» ИКИТ

инициалы, фамилия, должность, ученое звание и место работы

Исходные данные для ВКР: Требования к разрабатываемому программному обеспечению, описание предметной области

Перечень разделов ВКР: Введение, анализ предметной области, проектирование программного обеспечения, описание работы программного обеспечения, заключение, список использованных источников

Перечень графического материала: Презентация, выполненная в Microsoft Office PowerPoint 2013

Руководитель ВКР

\_\_\_\_\_

П.П. Дьячук

подпись

инициалы и фамилия

Задание принял к исполнению

\_\_\_\_\_

А. Г. Рыкова

подпись

инициалы и фамилия

«05» 03 2018 г.

## РЕФЕРАТ

Выпускная квалификационная работа по теме «Разработка программного обеспечения для распознавания объекта на графическом изображении» содержит 44 страницы текстового документа, 19 использованных источников, 25 рисунков, 2 таблицы.

РАСПОЗНАВАНИЕ, МАШИННОЕ ЗРЕНИЕ, КЛАССИФИКАЦИЯ ОБЪЕКТОВ, ОБУЧЕНИЕ СИСТЕМЫ, ИЗОБРАЖЕНИЯ, OPENCVSHARP, WINDOWSFORMS.

Цель: разработка программного обеспечения для распознавания объектов на изображении.

В выпускную квалификационную работу входит введение, 3 главы и заключение.

Во введении раскрывается актуальность работы, ставится цель и освещаются основные задачи.

В первой главе описан теоретический материал для разработки приложения.

Во второй главе описаны постановка задачи, средства разработки, логическая структура и архитектура программного обеспечения.

В третьей главе описаны используемые классы и методы, функционал, алгоритм работы программного обеспечения и его взаимодействия с пользователем.

В заключении подводятся итоги по выполненной работе.

## СОДЕРЖАНИЕ

|   |    |
|---|----|
| ВВЕДЕНИЕ .....  | 4  |
| 1 Анализ предметной области .....   | 5  |
| 1.1 Методы обучения машинному зрению.....   | 5  |
| 1.2 Методы обработки изображений .....  | 6  |
| 1.3 Методы распознавания изображений .....  | 8  |
| 1.4 Анализ существующего программного обеспечения .....                                     | 10 |
| 1.4.1 Самообучаемая программа IDE-LME от Disney Research для<br>распознавания образов. .... | 10 |
| 1.4.2 Система распознавания объектов на фотографии Google Lens. ....                        | 11 |
| 1.4.3 Распознавание текста.....   | 13 |
| 1.4.4 Распознавание лиц .....   | 15 |
| 1.5 Выводы.....   | 17 |
| 2 Проектирование программного обеспечения.....  | 19 |
| 2.1 Постановка задачи.....  | 19 |
| 2.2 Инструментальные средства разработки .....  | 19 |
| 2.3 Логическая структура .....  | 20 |
| 2.4 Архитектура программного обеспечения.....   | 22 |
| 3 Описание работы программного обеспечения.....   | 23 |
| 3.1 Используемые классы и методы.....   | 23 |
| 3.2 Функционал программного обеспечения .....   | 30 |
| 3.3 Алгоритм работы программы .....   | 30 |
| 3.4 Работа с программным обеспечением .....   | 36 |
| ЗАКЛЮЧЕНИЕ .....  | 41 |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....  | 42 |

## ВВЕДЕНИЕ

С развитием вычислительной техники всё большая часть задач стремится быть автоматизированной, упрощенной. Стремление к повышению скорости, точности, качества решения проблем и является основным толчком к развитию информационных технологий и вычислительной техники. Автоматизировать механический труд человека (упаковка, сборка деталей, рассылка e-mail и т.д.) уже удалось в большой мере, сейчас становится всё более востребованным разработка различных аналитических, обучающихся, принимающих решение систем. И один из больших разделов этой области: распознавание объектов и машинное зрение. Это очень объемный и интересный раздел, реализация которого может быть применена в разных областях.

Исходя из этого была поставлена цель выпускной квалификационной работы: разработка программного обеспечения для распознавания объектов на изображении.

Требуется создание программного обеспечения, использование которого позволит обучать распознаванию и классифицировать объекты, считанные с изображения.

Для достижения цели ставятся следующие задачи:

- исследовать методы обучения машинному зрению, обработки и распознавания изображений;
- определить используемые алгоритмы;
- исследовать существующие аналоги;
- определить требования к разрабатываемому программному обеспечению;
- определить инструментальные средства разработки;
- разработать основные сущности программного обеспечения;
- разработать программное обеспечение.

## **1 Анализ предметной области**

Для разработки программного обеспечения для распознавания необходимо изучить теоретические сведения по алгоритмам, используемым технологиям и сделать обзор на существующие аналоги. Это методы обучения машинному зрению, методы обработки и распознавания изображений, используемый язык программирования и необходимые библиотеки.

### **1.1 Методы обучения машинному зрению**

Первый и самый простой метод – сравнивать полученное изображение с сформированной базой данных изображений этого класса. В чем минусы такого подхода? Этот метод можно назвать «заучиванием наизусть» - а это очень ресурсозатратно: нужно постоянно хранить серьёзную подборку изображений каждого класса объектов, и постоянно пополнять её при возникновении новых, не предугаданных условий нахождения объекта на изображении (неожиданный ракурс, освещение, перекрытие объекта другим и т.д.). И алгоритм каждый раз, при каждом распознавании должен будет «пробегать» всю эту огромную подборку изображений, пытаясь найти общие признаки с исходным изображением.

Метод опорных векторов: суть данного метода заключается в том, чтобы каждый объект был описан определенным вектором признаков. Задача состоит в том, чтобы построить функцию, зависящую от вектора признаков. Для каждого вектора – построенная функция и будет ответом, к какому классу принадлежит объект. Эта функция по сути делит всё пространство на регионы, нахождение в которых считается отношением к определенному классу. В данном методе необходимо создать наиболее выгодную гиперплоскость – пространство между положительными и отрицательными признаками. Есть определенная сложность представления этого алгоритма для случаев когда классов больше двух, так как по сути необходимо организовать  $n$ -мерное пространство. Здесь определяются

ядровые функции, которые позволят строить функции в определенном пространстве, на определенной плоскости.

Метод Виолы-Джонса. Для этого метода необходимо, во-первых, представить исходные данные (изображения) в интегральном виде: это матрица чисел, отражающих яркость каждого пикселя. Это необходимо для того, чтобы быстрее рассчитывать признаки Хаара изображения. Признаки Хаара дают точечное значение перепада яркости по осям. Это происходит, как наложение масок и создание наиболее общей картины: можно сказать формируется изображение-шаблон (матрица-шаблон, в случае использования интегрального вида). Основным минус такого подхода, что точность зависит от количества, представленных на обучение изображений, а следовательно и время обучения с увеличением выборки растет.

## 1.2 Методы обработки изображений

Прежде чем система приступит к распознаванию объекта, система приводит изображение в необходимый, более «понятный» для нее вид, убирая лишнее. На рисунке 1 изображена схема последовательности обработки изображения.

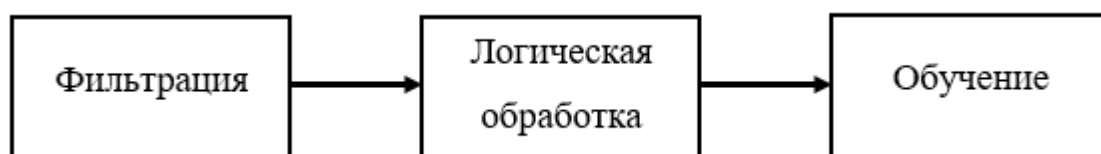


Рисунок 1 – Последовательность обработки изображения

На уровне фильтрации анализ изображения не производится, но точки, которые проходят фильтрацию, можно рассматривать как области с особыми характеристиками.

Самое простое преобразование — это бинаризация изображения по порогу. Пример бинаризации отображен на рисунке 2. Обычно бинаризация



осуществляется с помощью алгоритма, который адаптивно выбирает порог. Таким алгоритмом может быть выбор математического ожидания или моды. А можно выбрать наибольший пик гистограммы.



Рисунок 2 – Результат бинаризации

Отдельный класс фильтров — фильтрация границ и контуров. Контурные 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. На рисунке 3 отображен пример выделения контуров объекта.



Рисунок 3 – Выделение контуров объекта

Переходом от фильтрации к логике являются морфологические операции. По сути, это простейшие операции наращивания и эрозии бинарных изображений. Эти методы позволяют убрать шумы из бинарного изображения, увеличив или уменьшив имеющиеся элементы. На рисунке 4 будут отображены результаты применения морфологических операций. Особые точки — это уникальные характеристики объекта, которые позволяют сопоставлять объект сам с собой или с похожими классами объектов.

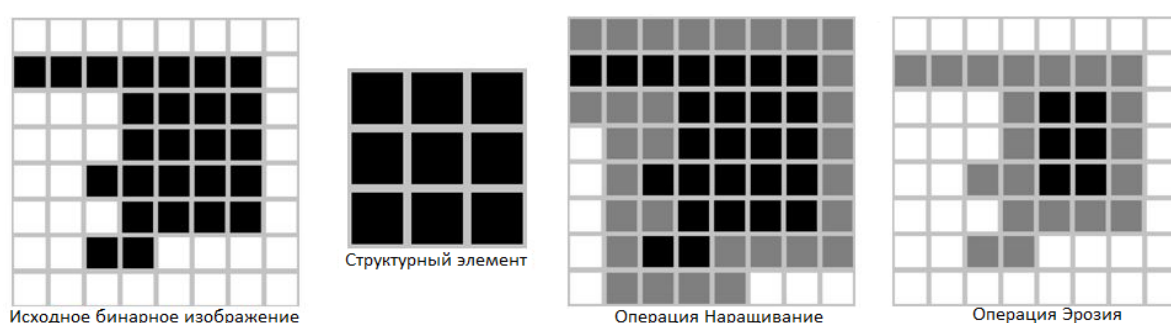


Рисунок 4 – Результат применения морфологических операций

### 1.3 Методы распознавания изображений

Методы, основанные на извлечении признаков. Один из возможных подходов к решению задачи детектирования состоит в том, чтобы использовать алгоритмы машинного обучения для построения моделей классов объектов и алгоритмы вывода для поиска объектов на изображении. Техники данной группы описывают объект с использованием векторов-признаков. Таким образом, объект описывается набором векторов признаков в характерных точках. В результате тренировки строится модель, содержащая "усредненные" вектора признаков. Качество рассматриваемых методов в основном зависит от того, насколько хорошо выбраны признаки, т.е. насколько хорошо эти признаки дифференцируют классы объектов. Все правила необходимо подбирать вручную, а признаков может быть очень много (цвет, площадь, положение,

границы, форма, соотношение сторон, каналы цвета). И это всё необходимо еще умножить на количество разнообразных ракурсов, фонов, освещений и т.д. Суть данного метода отображена на рисунке 5.

