****

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ МОСКОВСКОЙ ОБЛАСТИ**

**Государственное бюджетное профессиональное образовательное**

**учреждение Московской области**

**«Люберецкий техникум имени Героя Советского Союза,**

**лётчика-космонавта Ю.А. Гагарина»**

|  |
| --- |
|  |
|  |
|  |
|  |

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**(ДИПЛОМНЫЙ ПРОЕКТ)**

На тему: Чтение и распознавание номерных знаков автомобилей с помощью нейросетей

по специальности: 09.02.07 Информационные системы и программирование»

|  |  |  |
| --- | --- | --- |
| Выполнил |  | Н.Р. Никитин |
| Руководитель |  | Ю.В. Жирнова |

ДОПУЩЕН К ЗАЩИТЕ

Заместитель директора по информационным технологиям

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / А.В.Капранова /

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022г.

**Содержание**

[ВВЕДЕНИЕ 3](#_Toc105863400)

[1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 5](#_Toc105863401)

[1.1 Анализ литературных источников 5](#_Toc105863402)

[1.2 Обзор систем автоматического распознавания номерных знаков 5](#_Toc105863403)

[1.3 Сравнительный анализ языков программирования 10](#_Toc105863404)

[1.4 Анализ существующих сред разработки для Python 20](#_Toc105863405)

[1.5 Анализ предметной области 22](#_Toc105863406)

[1.6 Исследование предметной области 23](#_Toc105863407)

[2. ПРАКТИЧЕСКАЯ ЧАСТЬ 25](#_Toc105863408)

[2.1. Разработка структуры 25](#_Toc105863409)

[2.2. Создание приложения 26](#_Toc105863410)

[Выводы 39](#_Toc105863411)

# ВВЕДЕНИЕ

В современном мире на ежедневной основе люди постоянно используют информационные системы, поскольку они очень помогают в жизни и значительно упрощают ее. Также объем информации, с которой людям приходится работать увеличивается из года в год.

Поэтому появляется необходимость в постоянном развитии новых направлений в информационных системах для того, чтобы минимизировать затраты времени, материальных, трудовых ресурсов в ходе своей деятельности и упростить процесс обработки информации. Вследствие этого развитие нейронных сетей имеет большую значимость, так как они могут работать полностью автономно и с высокой скоростью.

Нейронные сети являются одним из важнейших направлений в машинном обучении. Главное их преимущество – это способность к обучению. Обучение заключается в нахождении коэффициентов связей. Данное преимущество дает возможность легко адаптировать нейронную сеть под свои нужды и оптимизировать работу.

К примеру, системы, обученные на автоматическое распознавание автомобильных номеров востребованы в самых различных областях. К примеру, они применяются в работе автотранспортных предприятий, станций техобслуживания, автомобильных парковок. Подобные системы дают возможность следить за безопасностью на территориях госучреждений, институтов, школ и т.д. Также эти системы могут выполнять пропуск по заданному автомобильному номеру и контролировать количество транспортных средств на территории или свободных мест на парковке.

Целью данной выпускной квалификационной работы является разработка приложения для распознавания номерных знаков с использованием нейронных сетей.

Для достижения поставленной цели необходимо решение следующих задач:

1. Выполнить обзор задач, решаемых с помощью нейросетей
2. Провести исследование предметной области
3. Проанализировать существующие инструменты для распознавания номерных знаков автомобилей
4. Разработать структуру
5. Прочитать и распознать номерные знаки автомобиля с помощью нейросети
6. Провести тестирование

# 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1 Анализ литературных источников

Для доказательства актуальности выбранной темы дипломной работы был проведён литературный обзор, раскрывающий тему значимости и востребованности данного приложения.

В статье «История развития нейронных сетей» Горбачевской Е. Н., и Краснова С. С. оценивают перспективы нейронных сетей и приходят к выводам, что за нейронными сетями будущее, но они также говорят о том, что они будут работать в сочетании традиционного управления.

Фаустов К.И в своей статье «Нейронные сети: применение сегодня и перспективы развития» приходит к тому, что нейронные сети имеют огромную перспективу, и что стартапы в этой области будут стоить сотни миллиардов долларов уже через несколько лет. Также развитие нейронных сетей привнесет множество улучшений во многие сферы жизни человек, но вместе с этим грядет опасность сокращения большого количества рабочих мест, а порой и полной ликвидации целой профессии. Как пример, с развитием систем автоматического распознавания номеров отпадет необходимость в шлагбаумах и систем внеуличных парковок.

Videomir в статье «Распознавание автомобильных номеров» рассматривает где необходимы подобного рода приложения. В статье пришли к выводу, что такие приложения необходимы для препятствия проезда, к примеру, на частную территорию или платную парковку, также такого рода приложения необходимы для управления и контроля потока автомобилей.

В связи с этими статьями, можно убедиться в актуальности данной темы.

1.2 Обзор систем автоматического распознавания номерных знаков

Рассмотрим несколько приложений с похожим функционалом, такие как: «iPera EX-LPR», «CVS-Авто», «AutoTRASSIR», «НомерОк», «Macroscop-Авто».

iPera EX-LPR – клиентское приложение системы видеонаблюдения ExacqVision и предназначен для автоматического распознавания всех автомобильных номер и их регистрацию (рис. 1).

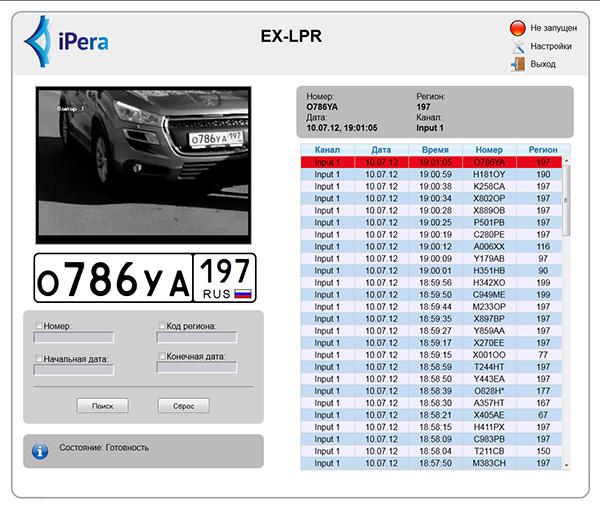


Рисунок 1. Интерфейс приложения ExacqVision

Преимущества:

* Вероятность распознавания номера 98%;
* Удобный и понятный интерфейс;
* Клиент-серверная архитектура;

Недостатки:

* Распознает только Российские, Белорусские, Казахстанские, Иорданские и ОАЭ номера автомобилей;

CVS-Авто - предназначена для автоматического распознавания и фиксации автомобильных номеров, попавших в поле зрения видеокамеры (рис. 2).

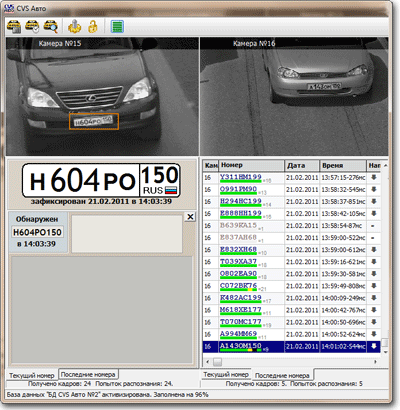


Рисунок 2. Интерфейс приложения CVS-Авто

Преимущества:

* Вероятность распознавания номера 92-98%;
* Распознает номера любых стран;
* Предусмотрена возможность ручной корректировки номера автомобиля;
* Можно настроить реакцию системы на отдельный, группу и тип номеров;

Недостатки:

* Высокая стоимость;

AutoTRASSIR - программа автоматического распознавания автомобильных государственных регистрационных знаков (рис. 3).

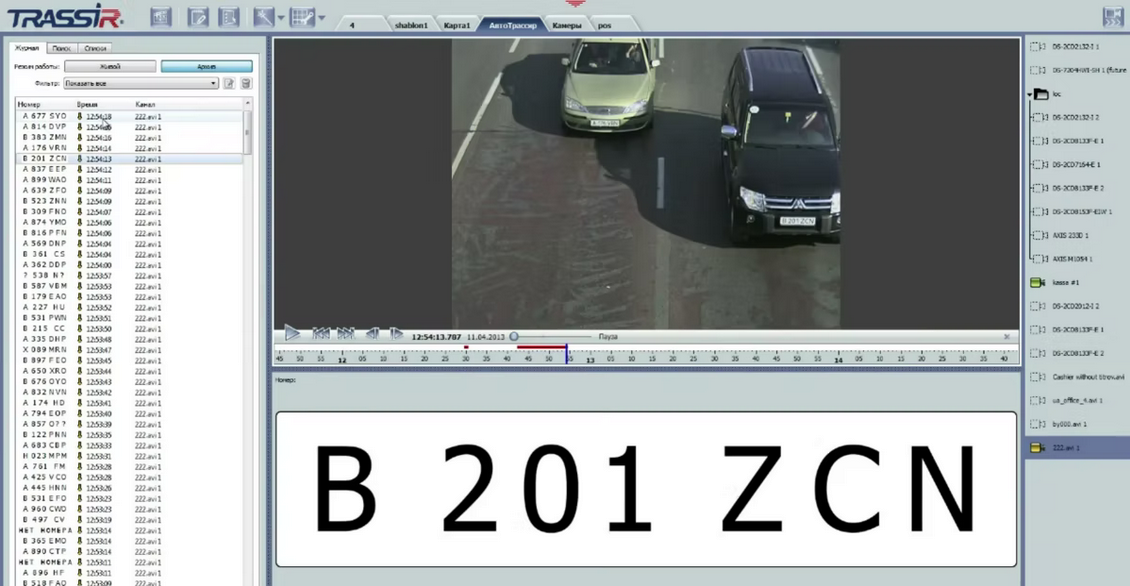


Рисунок 3. Интерфейс приложения AutoTRASSIR

Преимущества:

* Использует 3 нейросети для уменьшения количества ложных срабатываний;
* Распознает номера любых стран;
* Распознавание со сложных ракурсов;
* Гибкая настройка модуля распознавания;
* Возможность определять номер машин на скорости до 200 км/ч;

Недостатки:

* Высокая стоимость;
* Требуется высокое качество изображения/видео;

НомерОк - программно-аппаратный комплекс предназначен для распознавания автомобильных номеров и управления исполнительными устройствами (рис. 4).

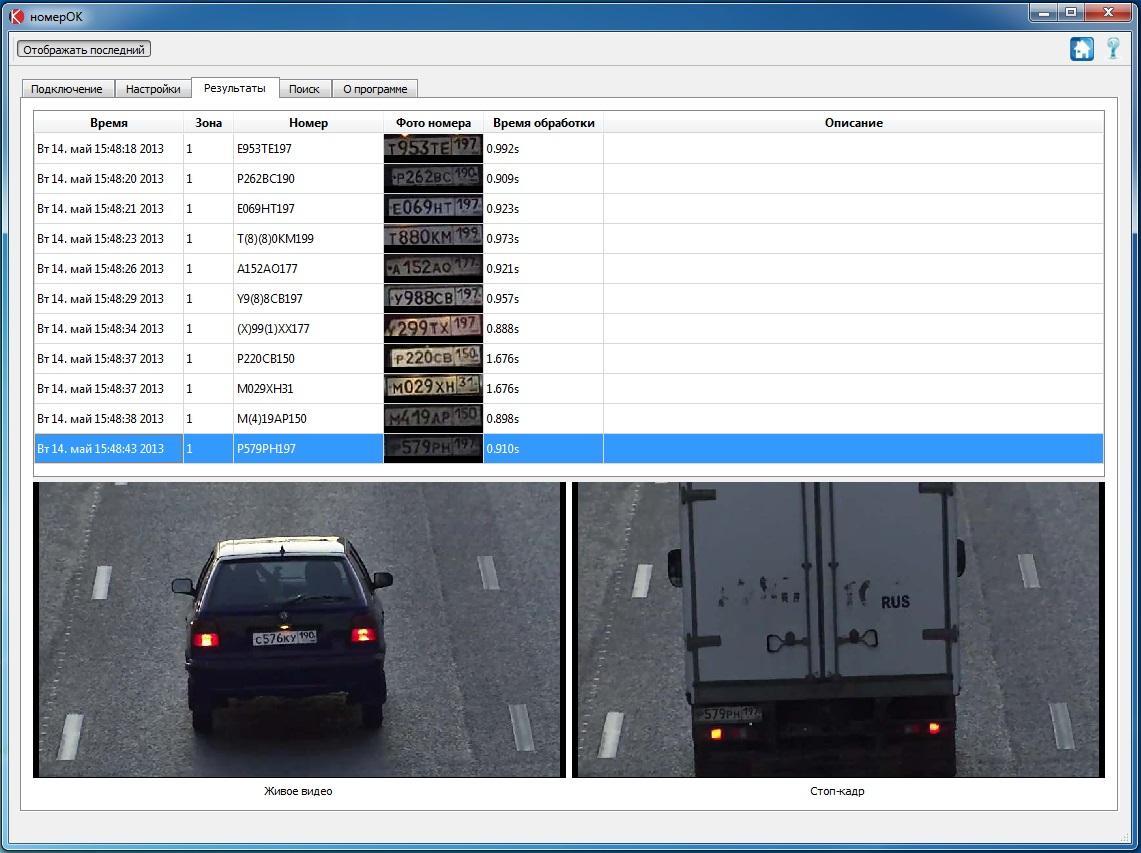


Рисунок 4. Интерфейс приложения НомерОк

Преимущества:

* Распознавание номерных знаков во всех регионах мира;
* Возможность определять номера машин на скорости до 240 км/ч;

Недостатки:

* Точность распознавания 95%;
* Высокая стоимость;

Macroscop-Авто - предназначен для распознавания регистрационных номеров движущихся автомобилей с сохранением в архиве информации о времени и дате распознавания, номере автомобиля, а также ссылки на соответствующий видеокадр (рис. 5).

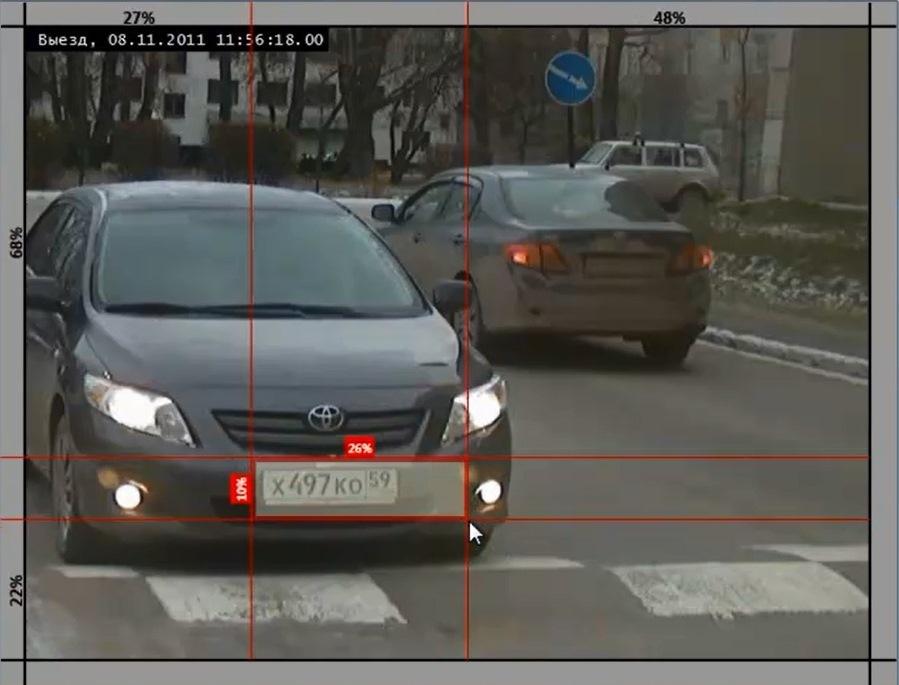


Рисунок 5. Интерфейс приложения Macroscop-Авто

Преимущества:

* Клиентское программное обеспечение для модуля распознавания есть двух видов - веб-интерфейс и программный клиент сервера базы данных.