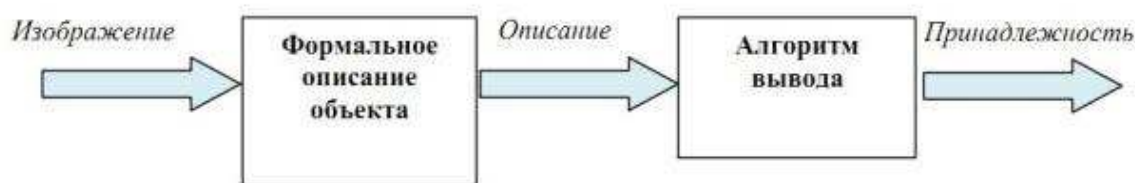


Рисунок 5 – Метод, основанный на извлечении признаков

Методы поиска по шаблону. Распознавание объектов на основании некоторого шаблона предполагает, что имеется изображение объекта с выделенными признаками – шаблон – и тестовое изображение, которое сопоставляется этому шаблону. Стоит учитывать, что методы детектирования по заданному шаблону эффективно работают при поиске одиночных объектов. При возникновении перекрытий в "бегущем окне" исчезают некоторые признаки в описании. Поэтому при сопоставлении окна шаблону вводится порог, по которому отсекаются неперспективные окна – окна, заведомо не содержащие объектов. Суть данного метода отображена на рисунке 6.



Рисунок 6 – Метод поиска по шаблону

## **1.4 Анализ существующего программного обеспечения**

Прежде чем приступать к разработке программного обеспечения, необходимо провести исследования существующих альтернативных решений.

### **1.4.1 Самообучаемая программа IDE-LME от Disney Research для распознавания образов.**

Группа исследователей из питтсбургского подразделения Disney Research разработала систему компьютерного зрения, которая использует некоторые принципы человеческого зрения. В частности, она содержит алгоритмы для самообучения и способна со временем улучшать распознавание объектов.

Как и большинство других систем компьютерного зрения, разработка Disney Research строит концептуальную модель для каждого объекта, будь это самолёт или дозатор мыла. При этом используется обучаемый алгоритм, который анализирует множество фотографий данного объекта.

Отличительная особенность алгоритма Disney Research — в том, что он впоследствии использует эту модель для распознавания объектов на видео, одновременно извлекая новую информацию об этих объектах и дополняя модель, заложенную изначально. Это позволяет распознавать объекты в более широком диапазоне, даже если они выглядят иначе, чем на встречавшихся ранее образцах.

На рисунках 7 и 8 показаны результаты распознавания образов. В верхнем ряду находятся тестовые изображения из базы ImageNet, которые использовались для обучения изначальной модели. В нижнем ряду — примеры корректного распознавания объектов программой IDE-LME. Исследователи обращают внимание, что распознанные объекты на фотографиях существенно отличаются по внешнему виду от тех, которые использовались для обучения системы.



Рисунок 7 – Результат распознавания образов программой IDE-LME



Рисунок 8 – Результат распознавания образов программой IDE-LME

Концептуальная модель для каждого объекта постепенно расширяется и уточняется по мере того, как система сталкивается с новой информацией. Теоретически, такой метод может привести к тому, что действуя без надзора, система припишет объекту несвойственные для него характеристики, из-за чего возникнут ошибки распознавания. Но авторы программы говорят, что такая проблема пока не замечена. Данные исследования были представлены в 2015 году, саму программу в свободном доступе найти не удалось.

#### 1.4.2 Система распознавания объектов на фотографии Google Lens.

Сервис является частью приложения Google Фото, доступного бесплатно для платформ Android и iOS. В настоящее время смартфоны Google Pixel и Pixel XL — это два девайса, у которых есть доступ к Google Lens. И в ближайшей



перспективе в этом коротеньком списке если и появятся новые позиции, то их будет не много. Так что, пока Lens — это супер-эксклюзив для владельцев Pixel.

С помощью Lens пользователь может получать подробную информацию о различных заведениях, достопримечательностях или билбордах компаний, достаточно просто навести камеру на объект. Кроме того, система может использовать полученную информацию, например, считывать логин и пароль при распознавании наклейки провайдера на роутере и автоматически подключать смартфон к Wi-Fi.

На рисунке 9 отображен результат работы данного сервиса.

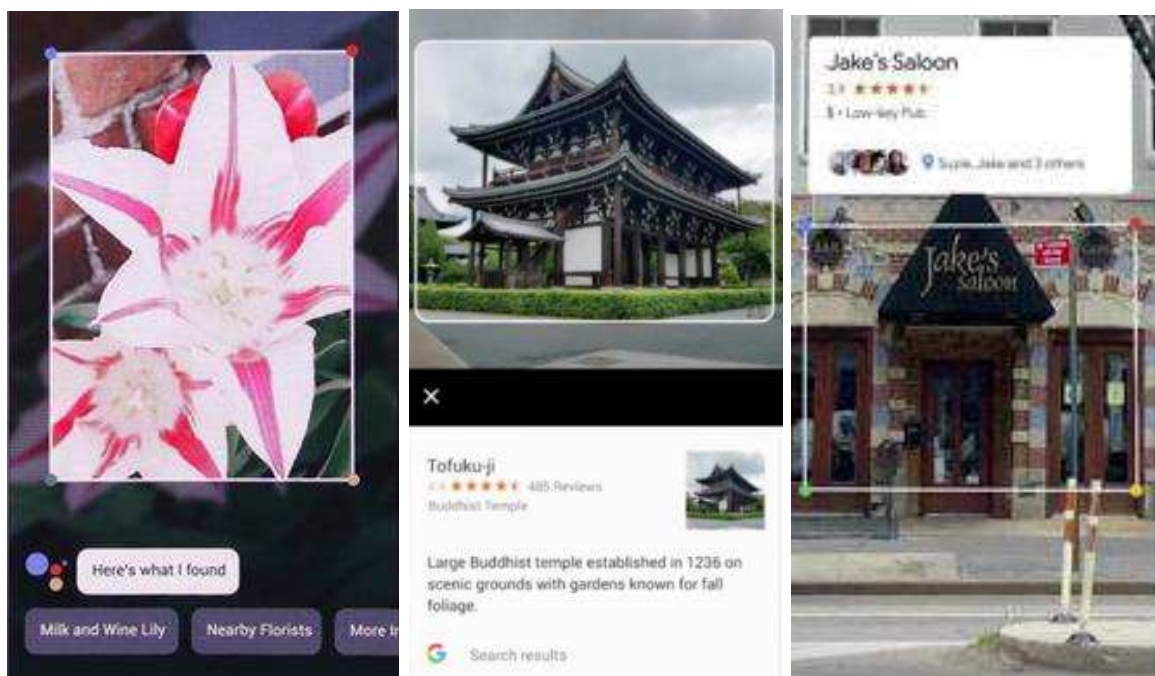


Рисунок 9 – Результат работы Google Lens

Для распознавания данных в Google Lens используется система искусственного интеллекта и машинного обучения. Помимо прочих факторов она учитывает текущее местоположение пользователя — эти данные полезны для распознавания сетевых заведений.

### 1.4.3 Распознавание текста

Пожалуй, самый часто встречающийся тип программ для распознавания – программы распознавания текста, полностью предназначенные для этого или частично использующие данные алгоритмы в своей работе.

- Один из ярких примеров таких программ: Office Lens. Этот сервис от компании Microsoft превращает камеру смартфона или ПК в мощный сканер документов. С помощью Office Lens вы можете распознать текст на любом физическом носителе и сохранить его в одном из «офисных» форматов или в PDF. Итоговые текстовые файлы можно редактировать в Word, OneNote и других сервисах Microsoft, интегрированных с Office Lens. Распознаёт: снимки камеры. Сохраняет: DOCX, PPTX, PDF.

- Настольная программа Free OCR to Word распознаёт выбранные пользователем изображения, извлекая из них чистый текст без форматирования. Его можно копировать в буфер обмена, сохранять в формате TXT или экспортировать в Word. Распознаёт: JPG, TIF, BMP, GIF, PNG, EMF, WMF, JPE, ICO, JFIF, PCX, PSD, PCD, TGA и другие форматы. Сохраняет: DOC, DOCX, TXT. Пример работы данной программы отображен на рисунке 10.

- Онлайн-сервис FineReader, который конвертирует не только тексты, но и таблицы. Увы, бесплатные возможности FineReader Online ограничены. После регистрации вам позволят распознать без оплаты всего 10 страниц. Зато каждый месяц будут начислять ещё по пять страниц в качестве бонуса. Поэтому сервис больше подойдёт тем, кто не нуждается в услугах распознавания слишком часто. Распознаёт: JPG, TIF, BMP, PNG, PCX, DCX, PDF (не защищённые паролем). Сохраняет: DOC, DOCX, XLS, XLSX, ODT, TXT, RTF, PDF, PDF/A.

- Сайт marathon-photo.ru – сервис для размещения фотографий со спортивных мероприятий по бегу, триатлону, велоспорту и лыжному спорту. Как правило фотографий с таких мероприятий получается очень и очень много,

пролистывать тысячи фотографий участникам в надежде найти 3-4 своих фотографии совсем нецелесообразно. Так что на этом сайте внедрен алгоритм распознавания номеров на фото, расставление номеров в качестве тэгов для каждой фотографии и реализация поиска по номеру участника или фамилии.

– Распознавание государственного номера автотранспортного средства. Внедрение алгоритмов распознавания в эту сферу необходимо для того, чтобы контроль соблюдения правил дорожного движения мог стать автоматическим и постоянным. Как правило, эти алгоритмы внедряются непосредственно в камеры, фиксирующие нарушения, и не требуют обратных действий пользователя. Также эти функции с успехом внедряются и в обычные «пользовательские» видеорегистраторы. На рисунке 11 отображен пример такого программного обеспечения.