Недостатки:

* Высокая стоимость;
* В случае некорректного распознавания сохраняет как новый номер;
* Может распознать только до скорости 150 км/ч;

1.3 Сравнительный анализ языков программирования

Для начала разработки приложения нам нужно выбрать на каком языке программирования он будет написан, ниже приведено сравнение таких языков для создания приложений как: Python, C#, Java, C++, PHP.

Python – это активно развивающийся язык сценариев, который используется для решения большого объема самых разнообразных проблем и задач. Python полезен при создании компьютерных и мобильных приложений, он используется при работе с большим объемом информации, при разработке веб-сайтов и других различных проектов, он используется в машинном обучении.

Если говорить про наиболее частое использование языка, то это такие сферы как: веб разработка, работа с данными, машинное обучение и нейронные сети, разработка ботов и написание скриптов под разные системы.

Преимущества:

1. Язык характеризуется логическим синтаксисом, что делает исходный код программ, написанных на Python, легким для чтения и понимания.

2. Еще одно преимущество этого языка программирования - его относительная легкость. Считается наиболее подходящим для начинающих специалистов: научиться разрабатывать простые программы можно уже через пару дней обучения.

3. Большое онлайн-сообщество. Если разработчик сталкивается с какими-либо вопросами и трудностями, он всегда может обратиться за советом к своим коллегам, что значительно ускоряет решение проблемы.

4. Гибкость и масштабируемость. Python позволяет разработчикам настраивать высокоуровневую логику приложения, упрощая расширение сложных приложений по мере необходимости.

5. Разработка на Python происходит быстрее, чем на большинстве других языков.

6. Python – это интерпретируемый язык программирования. Это означает, что перед запуском это просто текстовый файл. Таким образом, вы можете программировать практически на всех платформах.

Недостатки:

1. Скорость работы. На чистом Python будет сложно писать мощные проекты. Для этого требуется прибегать к помощи других языков.

2. Безопасность, обеспечиваемая моделью памяти Python, отменяет большинство возможных оптимизаций процессора.

3. Динамическая типизация – из-за динамической типизации Python потребляет больше ресурсов, чем мог бы, но это часто компенсируется внутренним кешированием.

4. Синтаксис – по мнению многих, синтаксис тоже минус, потому что, если переходить с другого языка программирования, синтаксис будет непривычным и немного странным, но это уже дело привычки.

C# (произносится как "си шарп") – прогрессивный объектно-ориентированный и весьма безопасный язык программирования. C# дозволяет разработчикам создавать много типов безопасных и надежных приложений, действующих в экосистеме .NET. C# относится к обширно знаменитому семейству языков C, и покажется отлично знакомым каждому, кто работал с C, C++, Java либо JavaScript.

C# – это объектно и компонентно-ориентированный язык программирования. C# дает языковые конструкции для непосредственной поддержки такой концепции работы. Вследствие этого C# подходит для создания и внедрения программных частей. С момента создания язык C# обогатился функциями для поддержки новых рабочих нагрузок и передовыми рекомендациями по разработке ПО.

Программное обеспечение C# исполняется в .NET, виртуальной системе выполнения, инициирующей общеязыковую среду выполнения (CLR) и комплект библиотек классов. Среда CLR – это исполнение общеязыковой инфраструктуры языка (CLI), являющейся интернациональным стандартом, от организации Майкрософт. CLI считается базой для создания сред исполнения и разработки, в которых языки и библиотеки прозрачно действуют друг с другом.

Исходный код, написанный на языке C# компилируется в промежуточный язык (IL), который отвечает спецификациям CLI. Код на языке IL и средства, в том числе растровые изображения и строчки, хранятся в сборке, чаще всего с расширением .dll. Ассемблирование содержит манифест с данными о типах, версии, языке и региональных параметрах для данной сборки.

При выполнении программы C# сборка загружается в среду CLR. Среда CLR выполняет JIT-компиляцию из кода на языке IL в инструкции машинного языка. Среда CLR также выполняет другие операции, например, автоматическую сборку мусора, обработку исключений и управление ресурсами. Код, выполняемый средой CLR, иногда называют "управляемым кодом", чтобы подчеркнуть отличия этого подхода от "неуправляемого кода", который сразу компилируется в машинный язык для определенной платформы.

Преимущества:

1. C# – это объектно-ориентированный, простой и в то же время мощный язык программирования, который позволяет разработчикам создавать многофункциональные приложения для всех видов операционных систем.

2. C# относиться к языкам компилируемого типа, поэтому он обладает каждым перечисленным преимуществами языков данного типа.

3. Из-за огромной вариации синтаксических конструкций и возможности работать с платформой .NET, C# позволяет быстрее, чем любой другой язык, разрабатывать программные решения.

4. C# отличается надежностью и элегантностью.

Недостатки:

1. Приоритетное внимание уделяется платформе Windows.

2. Язык бесплатный только для малого бизнеса, индивидуальных программистов, стартапов и студентов. Для крупной компании покупка лицензионной версии этого языка обойдется в немалую сумму.

3. В языке по-прежнему есть возможность использовать оператор безусловного перехода.

Java представляет собой язык программирования и платформу вычислений, которая была впервые выпущена Sun Microsystems в 1995 г. Java – строго типизированный объектно-ориентированный язык программирования общего назначения, созданный компанией Sun Microsystems.

Java – это больше чем язык программирования, это экосистема инструментов, охватывающая значительную часть того, что будет необходимо при программировании на Java. В нее входят:

1. Java Development Kit (JDK) — инструмент разработчика Java. При помощи JDK и стандартного блокнота возможно создавать/ компилировать код на Java;

2. Java Runtime Environment (JRE) — выполняющая система Java. Система распространения программного обеспечения, в данную систему входят: автономная виртуальная машины Java, стандартная библиотека Java (Java Class Library) и инструменты настройки.

3. Integrated Development Environment (IDE) — интегрированная среда разработки. Инструменты, которые помогают запускать, редактировать и компилировать код.

Преимущества:

1. Программирование ориентированное на обьект. В Java входит объектно-ориентированное программирование (OOP) – концепцию, в которой разработчик сам определяет тип данных и его структуру, а также все функции, которые будут выполнятся, применяемых к нему. Из этого следует, что структура данных становится объектом, которым можно управлять для создания отношений между различными объектами.

2. Java – язык высокого уровня с простым синтаксисом и плавной кривой обучения. Java – это язык высокого уровня, то есть он похож на человеческий язык. В отличие от низкоуровневых языков, напоминающих машинный код. Языки высокого уровня конвертируются с помощью компиляторов или интерпретаторов. Это упрощает программирование, упрощая написание, чтение и поддержку языка. Синтаксис Java основан на C ++, что делает Java похожей на C. Однако синтаксис Java проще, что позволяет новичкам быстрее изучать язык и более эффективно использовать код для достижения определенных результатов.

3. Стандарт для корпоративных вычислительных систем. Корпоративные приложения были главной силой Java с 1990-х годов, когда организации начали исследовать надежные инструменты программирования, отличные от C. Java поддерживает большое количество библиотек, строительных блоков любой системы. Библиотеки помогают разработчикам писать все функции, необходимые бизнесу. Java повсеместно распространена – это язык, который преподается как часть введения в программирование в большинстве школ и университетов. Потенциал интеграции Java впечатляет, поскольку многие хостинг провайдеры поддерживают Java. Кроме того, Java - недорогой в обслуживании язык: вы можете разрабатывать на Java с любого компьютера, независимо от конкретной аппаратной инфраструктуры.

4. Безопасность. Бытует мнение, что Java - безопасный язык, но это мнение не совсем верно. Сам язык не защищает от уязвимостей, но некоторые функции Java закрывают общие уязвимости. Во-первых, в отличие от C в Java нет указателя. Указатель – это объект, который хранит адрес места в памяти с другим значением, что может привести к несанкционированному доступу к памяти. Во-вторых, в Java есть диспетчер безопасности - политика безопасности, разработанная для каждого приложения, которая может указывать инструкции доступа. Это позволяет приложениям Java работать в изолированной программной среде, устраняя, таким образом, слабые места.

5. Автоматическое управление памятью. Автоматическое управление памятью (AMM) избавляет разработчиков Java от необходимости вручную писать код для управления памятью. AMM также используется в языке программирования Swift и для очистки памяти приложениями, которые автоматически обрабатывают выделение и восстановление памяти. Отдача программы напрямую связана с памятью. При этом объем хранилища ограничен. Написание приложения на языках с ручным управлением памятью может привести к тому, что разработчики забудут выделить память, увеличив объем памяти, используемый приложением, и вызовут проблемы с производительностью. Очистители памяти ищут объекты, которые больше не используются программой, и удаляют их. Это повлияет на производительность процессора, но интеллектуальная оптимизация и точная настройка могут смягчить это влияние.

Недостатки:

1. Платное коммерческое использование. Oracle недавно объявила, что с 2019 года будет взимать плату за «коммерческое использование» Java Standard Edition 8. Каждый новый патч – требует дополнительных вливаний денег. Комиссия прямо пропорционально количеству компьютеров.

2. Низкая производительность. У любого языка высокого уровня достаточно низкая производительность из-за компиляции и абстракции с помощью виртуальной машины. Однако это не единственный источник низкой скорости Java. Например, приложение очистки памяти: это полезная функция, которая, в большинстве случаев, приводит к значительным проблемам с производительностью, если требует более 20 процентов времени процессора. Плохая настройка кэширования вероятнее всего может вызвать чрезмерное употребление памяти. Возможна также взаимная блокировка потоков: так происходит, когда несколько потоков пытаются получить доступ к одному и тому же ресурсу. В этом случае происходит ужас каждого Java-разработчика – ошибка из-за нехватки памяти.

3. Отсутствие нативного дизайна. Разработчики используют различные языковые инструменты для создания графического пользовательского интерфейса (GUI). Для приложений Android существует Android Studio, которая помогает создавать приложения с собственным дизайном. Однако, когда дело доходит до пользовательского интерфейса на ПК, нет никакого собственного инструмента проектирования Java.

4. Многословный и сложный код. Многословие кода, кажется, может помочь выучить язык. Однако длинные и слишком сложные предложения затрудняют чтение и просмотр кода. Как и естественные языки, многие языки программирования высокого уровня содержат ненужную информацию. Java – это более легкая версия недоступного C ++, которая заставляет программистов писать свои действия англоязычными словами. Это делает язык более понятным для неспециалистов, но менее компактным.

C ++ (англ. C ++) – это скомпилированный строго типизированный язык программирования общего назначения. Поддерживает разные парадигмы программирования: процедурное, обобщенное, функциональное. Наибольшее внимание уделяется поддержке объектно-ориентированного программирования.

Преимущества:

1. Масштабируемость. Программы для самых разных платформ и систем разрабатываются на языке C ++.

2. Умение работать на низком уровне с памятью, адресами, портами. Что при неосторожном использовании может легко превратиться в недостаток.

3. Возможность создавать обобщенные алгоритмы для разных типов данных, их специализации и вычислений во время компиляции с использованием шаблонов.

4. Автоматический вызов деструкторов объектов (в порядке обратных вызовов конструкторов) упрощает и повышает надежность управления памятью и другими ресурсами (открытыми файлами, сетевыми соединениями и т. д.)

5. Высокая совместимость с языком C, позволяющая использовать весь существующий код C (код C может быть скомпилирован с помощью компилятора C ++ с минимальными изменениями; библиотеки, написанные на C, обычно можно вызывать непосредственно из C ++ без дополнительных затрат, даже на уровне функций обратного вызова, позволяя библиотекам, написанным на C, вызывать код, написанный на C ++).

Недостатки:

1. Наличие множества функций, нарушающих принципы безопасности типов, приводит к тому, что незаметная ошибка может легко закрасться в программы на C ++. Вместо того, чтобы находиться под контролем компилятора, разработчики вынуждены придерживаться весьма нетривиальных правил кодирования. По сути, эти правила ограничивают C ++ более безопасным подъязыком. Большинство проблем безопасности типов C ++ унаследованы от C, но важную роль в этом вопросе играет отказ автора языка от идеи использования автоматического управления памятью (например, сборка мусора). Таким образом, уязвимости, связанные с переполнением буфера, стали отличительной чертой C ++.

2. Плохая поддержка модульности. Подключение интерфейса внешнего модуля путем вставки файла заголовка (#include) препроцессором значительно замедляет компиляцию при подключении большого количества модулей. Чтобы преодолеть этот недостаток, многие компиляторы реализуют механизм предварительно скомпилированных заголовков для предварительной компиляции файлов заголовков.

3. Отсутствие информации о типе собираемых данных (CTTI).

4. Язык C ++ сложно выучить и скомпилировать.

5. Некоторые преобразования типов не интуитивно понятны. В частности, операции с числами без знака и со знаком дают беззнаковые результаты.

6. Препроцессор C ++ (унаследованный от C) очень примитивен. С одной стороны, это приводит к тому, что с его помощью невозможно (или сложно) выполнять некоторые задачи метапрограммирования, а с другой стороны, в силу своего примитивизма, он часто приводит к ошибкам и требует много действия по обходу потенциальных проблем. Некоторые языки программирования (такие как Scheme и Nemerle) имеют гораздо более мощные и безопасные системы метапрограммирования (также называемые макросами, но мало похожими на макросы C / C ++).

PHP – это мощный скриптованный язык общего назначения, ориентированный на разработку надежного веб-приложений.

Преимущества:

1. PHP - очень быстрый среди всех языков программирования. Так как он частично объектно-ориентирован, есть возможность повторно использовать код. Так же существует великое количество фреймворков помогающих с большим количеством задач.

2. PHP – полностью бесплатный язык, и это весьма экономит бюджет, а так=же увеличивает сообщество.

3. PHP- знаменит своей гибкостью и отличается почти полной совместимостью, он совмести со всеми операционными системами, поддерживает большинство серверов.

4. Очень хорошая поддержка данного языка программирования благодаря хорошей читабельности кода.

5. Современный «мощный» язык программирования: скриптованное программирование, шаблоны, гибкое управление распределением динамической памяти и т. д. и т. д.

Недостатки:

1. Веб-приложения, спрограммированный на языке PHP, в большинстве своем имеют проблемы с безопасностью.

2. Глобальные параметры конструкции влияют на базовый синтаксис языка, что усугубляет настройки серверов и разворачивание приложений;

3. Объекты отправляются по значению, что отталкивает многих программистов, привыкших к отправке частей по ссылке, как это делается в большинстве других языков;

4. На я зыке PHP невозможно создать десктопное приложение или любой системный компонент, он подходит только для веб-приложений;

6. Возможные противоречия синтаксиса. Поскольку над развитием PHP работают не люди, создавшие язык, большинство их действий оказались несогласованными. Код содержит элементы, характерные для различных языков, а также различные наименования встроенных функций, в связи с чем может возникать путаница.

На основе проведённого сравнения был выбран язык программирования Python так как он лучше подходит под наши задачи.

1.4 Анализ существующих сред разработки для Python

Выбор среды разработки на Python является одним из ключевых моментов при создании программ, нужно тщательно выбрать IDE для комфортного использования под конкретные задачи пользователя.

Рассмотрим три самые используемые среды разработки Python. Среды разработки не перестают создаваться, обновляться и совершенствоваться активными пользователями. Каждая из них практически не имеет ограничений в возможностях расширения функционала благодаря бескрайним массивам аддонов и плагинов. Тем не менее, какую лучше использовать IDE?

Конечно стоит отталкиваться от того, что нам необходимо. Поэтому выбор зависит от целей, знание IDE создателем приложения, и функций создаваемого проекта. Различия между самыми популярными средами разработки представлена в таблице 1.