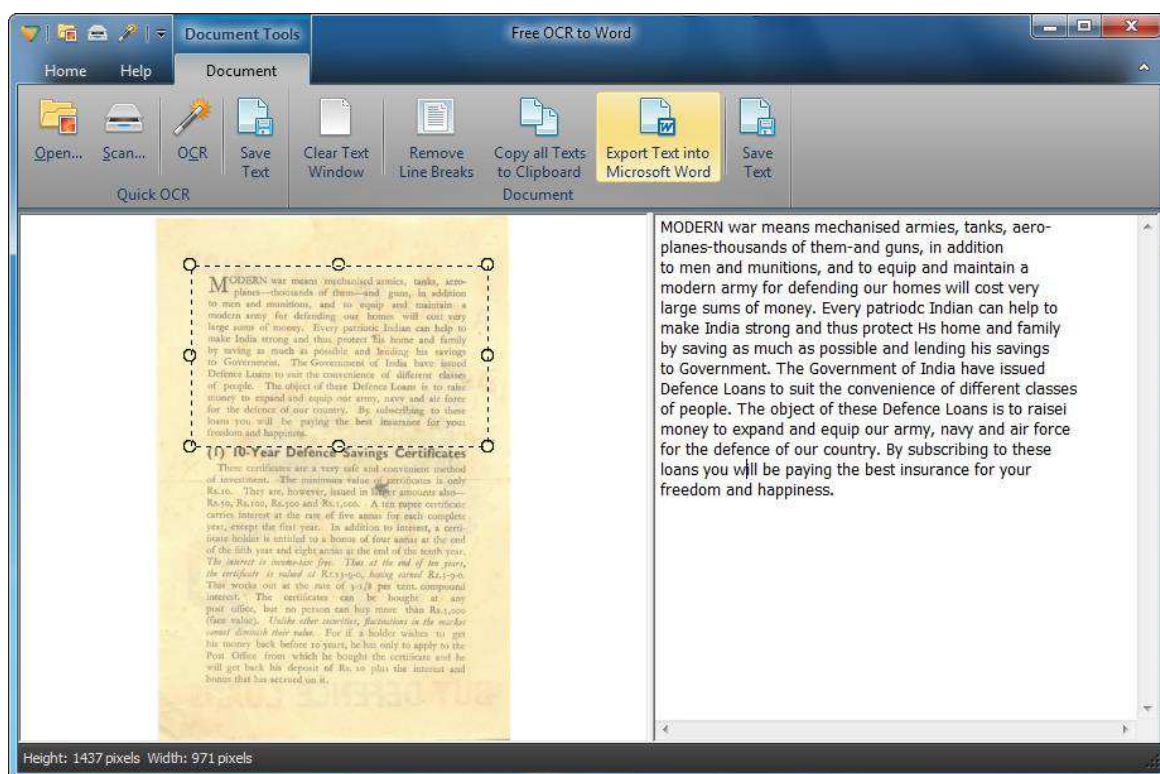


Рисунок 10 – Пример работы программы Free OCR to Word





Рисунок 11 – Пример распознавания номера автомобиля

#### 1.4.4 Распознавание лиц

Ещё 8-12 лет назад только начинали появляться цифровые фотоаппараты, которые выделяли квадратиком человеческое лицо, автоматически меняли настройки в связи с этим обнаружением. Как же обстоит дело с распознаванием лиц сейчас? Сейчас алгоритмы и технологии позволяют не просто найти лицо на изображении, но и с очень высокой точностью установить личность человека на изображении. Где же это реализовано?

– FaceID - это сканер 3D формы лица, который был разработан компанией Apple. Установлен в смартфоне iPhone X, используется для покупок с помощью платёжной системой Apple Pay, просмотра паролей в приложении «Настройки» для iOS, покупок в iTunes Store, App Store, iBooks, разблокировки сторонних приложений, например менеджера паролей 1Password и банк клиента Сбербанк Онлайн. Как заявляет компания Apple, зашифрованная биометрическая информация хранится на самом устройстве, в процессоре iPhone X - A11 Bionic, и не может быть отправлена куда-либо. По словам Apple, это математический образ отсканированного лица, но не само изображение.

– Open Garage: перевод денег по фотографии в мобильном приложении банка «Открытие.Переводы». Вместо того чтобы вбивать номер карты или телефона, достаточно просто сфотографировать человека, которому нужно сделать перевод. Система распознавания лиц сравнит фото с эталонным (делается, когда банк выдает карту) и подскажет имя и фамилию. Останется только выбрать карту и ввести сумму. Что особенно важно, клиенты сторонних банков также могут использовать эту функцию для переводов клиентам «Открытия» — отправитель переводов может пользоваться картой любого российского банка. На рисунке 12 отображена работа данной функции.

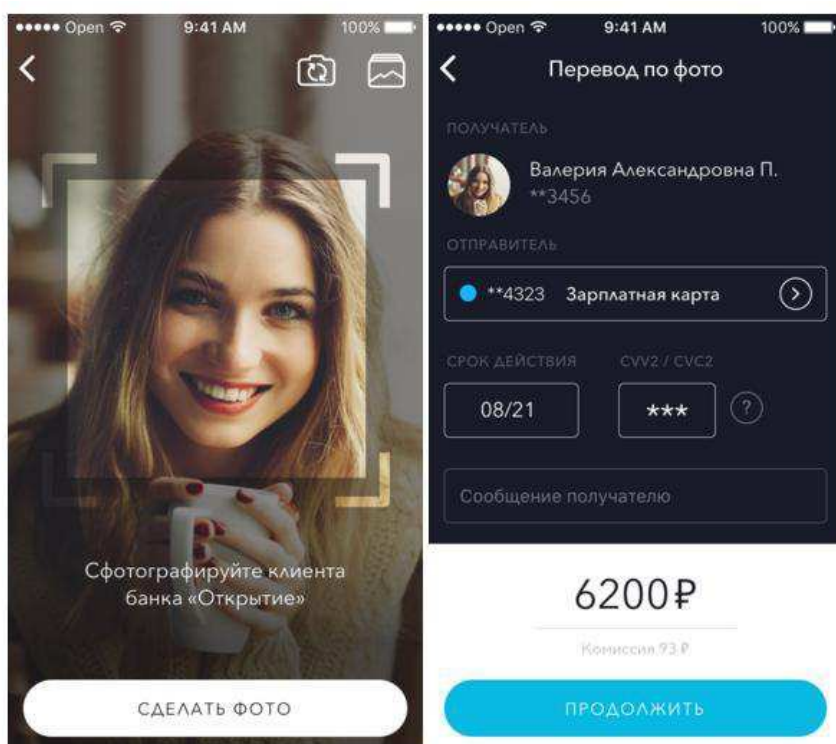


Рисунок 12 – Распознавание лица в приложении «Открытие. Переводы»

– Система идентификации лиц для безопасности. Технология позволяет через камеры проверять до 20 человек в секунду. Технологию начали внедрять только в 2018 году в Москве, но она уже имеет свои плоды: система обнаружила в потоке людей мужчину, находившегося в розыске и полиция

смогла его задержать. В столице сейчас насчитывается несколько тысяч уличных объектов, подключенных к системе идентификации лиц.

## **1.5 Выводы**

Обзор существующих программ, сервисов, сайтов для распознавания изображений или использующих методы распознавания для оптимизации целевой функции показал, что машинное зрение и обучение внедряется во всё большие части нашей жизни. Данная сфера прогрессирует особенно сильно последние годы. Именно 2017-2018 годы становятся запуском таких крупных проектов, как FaceID, Google Lens, системы идентификации в банках. Как правило, существующие аналоги привязаны к определенной предметной области и пользователю доступны алгоритмы распознавания, но не обучения системы. Хотя и обучение системы может стать очень полезным для более точной классификации объектов необходимой пользователю предметной области.

На основе вышеизложенных данных о методах и алгоритмах для распознавания изображения необходимо сделать выбор методов, которые наиболее удовлетворяют ожидаемому результату на каждом этапе.

Необходимо сделать преобразование изображения в интегральный вид для увеличения скорости работы алгоритмов обучения, сравнения с шаблоном и возможности использования классов и методов библиотеки OpenCV. На этом этапе будет использован алгоритм, наиболее близкий к описанному выше методу Виолы-Джонса.

Непосредственно распознавание изображения будет происходить путем сравнения с шаблоном, сформированном на этапе обучения. В процессе поиска осуществляется проход "бегущим окном", имеющим размеры шаблона, по изображению и сравнение описания части исходного изображения, покрываемого окном, и шаблона. Сопоставление с шаблоном подразумевает сравнение описания тестового и шаблонного изображений по некоторой выбранной метрике. Результатом такого сопоставления является мера сходства.

Считается, что если эта мера больше некоторого порога, то тестовое изображение – это изображение объекта.

Классификация объектов реализуется через алгоритм случайного леса решений, заключающийся в использовании комитета (ансамбля) решающих деревьев. Основная идея заключается в использовании большого ансамбля решающих деревьев, каждое из которых само по себе даёт очень невысокое качество классификации, но за счёт их большого количества результат получается хорошим.

## **2 Проектирование программного обеспечения**

### **2.1 Постановка задачи**

Требуется создание программного обеспечения, использование которого позволит обучать распознаванию и классифицировать объекты, считанные с изображения.

Требования к функциональности:

- обучение системы классам объектов;
- распознавание объекта на изображении;
- классификация объекта с изображения;
- тестирование точности классификации.

Требования к интерфейсу:

- функциональность (соответствие задачам пользователя);
- понятность и логичность;
- быстрое освоение приложения;
- реализация посредством WindowsForms.

Требования к программной платформе:

- 64-разрядная операционная система Windows 7, 8, 10

### **2.2 Инструментальные средства разработки**

Microsoft Visual Studio — линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms. Используемая для разработки версия – Microsoft Visual Studio 2017 Community.

.NET Framework — программная платформа, выпущенная компанией Microsoft в 2002 году. Основой платформы является общезыковая среда исполнения Common Language Runtime (CLR), которая подходит для разных языков программирования. Функциональные возможности CLR доступны в любых языках программирования, использующих эту среду. Используемый язык программирования – объектно-ориентированный C#. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

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. OpenCVSharp является кросс-платформенным .Net дополнением для библиотеки OpenCV для обработки изображений. Установка данной библиотеки выполняется через встроенный в Visual Studio менеджер пакетов NuGet.

## **2.3 Логическая структура**

Программное обеспечение содержит три основных блока: обучение системы, распознавание объектов и тестирование. Особенность программы состоит в её независимости от предметной области. Пользователю доступен модуль обучения программы абсолютно любым классам объектов, которые необходимы конкретному пользователю для конкретной деятельности. И программа, соответственно будет классифицировать поступающие изображения на основе известных ей классов объектов. На рисунке 13 отображена Use-case диаграмма использования программного обеспечения.

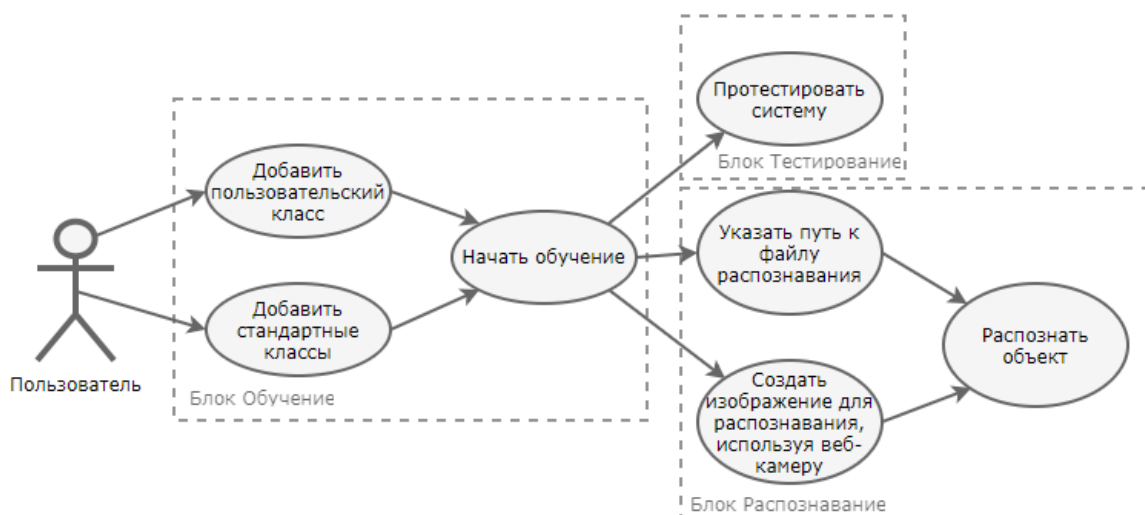


Рисунок 13 - Use-case диаграмма использования программного обеспечения

В блоке обучение системы пользователь может:

- а) указать наименование класса объекта;
- б) загрузить исходные данные одним из способов:
  - 1) использовать стандартный набор классов;
  - 2) указать путь к папке с выборкой изображений соответствующих указанному классу;
- в) просмотреть результаты обучения.

В блоке распознавание объектов пользователь может указать изображение для распознавания одним из способов:

- указав путь к файлу;
- используя веб-камеру.

После обработки полученного изображения программа должна сообщить пользователю ответ, на что из известных ей объектов наиболее похож объект с загруженного изображения. Таким образом, благодаря внедренному алгоритму обучения, программа может быть использована для любой предметной области.

Блок тестирование подразумевает определение точности классификации объектов программой. Подразумевается:

- Пользователь собирает контрольную выборку;

- Пользователь указывает, к какому (уже известному программе) классу принадлежат изображения данной выборки;
- Программа классифицирует каждое изображение данной выборки;
- Программа подсчитывает, сколько процентов изображений классифицировано ей соответственно указанному пользователем значением;
- На основе этого результата пользователь может делать выводы о точности работы программы.

## **2.4 Архитектура программного обеспечения**

Структурно программа состоит из следующих основных модулей:

- Program.cs – главный модуль программы, точка входа в программу;
- Form1.cs – компонент WindowsForms, состоит из интерфейса программы и обработчиков событий;
- Model.cs – дополнительные сущности для удобства разработки, управления данными и оптимизации кода;
- Динамически подключенные модули библиотеки OpenCvSharp.

Каждый уровень имеет модуль имеет доступ только к нижестоящим. Это необходимо для возможности модифицировать уровни без избыточного влияния на работающие алгоритмы.



### 3 Описание работы программного обеспечения

#### 3.1 Используемые классы и методы

В программе используются классы и методы из подключенной библиотеки OpenCvSharp. Перечень используемых классов отображен в таблице 1, перечень используемых методов отображен в таблице 2.

Структурно программа состоит из трех основных модулей:

- Program.cs – главный модуль программы, точка входа в приложение;
- Form1.cs – компонент WindowsForms, состоит из интерфейса программы и обработчиков событий;
- RecognitionClass.cs – класс распознавания.

Таблица 1 – Используемые классы библиотеки OpenCvSharp

| Наименование                          | Описание  |
|---------------------------------------|---|
| OpenCvSharp.Mat                       | Контейнер изображения   |
| OpenCvSharp.KeyPoint[]                | Массив ключевых точек   |
| OpenCvSharp.XFeatures2D.SIFT          | Класс для извлечения особенностей изображения                                     |
| OpenCvSharp.XFeatures2D.SURF          | Класс для ускоренного извлечения особенностей изображения                         |
| OpenCvSharp.ML.RTrees                 | Алгоритм случайного леса решений  |
| OpenCvSharp.BOWKMeansTrainer          | Класс для обучения словаря дескрипторов с использованием алгоритмов кластеризации |
| OpenCvSharp.DescriptorMatcher         | Класс для сопоставления дескрипторов ключевых точек                               |
| OpenCvSharp.BOWImgDescriptorExtractor | Класс для вычисления дескриптора изображения                                      |
| OpenCvSharp.VideoCapture              | Класс для захвата видео   |

Таблица 2 – Используемые методы библиотеки OpenCvSharp

| Наименование   | Описание  |
|--|---|
| New Mat()  | Создает пустой экземпляр класса Mat   |
| New Mat(string filename)   | Создает экземпляр класса Mat и загружает в него изображение, находящееся по адресу filename     |
| Mat.PushBack(Mat m)  | Добавляет m в конец матрицы   |
| SURF.Create(double hessianThreshold)   | Создает экземпляр класса SURF с указанным гессианским порогом                                   |
| SURF.Detect(Mat img)   | Вычисляет ключевые точки для изображения img  |
| SIFT.Create()  | Создает пустой экземпляр класса SIFT  |
| SIFT.Compute(InputArray img, ref KeyPoint[] keyPoint, OutputArray descImg)                 | Вычисляет дескриптор  |
| new BOWKMeansTrainer(int clusterCount)   | Создает экземпляр класса BOWKMeansTrainer с количеством кластеров равной параметру clusterCount |
| BOWKMeansTrainer.Cluster(Mat descriptors)  | Создает словарь, состоящий из центров кластеров, на основе переданных дескрипторов descriptors  |
| DescriptorMatcher.Create("BruteForce")   | Создает сопоставитель дескрипторов типа BruteForce  |
| new BOWImgDescriptorExtractor (extractor, matcher)   | Создает экземпляр класса BOWImgDescriptorExtractor  |
| BOWImgDescriptorExtractor. SetVocabulary(Mat centr);                                       | Устанавливает визуальный словарь на основе словаря центров кластеров                            |
| BOWImgDescriptorExtractor.Compute2(Mat image, ref KeyPoint[] keypoints, Mat imgDescriptor) | Вычисляет дескриптор изображения, используя визуальный словарь                                  |
| RTrees.Create()  | Создает пустой экземпляр класса RTrees  |
| RTrees.Predict(Mat desc)   | Предполагает реакцию на предоставленные примеры   |

Класс RecognitionClass.cs введен для удобства разработки, оптимизации кода и структурирования данных. Включает в себя следующие свойства, поля и методы:

- public string Name – открытое свойство типа строка, позволяет устанавливать и получать имя класса;
- public List<Mat> Images – открытое свойство. Представляет собой список объектов типа OpenCvSharp.Mat. Позволяет устанавливать и получать список изображений интегрального вида, соответствующих данному классу;

– `public List<Keypoint[]> Keypoints` – открытое вычисляемое поле. Представляет собой список объектов типа `OpenCvSharp.Keypoint[]`. Позволяет получить список ключевых точек для каждого изображения этого класса.

– `Public List<Mat> Descriptors` – открытое вычисляемое поле. Представляет собой список объектов типа `OpenCvSharp.Mat`. Позволяет получить список дескрипторов для каждого изображения этого класса.

– `Keypoint[] CalcSURF(Mat img)` – приватная функция для вычисления ключевых точек. В качестве параметра принимает изображение в интегральном виде типа `OpenCvSharp.Mat`;

– `List<Keypoint[]> CalcSURF(List<Mat> imgs)` – приватная функция. В качестве параметра принимает список изображений в интегральном виде типа `OpenCvSharp.Mat`. Представляет собой перегрузку функции `Keypoint[] CalcSURF(Mat img)` для списка изображений.

– `Mat CalcDescriptor(Mat img, Keypoint[] keyPoint)` – приватная функция для вычисления дескрипторов. В качестве параметров принимает изображение в интегральном виде типа `OpenCvSharp.Mat` и массив ключевых точек типа `OpenCvSharp.Keypoint[]`.

– `List<Mat> CalcDescriptor(List<Mat> imgs, List<Keypoint[]> keyPoints)` – приватная функция. В качестве параметров принимает список изображений в интегральном виде типа `OpenCvSharp.Mat` и список массивов ключевых точек типа `OpenCvSharp.Keypoint[]`. Представляет собой перегрузку функции `Mat CalcDescriptor(Mat img, Keypoint[] keyPoint)` для списка изображений.

В модуле `Form1.cs` содержатся следующие поля, методы и функции:

– `public List<RecognitionClass> RecognitionClasses` – открытое поле, представляет собой список сущностей типа `RecognitionClass`. Содержит классы объектов, добавленные перед обучением;

– `public Mat trainClasses` – открытое поле, представляет собой изображение в интегральном виде типа `OpenCvSharp.Mat`. Содержит ответы (реакции) на тренировочные данные (дескрипторы изображений);

- `public RTrees tree` – открытое поле, представляет собой случайное дерево решений типа `OpenCvSharp.RTrees`. Необходимо для классификации объектов;
- `public Mat src` – открытое поле типа `OpenCvSharp.Mat`. Хранит интегральный вид изображения, загруженного для распознавания;
- `Mat Learn()` – функция, возвращающая словарь, состоящий из центров кластеров. Алгоритм данной функции отражен на рисунке 14;
- `Mat ImgDescExtractor(Mat centr)` – функция, возвращающая дескрипторы изображений, используя визуальный словарь `centr`. Алгоритм данной функции отражен на рисунке 15;
- `private void usualLearningButton_Click(object sender, EventArgs e)` – обработчик события кнопки «Добавить стандартные классы»;
- `private void startRecognitionButton_Click(object sender, EventArgs e)` – обработчик события кнопки «Распознать объект»;
- `private void addClassButton1_Click(object sender, EventArgs e)` – обработчик события кнопки «Добавить пользовательский класс»;
- `private void pathLearningButton_Click(object sender, EventArgs e)` – обработчик события кнопки «Обзор...»;
- `private void addClassButton2_Click(object sender, EventArgs e)` – обработчик события кнопки «Добавить класс»;
- `private void recognitionFilePathButton_Click(object sender, EventArgs e)` – обработчик события кнопки «Обзор...»;
- `private void useCamButton_Click(object sender, EventArgs e)` – обработчик события кнопки «Использовать веб-камеру»;
- `private void startLearningButton_Click(object sender, EventArgs e)` – обработчик события кнопки «Все классы добавлены, начать обучение»;

- `private void testPathFolderButton_Click(object sender, EventArgs e)` – обработчик события кнопки «Обзор...»;
- `private void startTestButton_Click(object sender, EventArgs e)` – обработчик события кнопки «Тестировать».