Таблица 1

|  |  |  |  |
| --- | --- | --- | --- |
|  | **PyCharm** | **Eclipse** | **Visual Studio** |
| **Описание** | PyCharm —продукция компании JetBrains, разработанная специально для программирования на языке Python, также имеет встроенные инструменты разработчика | Eclipse, — свободная внутренняя среда разработки модульных кроссплатформенных приложений. Развивается и поддерживается Eclipse Foundation. | Microsoft Visual Studio —продукция компании Microsoft, включающих в себя внутреннюю среду разработки программного обеспечения и ряд других инструментальных инструментов. |
| **Простота использования** | Так как это среда разработки была разработана специально для Python, то она лучше всего подойдет для нашего проекта. | Бесплатность. Это открытый проект, полностью бесплатный. | Из-за того, что данное приложение было разработанно Microsoft, следует вывод, что нет ничего более функционального во всем интернете. |
| **Особенности** | PyCharm предоставляет встроенный отладчик и инструмент запуска тестов, профилировщик Python, встроенный терминал, инструменты для работы с базами данных и интеграцию с популярными системами контроля версий. | Множество плагинов. У Eclipse возможно самое огромное количество надстроек. Активное сообщество. Поможет как можно скорее овладеть средой разработки. | Совместим с платформами .NET. Visual Studio имеет широчайшие возможности по разработке приложений под Windows, в том числе в .NET-сегменте. |
| **Установка** | C сайта https://www.jetbrains.com/pycharm/ | C официального сайта: https://www.eclipse.org/ | C сайта https://code.visualstudio.com/ |

В связи с проведённым сравнением средств разработки было принято решение использовать PyCharm Community Edition.

1.5 Анализ предметной области

Распознавание изображения относится к задаче ввода изображения в нейронную сеть и присвоения какой-либо метки для этого изображения. Метка, которую выводит сеть, будет соответствовать заранее определенному классу. Может быть присвоено как сразу несколько классов, так и только один. Если существует всего только один класс, обычно применяется термин «распознавание», тогда как задача распознавания нескольких классов часто называется «классификацией».

Подмножество классификаций изображений - является уже определением объектов, когда определенные экземпляры объектов идентифицируются как принадлежащие к определенному классу, например, животные, автомобили или люди.

Ярким примером такой классификации является решение самой распространённой капчи — ReCaptcha v2 от Google, где из набора картинок необходимо выбрать только те, которые принадлежат к указанному в описании классу.

Чтобы выполнить распознавание или классификацию изображений, нейронная сеть должна выполнить извлечение признаков. Признаки - это элементы данных, которые представляют максимальный интерес и которые будут передаваться по нейросети. В конкретном случае распознавания изображений такими признаками являются группы пикселей, такие как линии и точки, которые сеть будет анализировать на наличие паттерна.

Распознавание признаков (или извлечение признаков) - это процесс извлечения соответствующих признаков из входного изображения, чтобы их можно было проанализировать. Многие изображения содержат аннотации или метаданные, которые помогают нейросети находить соответствующие признаки.

Созданная программа должна распознавать номерные знаки на автомобилях.

1.6 Исследование предметной области

Предметная область — это часть реального мира, подлежащая изучению с целью создания базы данных для автоматизации процесса управления.

Информационная система является совокупностью программно-аппаратных средств, способов и людей, которые обеспечивают сбор, хранение, обработку и выдачу информации для решения поставленных задач написанный на языке Python.

Python - высокоуровневый язык программирования общего назначения, созданный Python Software Foundation и Гвидо ван Россумом. На этом языку программирования можно написать почти всего, что можно, от веб-сайтов до создания искусственного интеллекта и нейросетей.

За счёт читабельности, простого синтаксиса и отсутствия необходимости в компиляции язык хорошо подходит для обучения программированию, позволяя концентрироваться на изучении алгоритмов, концептов и парадигм. Язык также является кроссплатформенным, это значит, что написанная программа будет работать на любой операционной системе.

Каждый этап подразумевает последовательное проектирование и включает ряд особенностей.

Любая информационная система в зависимости от ее назначения работает с той или иной частью реального мира, которую необходимо называть предметной областью системы. Предметная область информационной системы описывается, прежде всего, как некоторая совокупность реальных объектов (сущностей), которые представляют интерес для ее пользователей.

Пользователь сможет:

1. Загрузить изображение
2. Распознать номер на изображении

# ПРАКТИЧЕСКАЯ ЧАСТЬ

* 1. Разработка структуры

Так как количество транспортных средств увеличивается пропорционально количеству населения возникает необходимость в системах, которые будут следить и вести учет проезжающих транспортных средств.

В настоящее время автомобилей огромное количество и создание автоматических систем, которые могли бы облегчить процесс распознавания номеров и автоматизировать его, необходимы для прежде всего для безопасности. По этой причине, приложение актуально.

Главными плюсами использования системы могут выступать:

- автоматизация предприятия

- повышение уровня безопасности

- повышение производительности предприятия

- уменьшение ложных срабатываний

Главной целью этой выпускной квалификационной работы является конструирование системы автоматического распознавания автомобильных номерных знаков, позволяющего упростить работу людей и ускорить ее.

Необходимо рассмотреть цели автоматизированного варианта решения задачи:

- уменьшение времени определения номера и сохранение номера в базу данных;

- повышение точности определения;

Система для автоматического распознавания номеров дожна обеспечивать выполнение следующих основных функций:

- загрузка изображения;

- определение номера на изображение;

* 1. Создание приложения

Для дипломной работы была использована среда разработки PyCharm Community Editon 2021. При открытии данной программы необходимо создать новый проект для этого нажимаем на кнопку «New Project» (рис. 6).

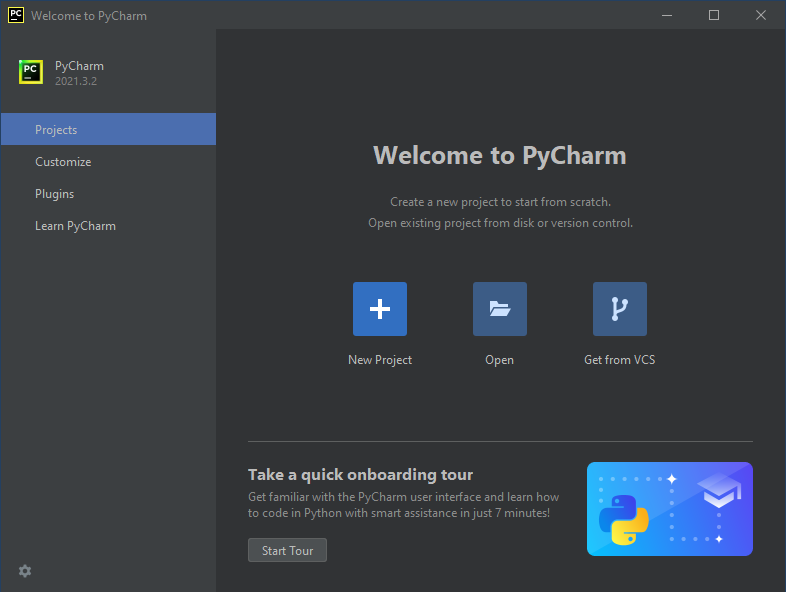


Рисунок 6. Создание нового проекта

После чего присваиваем проекту имя на английском и выбираем путь, после чего жмем кнопку «Create» (рис. 7).