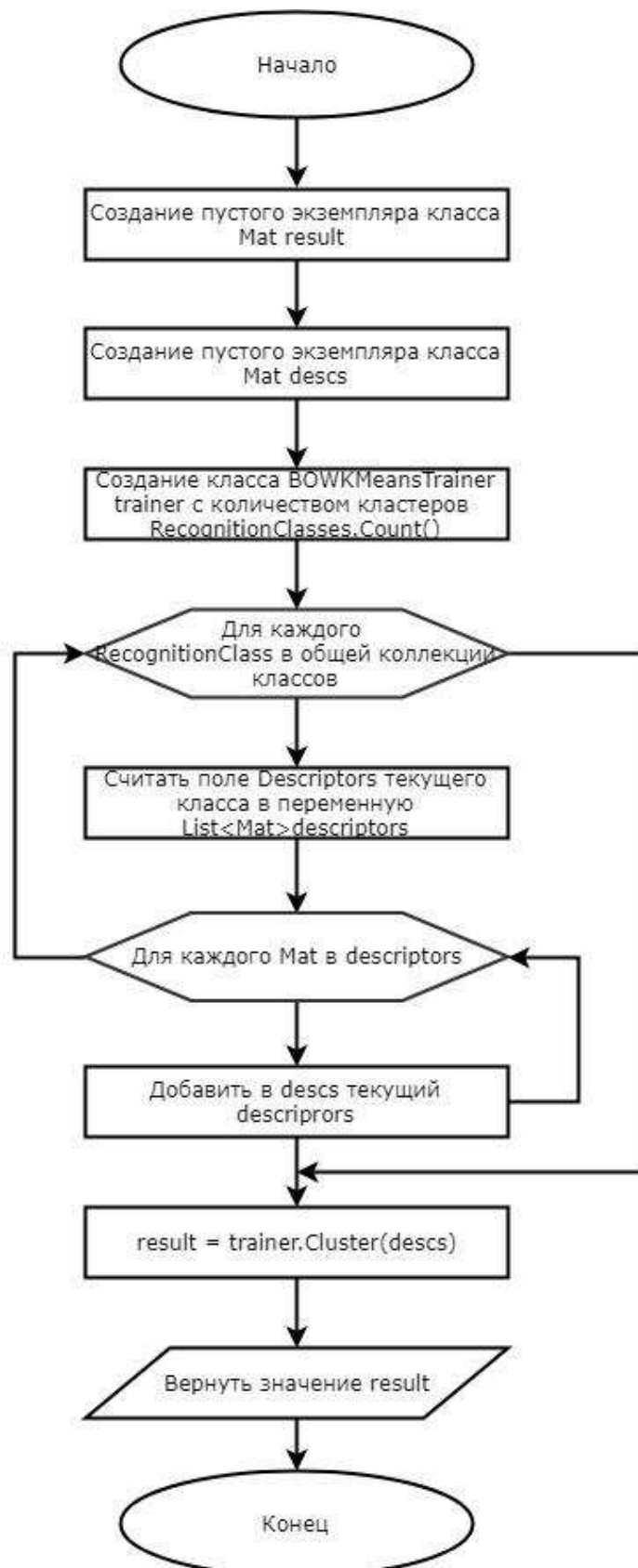


Рисунок 14 – Алгоритм функции `Learn()`

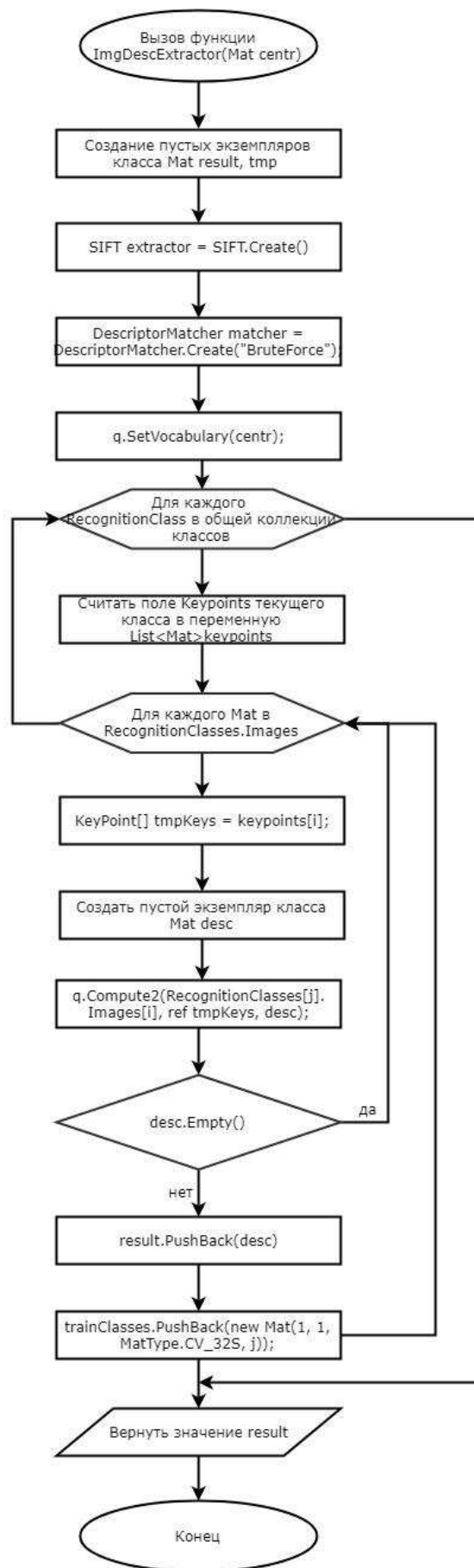


Рисунок 15 – Алгоритм функции ImgDescExtractor(Mat centr)

### **3.2 Функционал программного обеспечения**

Приложение предназначено для распознавания и классификации изображений и должно выполнять следующие функции:

- обучение системы классам объектов;
- распознавание объекта на изображении;
- классификация объекта с изображения;
- тестирование точности классификации.

Система состоит из трех основных блоков: обучение, распознавание объектов и тестирование. Особенность программы состоит в её независимости от предметной области. Пользователю доступен модуль обучения программы абсолютно любым классам объектов, которые необходимы конкретному пользователю для конкретной деятельности. И программа, соответственно будет классифицировать поступающие изображения на основе известных ей классов объектов. Таким образом увеличивается точность распознавания объектов, благодаря ограниченному количеству классов.

### **3.3 Алгоритм работы программы**

Точкой входа в программу является файл Program.cs. При запуске инициализируются все компоненты формы, загружается пользовательский интерфейс. Блоки «Тестирование» и «Распознавание» недоступны до того, как будет выполнено обучение системы.

При нажатии на кнопку «Добавить пользовательский класс» открывается панель «Добавление класса». Пользователю необходимо ввести имя класса и путь к папке с выборкой изображений для обучения. По нажатию на кнопку «Добавить класс» будет создан экземпляр класса, заполнен введенными данными и добавлен в коллекцию классов. Подробнее алгоритм обработчика события кнопки «Добавить класс» отображен на рисунке 16.



Пользователь может повторить данное действие неограниченное количество раз. В том числе, пользователь может добавить стандартные классы в коллекцию к пользовательским и наоборот. Алгоритм добавления стандартных классов отличается от добавления пользовательского лишь тем, что данные о именах классов и путях к папкам с обучающими выборками не указываются пользователем, а введены программно внутри кода.

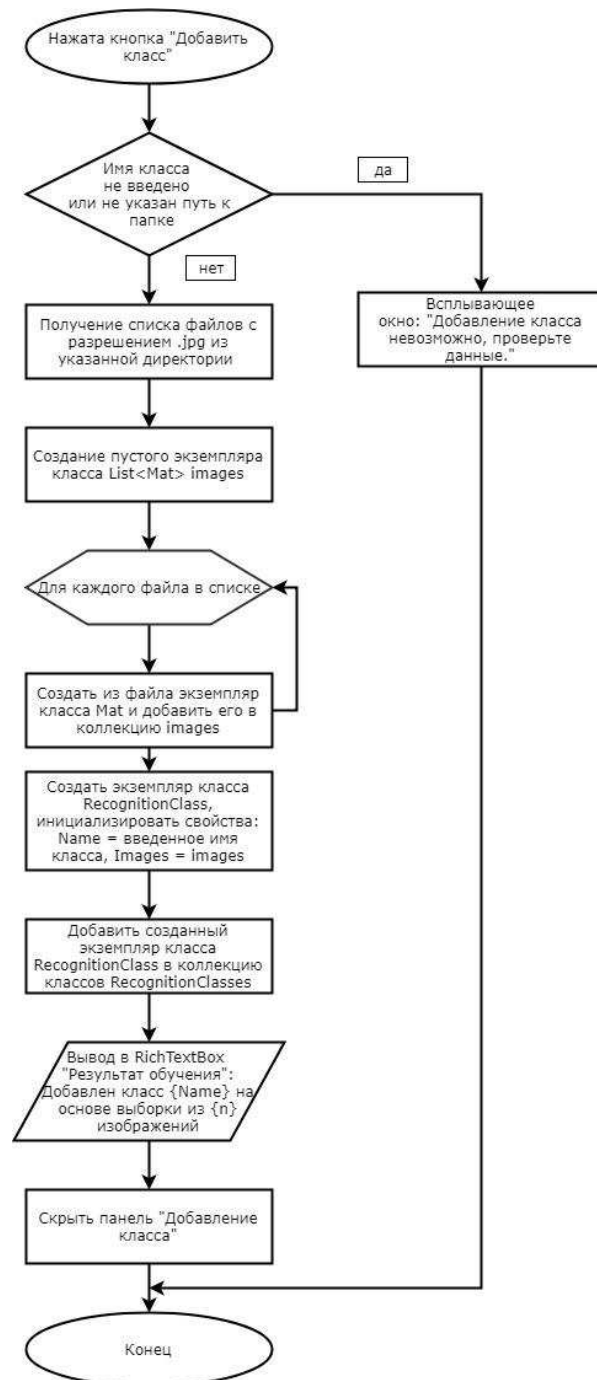


Рисунок 16 – Алгоритм обработчика события кнопки «Добавить класс»

После того, как пользователь добавил все необходимые классы, можно начинать обучение системы. Для этого необходимо нажать кнопку «Все классы добавлены, начать обучение». Сработает алгоритм обработчика события. В алгоритме создается словарь центров кластеров (пользовательская функция `Learn()`), на основе которого затем вычисляется дескриптор изображения и заполняется массив ответов (номер класса) на данный дескриптор (пользовательская функция `ImgDescExtractor(Mat centr)`). Устанавливается критерий остановки алгоритма, максимальная глубина деревьев решений. Для леса решений вызывается функция `Train` (из библиотеки `OpenCvSharp`), которая сопоставляет примеры изображений с массивом ответов, таким образом формируя представление об общих признаках каждого класса. В текстовое поле выводится сообщение об успешном завершении обучения, активируются панели «Тестирование» и «Обучение». В `ComboBox` «Имя класса» на панели «Тестирование» добавляются все имена классов из общей коллекции. Подробный алгоритм работы обработчика события нажатия на кнопку «Все классы добавлены, начать обучение» отображен на рисунке 17.