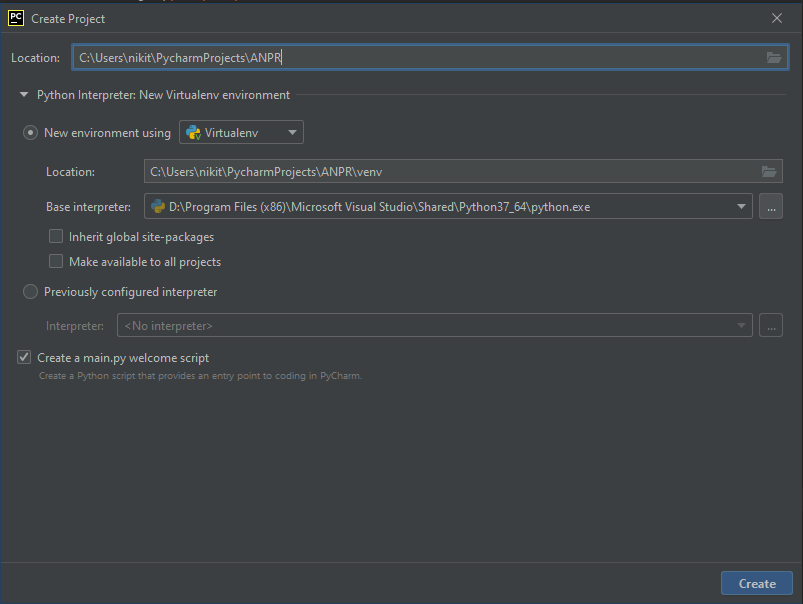


Рисунок 7. Выбор имени проекта

После создания проекта нужно подключить все необходимые библиотеки и системы для будущего приложения в терминале используя команду «pip install» (листинг 1) (рис. 8).

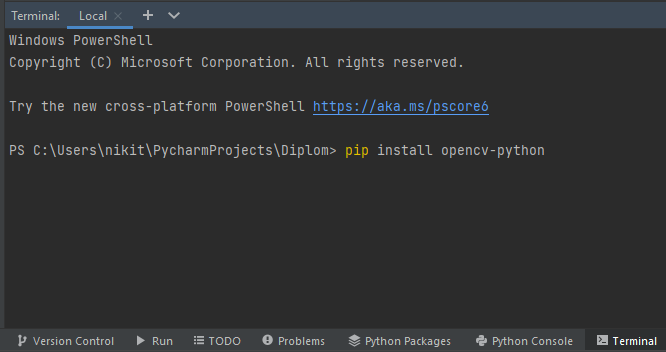
Листинг 1. Подключение библиотек через терминал

pip install opencv-python

pip install easyocr

pip install imutils

pip install matplotlib

****Рисунок 8. Подключение OpenCV через Terminal

Библиотека OpenCV расшифровывается как Open Source Computer Vision Library (*библиотека с открытым исходным кодом для компьютерного зрения*). Название говорит само за себя. Данная библиотека позволяет работать с изображениями и видео, распознавать на них объекты, информацию, текст и прочие моменты.

Вместе с OpenCV в проект также автоматически устанавливается библиотека NumPy. Нам потребуется обрабатывать огромные массивы данных. К примеру, чтобы понять, что находится на фото его нужно будет привести в нужный формат и далее перебрать каждый пиксель. Для работы со столь большим набором данных можно использовать «NumPy». Речь про эту библиотеку заходит практически каждый раз при разработке проектов с искусственным интеллектом. Библиотека содержит набор встроенных функций для быстрой обработки многомерных массивов данных.

Система EasyOCR(Optical Character Recognition), которая с помощью методов машинного обучения, определяет текст на изображениях. Проектом EasyOCR развивается новая система оптического распознавания текста, поддерживающая более 40 языков, включая английский, немецкий, французский, японский, китайский, корейский, узбекский, азербайджанский и литовский. Языки на основе кириллицы пока не поддерживаются, но их добавление в списке планов. Для определения текста используется алгоритм машинного обучения CRAFT (Character-Region Awareness For Text) в реализации для PyTorch, способный выделять текст на произвольных объектах, включая этикетки, информационные таблички и дорожные знаки. Для распознавания последовательностей символов применяется свёрточно-рекуррентная нейронная сеть CRNN (Convolutional Recurrent Neural Network, комбинация DCNN и RNN) и алгоритм CTC BeamSearch (Connectionist Temporal Classification) для декодирования выходных данных нейронной сети в текстовое представление.

Пакет imutils, так как мы будем работать с изображениями нам необходим пакет, который упростит вызов интерфейсов OpenCV, который также может легко реализовывать серии операций, таких как перевод изображения, вращение, масштабирование скелет и отображении Matplotlib.

Библиотека Matplotlib используется для двумерной графики для языка программирования Python с помощью, которой можно создавать высококачественные рисунки различных форматов.

После подключения всех необходимых библиотек их нужно импортировать в наш проект, чтобы мы в дальнейшем могли к ним обращаться. Используя команду «import» импортируем установленные библиотеки и пакеты (листниг 2) (рис. 9).

Листинг 2. Импортирование библиотек и пакетов в проект

import cv2

import numpy as np

import imutils

import easyocr

from matplotlib import pyplot as pl

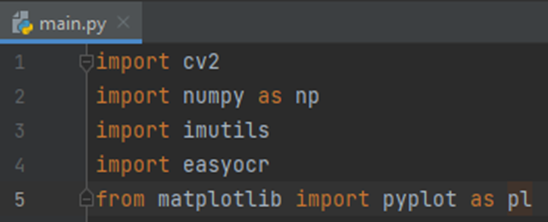


Рисунок 9. Импортирование библиотек и пакетов в проект

Далее создадим переменную img, которая будет считывать изображения используя установленную нам библиотеку OpenCV и после нее переменную gray, которая переведет изображение из цветного в черно-белый формат (листинг 3) (рис. 10).

Листинг 3. Создание переменных img и gray

img = cv2.imread('images/car6.jpg')

gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

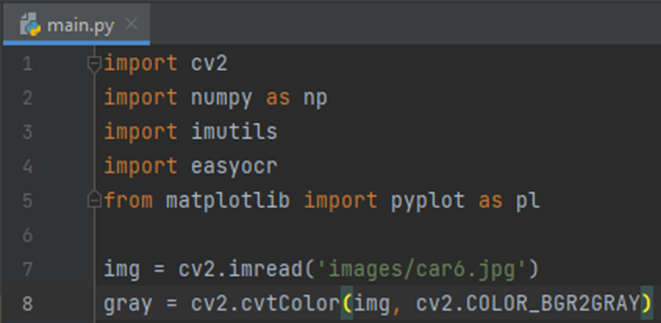


Рисунок 10. Создание переменных img и gray

После создания этих переменных создадим переменные img\_filter и edges. img\_filter будет отвечать за оптимизацию изображения уменьшая количество «шумов» на изображении. В edges пропишем выделение краев на изображении (листинг 4) (рис. 11).

Листинг 4. Уменьшение шумов и выделение краев

img\_filter = cv2.bilateralFilter(gray, 11, 15, 15)

edges = cv2.Canny(img\_filter, 30, 200)

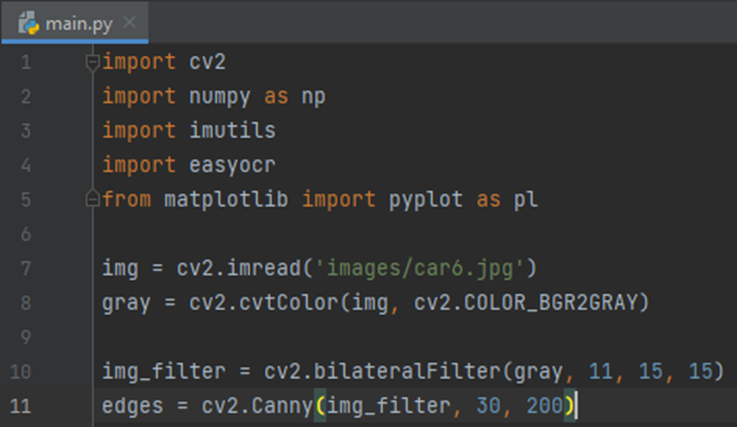


Рисунок 11. Уменьшение шумов и выделение краев

Теперь создадим переменную cont, пропишем ей нахождение начальной и конечной точки контура. Далее прописываем переменной считывание контуров при помощи, установленной нами ранее библиотекой imutils. После этого мы отфильтруем все контуры оставляя только те, которые похожи на квадрат или прямоугольник (листинг 5) (рис. 12).

Листинг 5. Нахождение контуров

cont = cv2.findContours(edges.copy(), cv2.RETR\_TREE, cv2.CHAIN\_APPROX\_SIMPLE)

cont = imutils.grab\_contours(cont)

cont = sorted(cont, key=cv2.contourArea, reverse=True)



Рисунок 12. Нахождение контуров

Так как мы уже отфильтровали контуры, которые сильнее всего напоминают прямоугольник, теперь нужно считать номер машины в этом прямоугольнике. Для этого создадим переменную pos и пока что пропишем в ней none, далее создадим цикл «c» в которой будет переменная approx ей мы пропишем, чтобы она искала в нашей отфильтрованной переменной cont наиболее подходящий контур похожий на прямоугольник и который имеет начальную и конечную точку в том же месте. После этого сделаем небольшую проверку, если длина в этом прямоугольнике состоит из 4 элементов. После завершения цикла присваиваем значения approx переменной pos (листинг 6) (рис. 13).

Листинг 6. Создание цикла

pos = None

for c in cont:

approx = cv2.approxPolyDP(c, 10, True)

if len(approx) == 4:

pos = approx

break

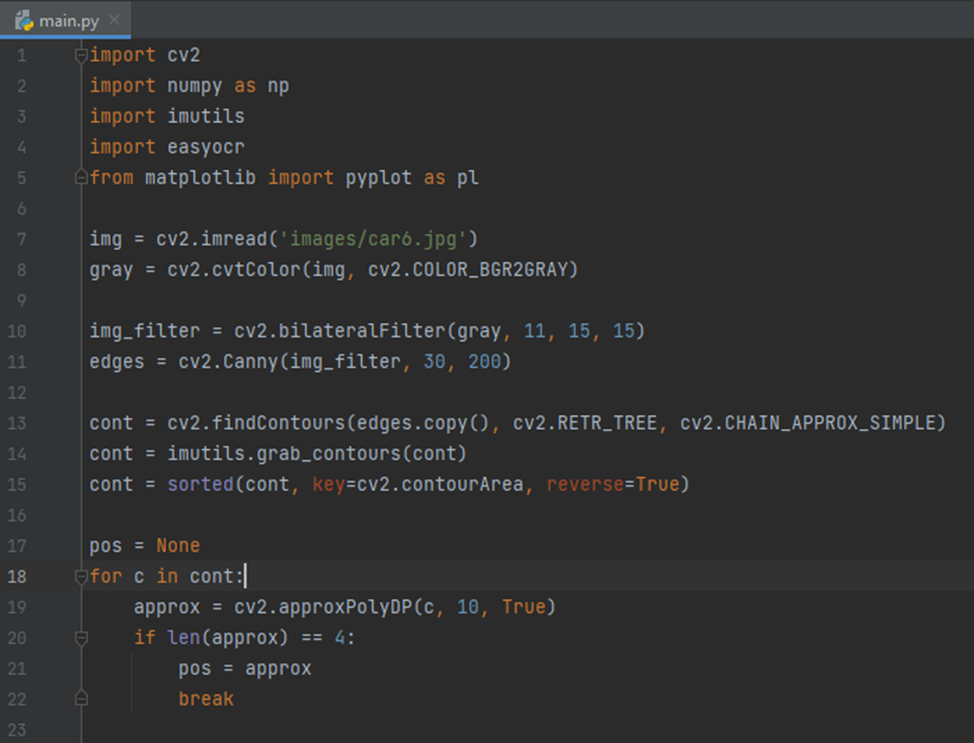


Рисунок 13. Создание цикла

После этого создадим маску, которая выделит нужный нам прямоугольник для этого нам и пригодится библиотека numpy. Создаем переменную mask и выбираем полотно. После этого создаем новое изображение new\_image, на котором будет рисоваться контур маски и где он будет расположен. Теперь мы создадим переменную bitwise\_img, которая и будет применять созданную нами маску (листинг 7) (рис. 14).

Листинг 7. Создание маски

mask = np.zeros(gray.shape, np.uint8)

new\_img = cv2.drawContours(mask, [pos], 0, 255, -1)

bitwise\_img = cv2.bitwise\_and(img, img, mask=mask)

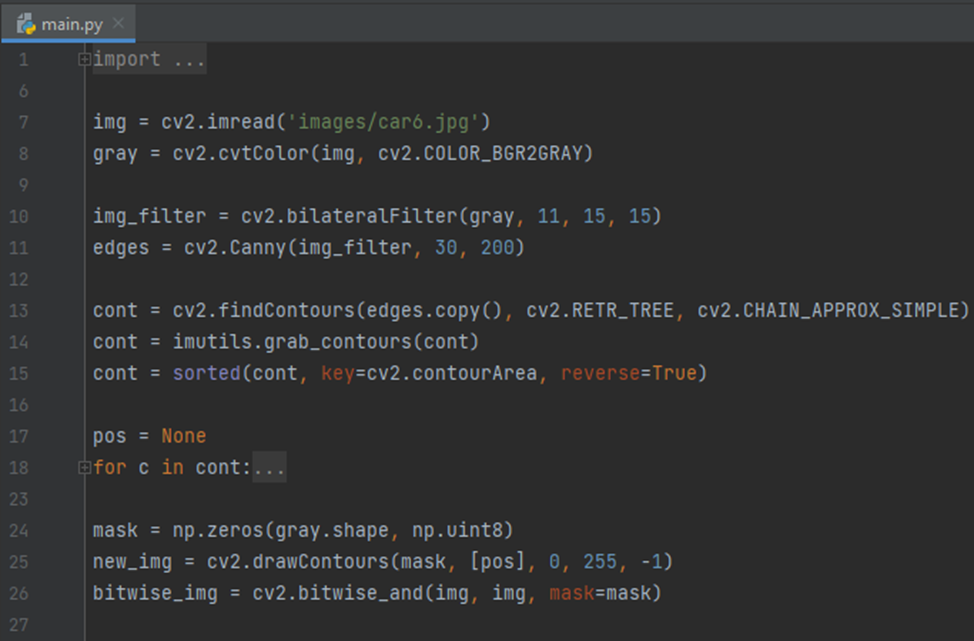


Рисунок 14. Создание маски и ее отображение

Для упрощения прочтения номерных знаков мы обрежем изображение оставив только прямоугольник, который выделила маска. Для этого создаем переменные «x» и «y», прописываем их, чтобы они «вытянули» весь белый цвет, который сделала маска. Далее создаем переменные «x1», «y1» и «x2», «y2», чтобы они отвечали за верхние и нижние координаты нашего знака. Теперь уже можно создать переменную crop, которая и отобразит вырезанный знак (листинг 8) (рис. 15).

Листинг 8. Вырезание номерного знака

(x, y) = np.where(mask == 255)

(x1, y1) = (np.min(x), np.min(y))

(x2, y2) = (np.max(x), np.max(y))

crop = gray[x1:x2, y1:y2]