Следующее возможное действие пользователя – распознавание изображения. Для этого необходимо указать путь к изображению или сделать снимок с веб-камеры. Выбранное изображение загружается в компонент `PictureBox`. При нажатии на кнопку «Распознать объект» срабатывает обработчик события и выполняется алгоритм распознавания. Для экземпляра класса `BOWImgDescriptorExtractor` устанавливается словарь, вычисленный ранее при обучении. Затем вычисляются ключевые точки и дескриптор загруженного изображения. Полученный дескриптор передается в функцию `Predict` обученного леса. Функция возвращает номер класса, которому наиболее соответствует загруженное описание изображения. Результат выводится на экран. Подробно алгоритм обработчика события нажатия кнопки «Распознать объект» отображен на рисунке 18.

В блоке «Тестирование» пользователь должен указать имя класса и путь к выборке изображений для теста. После нажатия кнопки «Тестировать» сработает обработчик события и начнется выполнения алгоритма. В целом работает тот же алгоритм распознавания, что и на кнопке «Распознать объект» с той разницей, что распознается много изображений и результат не выводится на экран, а сохраняется в список результатов. Когда все изображения пройдены, по простой формуле рассчитывается процент ошибок и на экран выводится результат тестирования. Подробнее алгоритм представлен на рисунке 19.

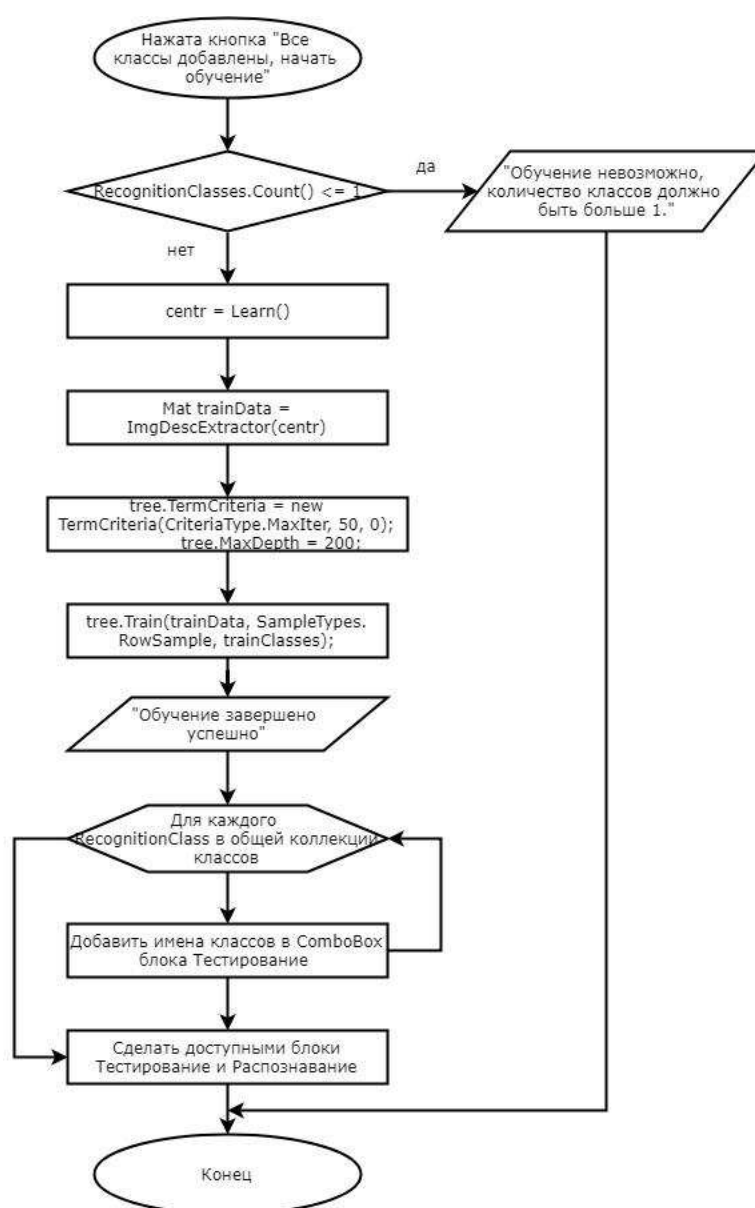


Рисунок 17 – Алгоритм обработчика события кнопки «Все классы добавлены, начать обучение»

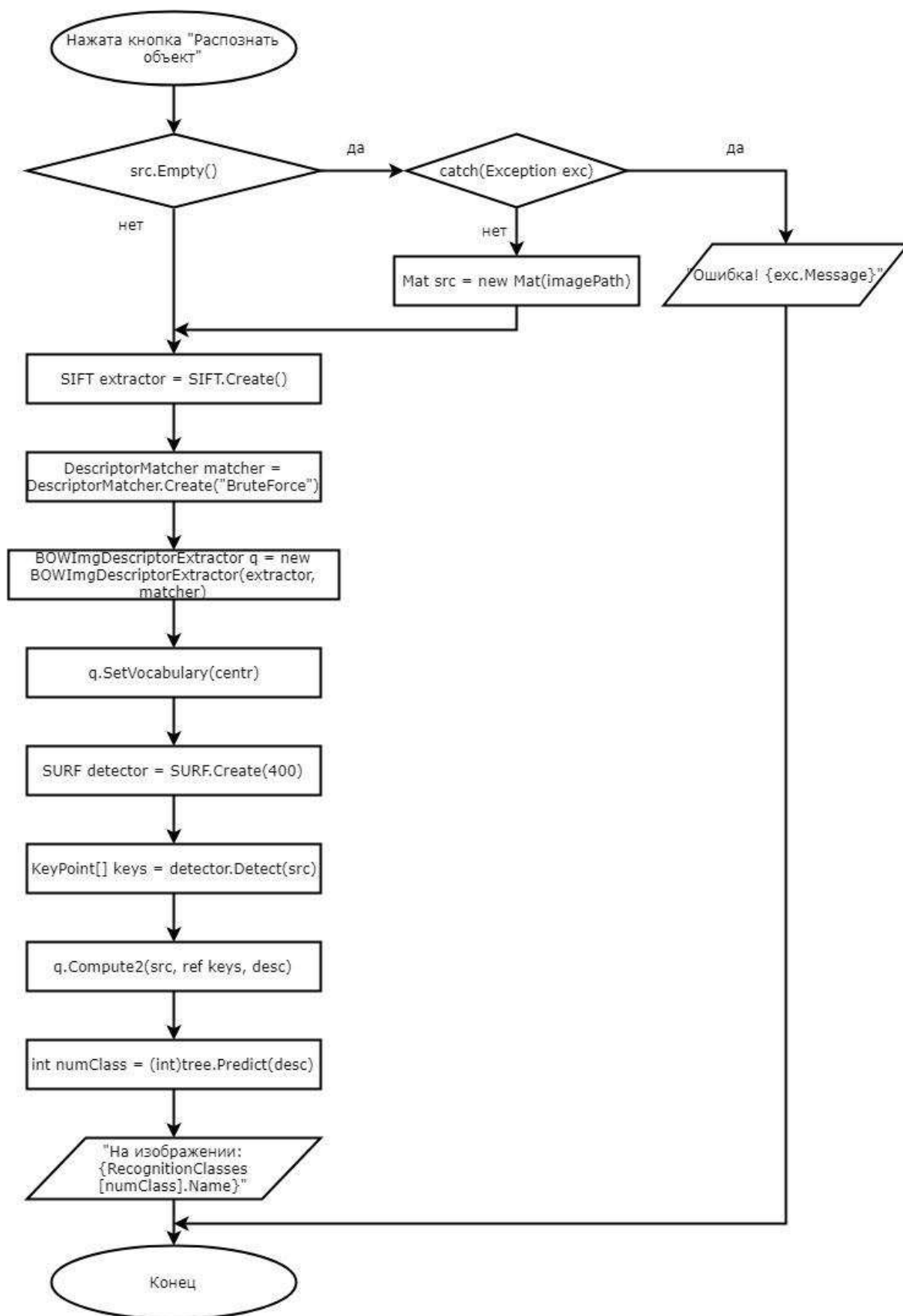


Рисунок 18 – Алгоритм обработчика события кнопки «Распознать объект»

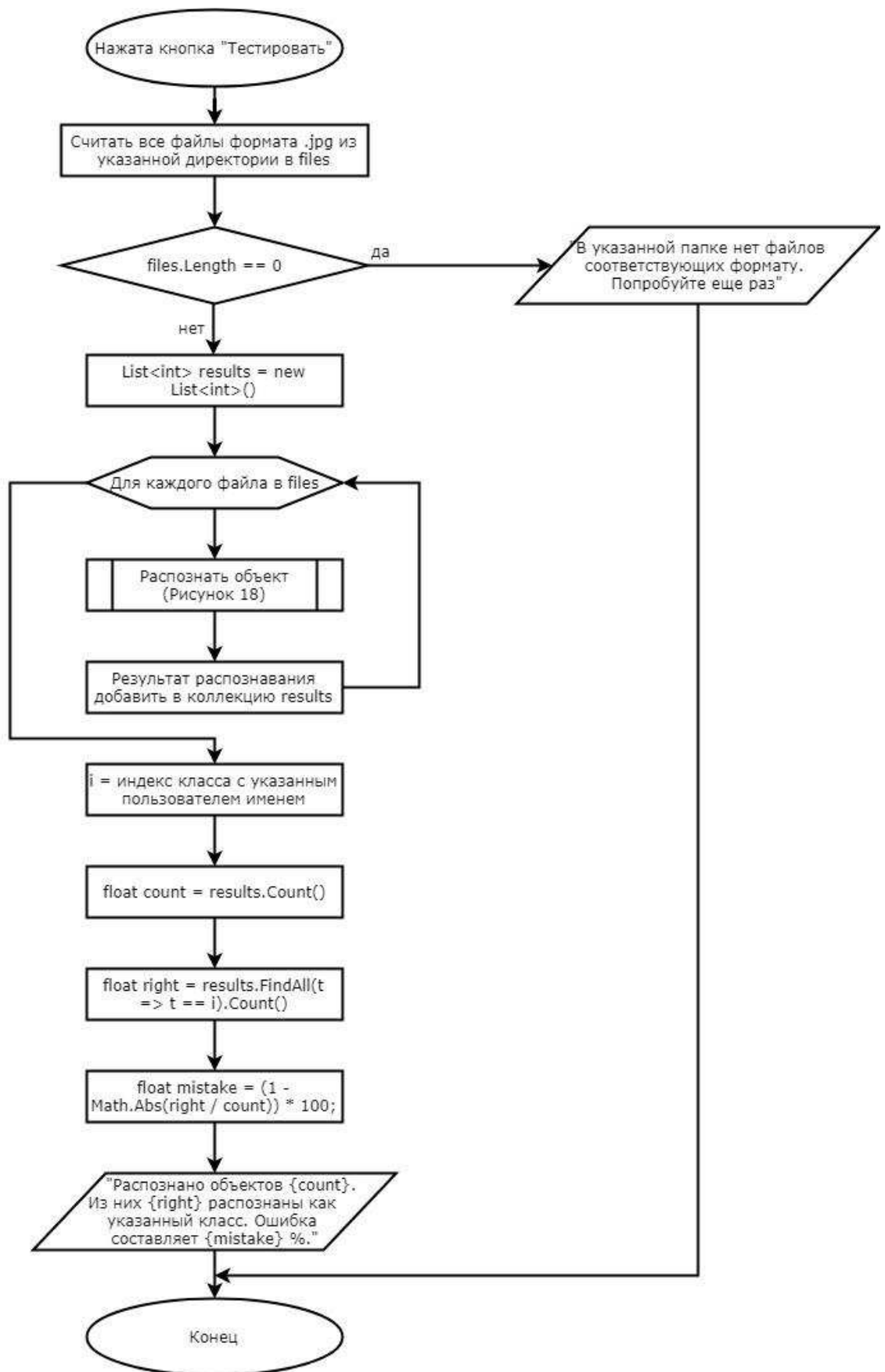


Рисунок 19 – Алгоритм обработчика события кнопки «Тестировать»

### 3.4 Работа с программным обеспечением

На рисунке 20 отображено окно программы при запуске. Главное окно поделено на 3 блока, используя компонент GroupBox:

- Обучение системы;
- Распознавание изображения;
- Тестирование.

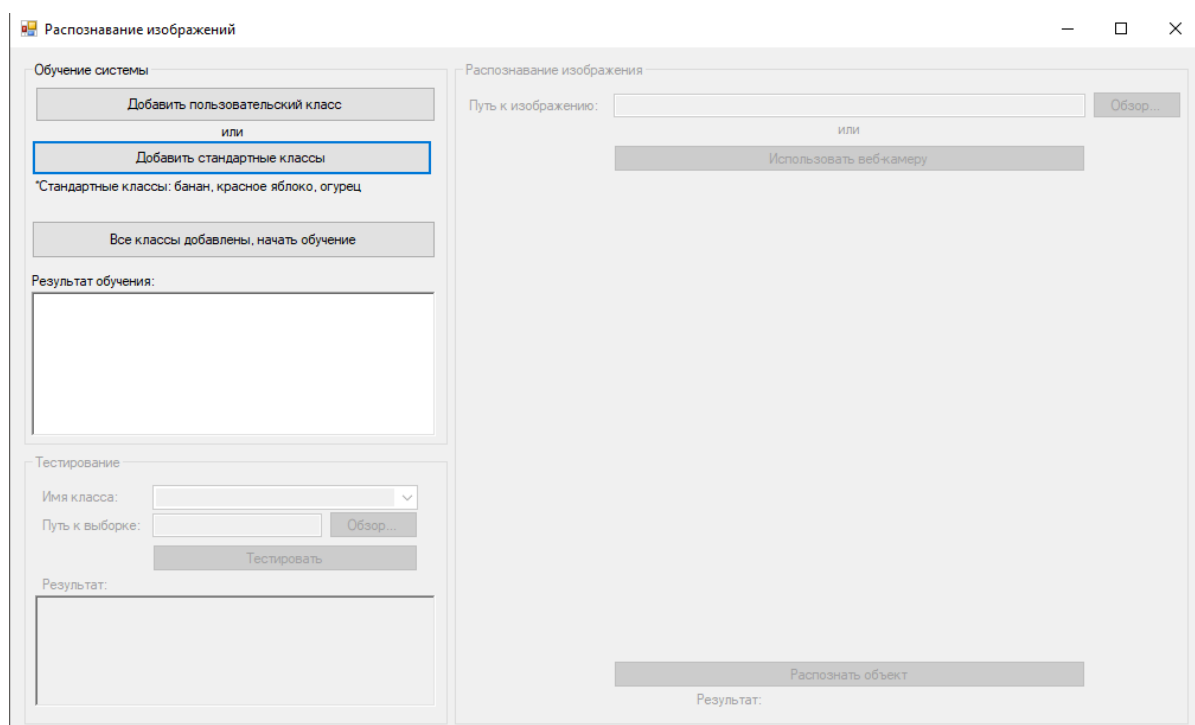


Рисунок 20 – Окно программы при запуске

Обучение системы содержит:

- Кнопку «Добавить пользовательский класс» для открытия панели «Добавление класса»;
- Панель «Добавление класса»;
- Кнопку «Добавить стандартные классы»;
- Кнопку «Все классы добавлены, начать обучение» для старта алгоритма обучения системы;
- Компонент RichTextBox для вывода результатов.

Распознавание изображения содержит:

- Компонент TextBox для ввода пути к изображению;
- Кнопку «Обзор...» для выбора файла изображения с использованием компонента OpenFileDialog;
- Кнопку «Использовать веб-камеру» для ввода изображения через веб-камеру;
- Компонент PictureBox для отображения загруженного изображения;
- Кнопку «Распознать объект» для старта алгоритма распознавания;
- Компонент Label для вывод результата.

Стандартные классы включают в себя объекты: банан, красное яблоко, огурец. При нажатии кнопки «Добавить пользовательский класс» открывается панель ввода (отображено на рисунке 21). Пользователь должен ввести имя класса и путь к папке с выборкой изображений для обучения (вручную или используя кнопку «Обзор...»). После подтверждения добавления класса панель исчезает, программа сообщает об успешном добавлении класса и можно добавить новые классы. На рисунке 22 отображена заполненная панель «Обучение системы», после добавления классов.

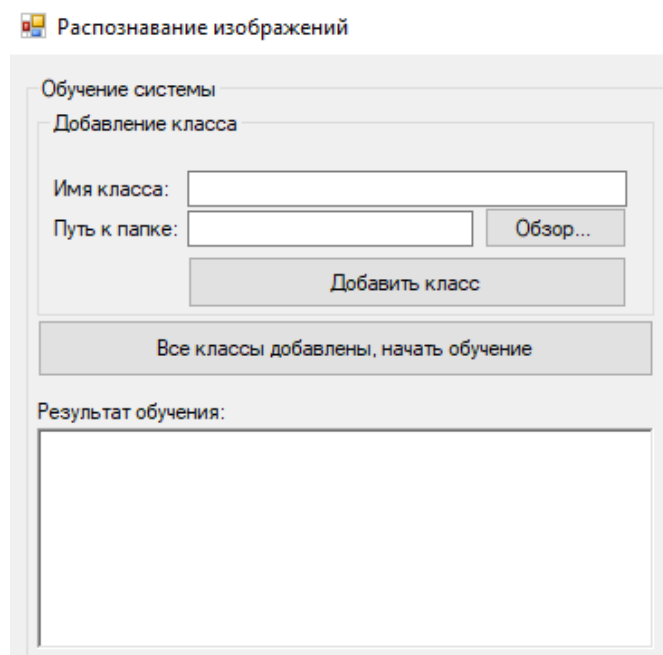


Рисунок 21 – Панель добавления класса

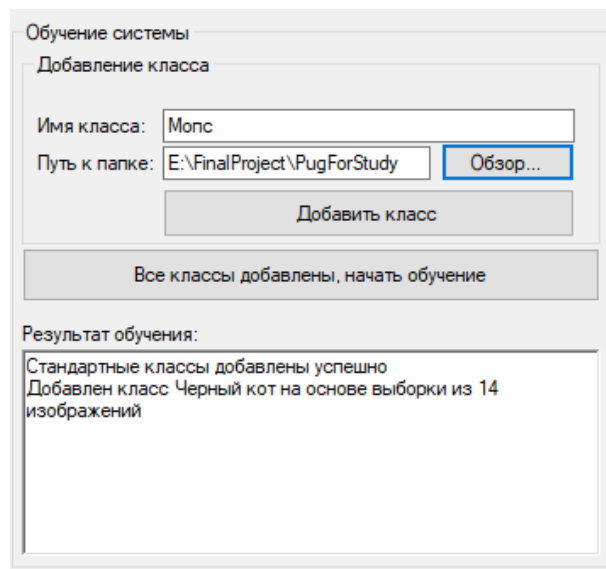


Рисунок 22 – Заполненная панель «Обучение системы»

На панели «Распознавание изображения» пользователь может ввести путь к изображению вручную, или используя кнопку «Обзор...». Также пользователь может нажать кнопку «Использовать веб-камеру» и произойдет захват изображения с веб-камеры. Полученное изображение будет загружено в компонент PictureBox (отображено на рисунке 23).



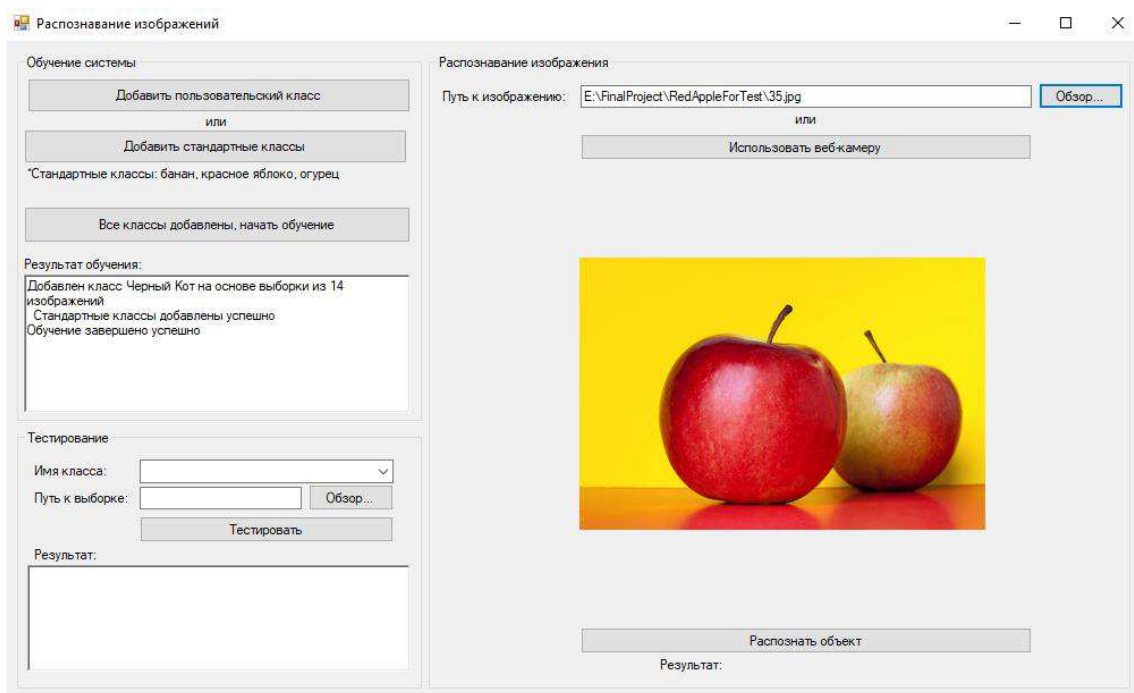


Рисунок 23 – Загрузка изображения

После нажатия кнопки «Распознать объект» программа выдаст результат обработки изображения, присвоив ему значение одного из изученных классов. Результат работы программы отображен на рисунке 24.