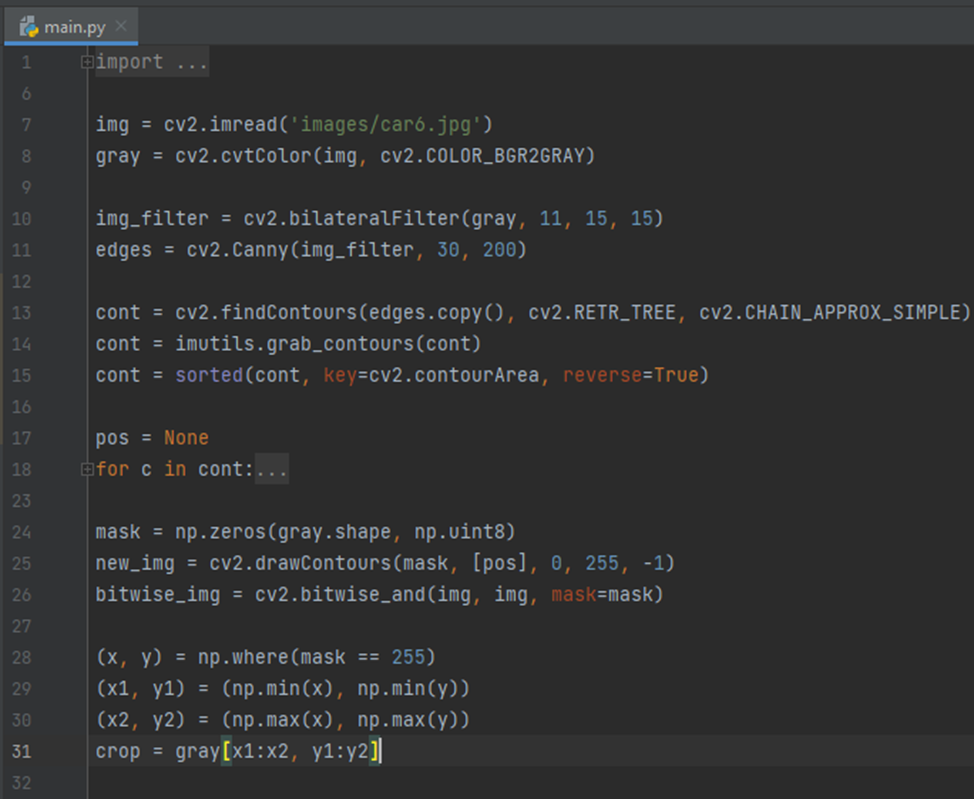


Рисунок 15. Вырезание номерного знака

После обрезания номерного знака можно начать считывать то, что на нем написано. Создаем переменную text и используя систему EasyORC считываем текст с переменной crop, предварительно прописав, символы какого языка искать (листинг 9) (рис. 16).

Листинг 9. Считывание номерного знака

text = easyocr.Reader(['en'])

text = text.readtext(crop)

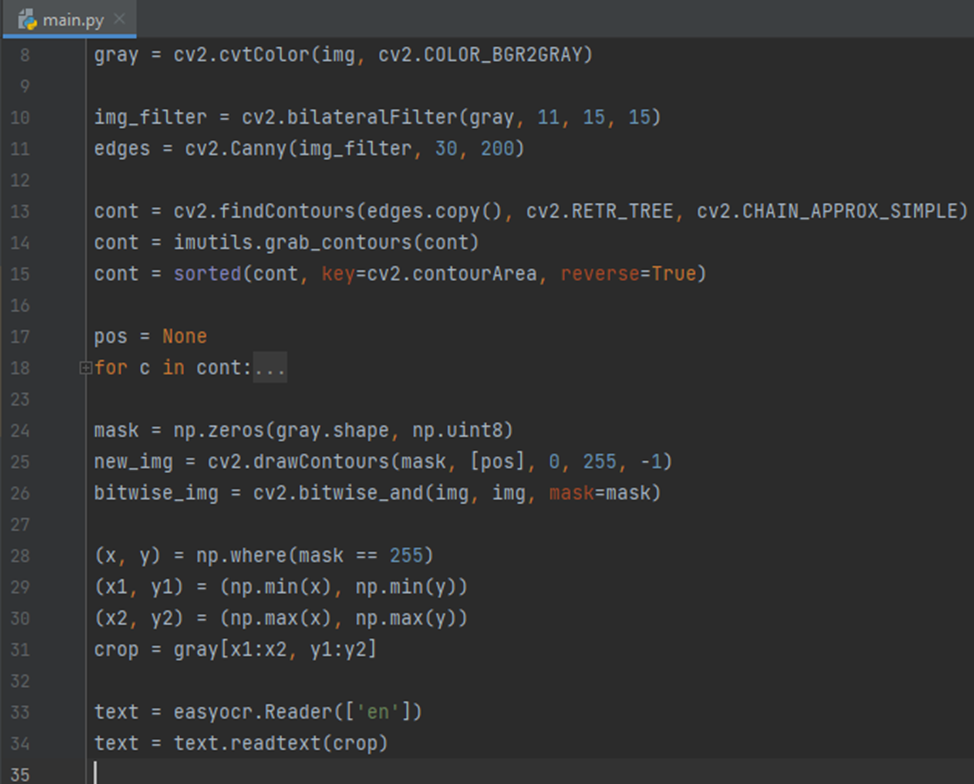


Рисунок 16. Считывание номерного знака

Для отображения проделанной работы приложения создадим переменную res прописываем ей, чтобы она из всего, что считала с номерного знака отобразила только буквы и цифры номерного знака. Далее создадим переменную final\_image и пропишем ей, чтобы создала красный текс со считанным номером и выделила зеленым прямоугольником номерной знак. А также напишем вывод изображения в приложение с помощью matplotlib (листинг 10) (рис. 17).

Листинг 10. Выделение номерного знака и его вывод

res = text[0][-2]

final\_image = cv2.putText(img, res, (y2, x1 + 60), cv2.FONT\_HERSHEY\_PLAIN, 2, (0, 0, 255), 2)

final\_image = cv2.rectangle(img, (y2, x2), (y1, x1), (0, 255, 0), 2)

pl.imshow(cv2.cvtColor(final\_image, cv2.COLOR\_BGR2RGB))

pl.show()

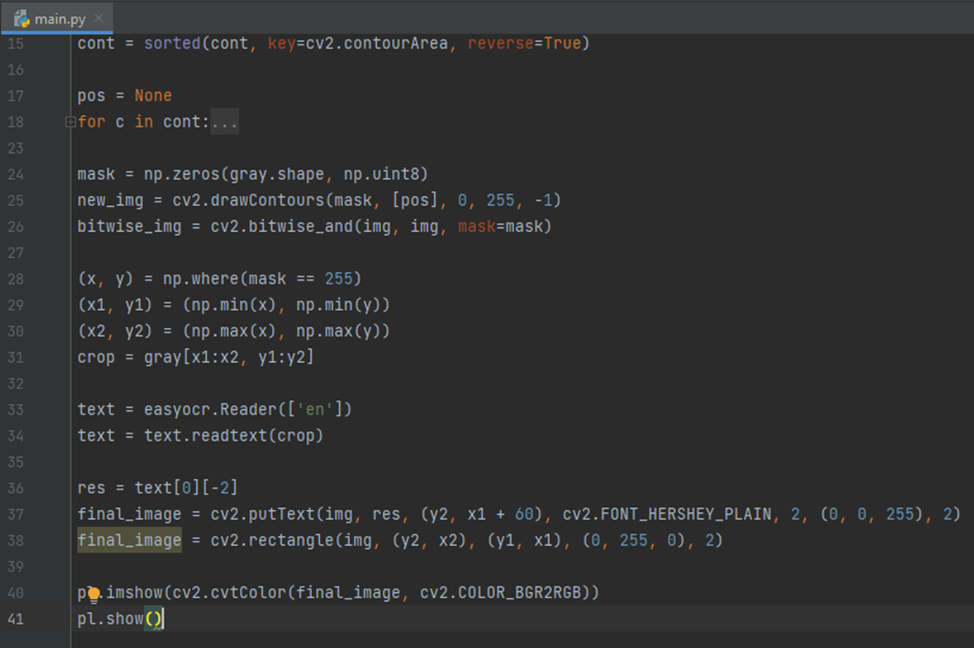


Рисунок 17. Выделение номерного знака и его вывод

Итоговый вариант приложения выглядит вот так (рис. 18):



Рисунок 18. Итоговое изображение

# Выводы

В результате проделанной работы была разработана система для распознавания автомобильных номерных знаках, а также были решены следующие задачи:

1. Был проведен обзор литературы по данной тематике от 4 авторов: Горбачевской Е. Н., Краснова С. С., Фаустов К.И и «Videomir». А также прочитаны статьи в интернет ресурсах.
2. Сделан обзор существующих автоматических систем для распознавания номерных знаков. Был выполнен обзор и сравнение следующих систем для распознавания номерных знаков: «iPera EX-LPR», «CVS-Авто», «AutoTRASSIR», «НомерОк», «Macroscop-Авто», а также были выделены преимущества и недостатки этих систем.

Запрограммированы основные модули программы, а именно загрузка изображения, распознавание номера и вывод распознанного номера на изображении.