В блоке тестирования пользователь выбирает из выпадающего списка (компонент ComboBox) имя класса, известного программе и указывает путь к выборке для тестирования (введя путь к папке вручную или используя кнопку «Обзор...»). После нажатия кнопки «Тестировать» программа выдаст результат (отображено на рисунке 25).

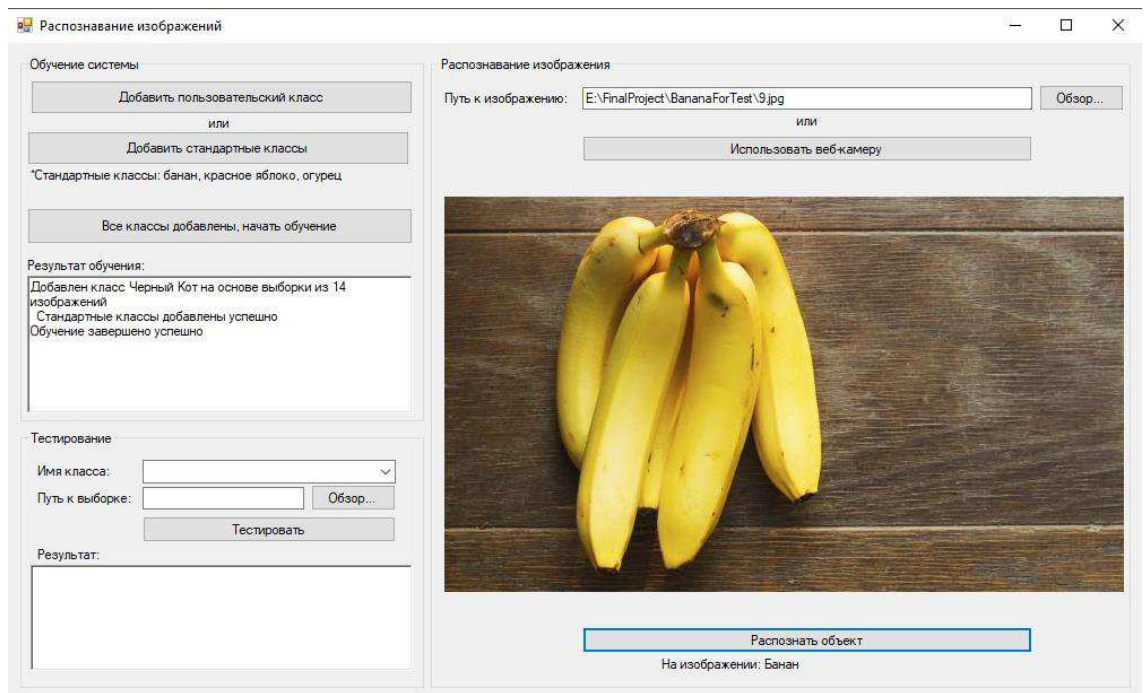


Рисунок 24 – Результат работы программы

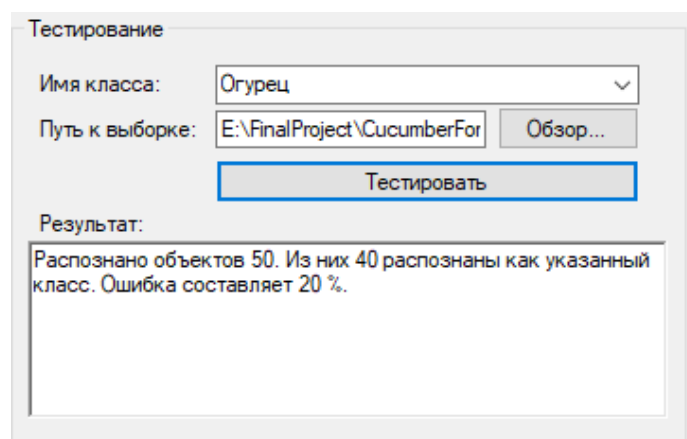


Рисунок 25 – Результат работы блока «Тестирование»

## ЗАКЛЮЧЕНИЕ

В результате выпускной квалификационной работы было реализовано программное обеспечение для распознавания объектов на изображении, изучены методы обработки и распознавания изображений, алгоритмы обучения системы, изучения языка программирования С# и использование библиотеки OpenCvSharp. Все задачи ВКР выполнены. Приложение соответствует требованиям и готово для установки на устройства пользователей.

В дальнейшем возможно расширить функционал программного обеспечения следующими функциями:

- Организация базы данных для хранения классов объектов, обученных компонентов RTrees, тренировочных данных и реакций;
- Возможность выбора алгоритмов обучения, распознавания, классификации объекта и сравнения их точности;
- Алгоритм и визуализация распознавания положения объекта на изображении;
- Ввод пользователем положения объекта на изображении.

В ходе выполнения получены знания о методах распознавания объектов, способах обучения машинному зрению, алгоритмах классификации и принятия решений. Также получен опыт работы с подключением сторонних библиотек в среду Visual Studio 2017, в частности взаимодействие с библиотекой OpenCvSharp, классы и методы которой активно использовались при разработке программного обеспечения.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Draw.io – Онлайн-сервис для создания диаграмм. [Электронный ресурс] : Create New Diagram – Режим доступа: <https://www.draw.io/>
2. Блог об информационных технологиях [Электронный ресурс] : Системы распознавания образов (идентификации) – Режим доступа: [http://www.codenet.ru/progr/alg/ai/htm/gl3\\_1.php](http://www.codenet.ru/progr/alg/ai/htm/gl3_1.php)
3. Веб-сервис для хостинга IT-проектов и их совместной разработки GitHub [Электронный ресурс] : .NET Framework wrapper for OpenCV – Режим доступа: <https://github.com/shimat/opencvsharp>
4. Гонсалес, Р. Мир цифровой обработки: цифровая обработка изображений // Р. Гонсалес, Р. Вудс; перевод с английского под редакцией П. А. Чочиа – Москва: Техносфера, 2005. – 1072 с.
5. Дружков, П. Н. Разработка мультимедийных приложений с использованием библиотек OpenCV и IPP : учебный курс / П. Н. Дружков. – Нижний Новгород : Нижегородский государственный университет им. Н.И. Лобачевского, 2013. – 41 с.
6. Козинев, Е. А. Подходы к оптимизации и распараллеливанию вычислений в задаче детектирования объектов разных классов на изображении / Е. А. Козинев, В. Д. Кустикова, И. Б. Мееров, Половинкин А. Н., Сиднев А. А. // Вестник Южно-Уральского государственного университета. Серия: Вычислительная математика и информатика, – 2012. – 15 с.
7. Красильников, Н. Н. Цифровая обработка 2D- и 3D-изображений: учеб. пособие // Н. Н. Красильников; – Санкт-Петербург: БХВ-Петербург, 2011. – 608 с.
8. Официальный сайт библиотеки OpenCV [Электронный ресурс] : OpenCVTutorials – Режим доступа: [https://docs.opencv.org/master/d9/df8/tutorial\\_root.html](https://docs.opencv.org/master/d9/df8/tutorial_root.html)

9. Распознавание образов для программистов [Электронный ресурс] : Детектирование объектов с помощью особенностей в OpenCV: FREAK. Детектирование множества объектов – Режим доступа: <http://recog.ru/blog/opencv/200.html>
10. Распознавание образов для программистов [Электронный ресурс] : Основы работы с матрицами в OpenCV – Режим доступа: <http://recog.ru/blog/opencv/206.html>
11. РобоКрафт – сообщество любителей робототехники, электроники и программирования. [Электронный ресурс] : OpenCV шаг за шагом – Режим доступа: <http://robocraft.ru/blog/computervision/264.html>
12. Свободная энциклопедия Википедия [Электронный ресурс] : Random forest – Режим доступа: [https://ru.wikipedia.org/wiki/Random\\_forest](https://ru.wikipedia.org/wiki/Random_forest)
13. Свободная энциклопедия Википедия [Электронный ресурс] : Метод Виолы-Джонса – Режим доступа: [https://ru.wikipedia.org/wiki/Метод\\_Виолы\\_—\\_Джонса](https://ru.wikipedia.org/wiki/Метод_Виолы_—_Джонса)
14. СТО 4.2-07-2014 Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной деятельности. – Введ. 9.01.2014. – Красноярск : ИПК СФУ, 2014. – 60 с.
15. Хабрахабр – крупнейший ресурс для IT-специалистов. [Электронный ресурс] : Классификатор изображений – Режим доступа: <https://habr.com/company/dmlabs/blog/205842/>
16. Хабрахабр – крупнейший ресурс для IT-специалистов. [Электронный ресурс] : Самообучаемая программа от Disney Research для распознавания образов – Режим доступа: <https://habr.com/post/380363/>
17. Хабрахабр – крупнейший ресурс для IT-специалистов. [Электронный ресурс] : Пару слов о распознавании образов – Режим доступа: <https://habr.com/post/208090/>

18. Шапиро, Л. Компьютерное зрение // Л. Шапиро, Дж. Стокман; перевод с английского А. А. Богуславского под редакцией С. М. Соколовой – Москва: БИНОМ. Лаборатория знаний, 2006. – 752 с.

19. Шепелев, К. В. Комбинированный метод детектирования и классификации движущихся объектов в видеопоследовательности : дис. ... канд. техн. наук : 05.13.17 / Шепелев Кирилл Валерьевич. – Пенза, 2017. – 130 с.

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт Космических и Информационных Технологий  
институт  
Информационные системы  
кафедра

УТВЕРЖДАЮ

Заведующий кафедрой ИС

  
подпись

Л.С. Троценко  
инициалы, фамилия

« 13 » 06 2018 г.

**БАКАЛАВРСКАЯ РАБОТА**

09.03.02 Информационные системы и технологии


Разработка программного обеспечения для распознавания объектов на  
графическом изображении

Руководитель

  
13.06.2018  
подпись, дата

П. П. Дьячук  
инициалы, фамилия

Выпускник

  
13.06.2018  
подпись, дата

А. Г. Рыкова  
инициалы, фамилия

Нормоконтролер

  
13.06.18  
подпись, дата

Ю.В. Шмагрис  
инициалы, фамилия

Красноярск 2018