КУРСОВАЯ РАБОТА

на тему: «Автоматизированное распознавание государственных номеров для учета времени стоянки и оплаты»

2023

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc147223311)

[1 ОБЗОР МЕТОДОВ И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ РАСПОЗНОВАНИЕ НОМЕРОВ АВТОМОБИЛЕЙ 4](#_Toc147223312)

[1.1 Методы распознавания номерных знаков 5](#_Toc147223313)

[1.1.1 Метод Виолы-Джонса 5](#_Toc147223314)

[1.1.2 Машины опорных векторов 7](#_Toc147223315)

[1.2.3 Использование искусственных нейронных сетей 9](#_Toc147223316)

[1.2 Обзор систем распознавания номерных знаков 14](#_Toc147223317)

[1.2.1 «Автомаршал» 14](#_Toc147223318)

[1.2.2 "Spetslab-Traffic" 17](#_Toc147223319)

[1.2.3 "SecurOS Auto" 19](#_Toc147223320)

[1.3 Проверка систем распознавания номерных знаков 25](#_Toc147223321)

[1.4 Итог камеральной проверки 26](#_Toc147223322)

[1.5 Требования к разработанному алгоритму 27](#_Toc147223323)

[2 АЛГОРИТМ РАЗОЗНАЧЕНИЯ АВТОМОБИЛЬНЫХ НОМЕРОВ НЕЙРОСЕТЬЯМИ 28](#_Toc147223324)

[2.1 Общее решение задачи 28](#_Toc147223325)

[2.2 Поэтапная работа с изображением 28](#_Toc147223326)

[2.2.1 Сегментация номерных знаков 29](#_Toc147223327)

[2.2.2 Удаление фрагмента с номерным знаком 31](#_Toc147223328)

[2.2.3 Предварительная обработка вырезанного фрагмента 34](#_Toc147223329)

[2.2.4 Классификация сегментов номерных знаков 35](#_Toc147223330)

[3 РЕАЛИЗАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 38](#_Toc147223331)

[3.1 Язык разработки 38](#_Toc147223332)

[3.2 Среда разработки 42](#_Toc147223333)

[3.3 Разработка программы 44](#_Toc147223334)

[3.4 Работа с базой данных 46](#_Toc147223335)

[3.5 Результат работы программы 48](#_Toc147223336)

[ЗАКЛЮЧЕНИЕ 54](#_Toc147223337)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 56](#_Toc147223338)

# ВВЕДЕНИЕ

Актуальность работы.

Задача распознавания номерных знаков востребована в различных системах управления транспортными средствами. Эти системы могут использоваться разными государственными учреждениями, пунктами контроля скорости и т.д.

Цель работы.

Целью работы является разработка и внедрение алгоритма обнаружения и распознавания номерных знаков на изображениях реальных сцен с помощью аппарата искусственных нейронных сетей для учета времени стоянки и оплаты.

Постановка задачки.

Достижение этой цели предполагает решение следующих задач:

* разработка и внедрение алгоритма обнаружения номерных знаков;
* предварительная обработка изображения для дальнейшего распознавания;
* разработка и внедрение алгоритма распознавания номерных знаков;
* проведение многочисленных экспериментов.

Предмет исследования.

Предметом исследования есть методы искусственного интеллекта, которые используются в алгоритмах выявления и распознавания автомобильных номерных знаков. Применение сверточных нейронных сетей для сегментации, а также классификаторы буквенно-цифровых символов автомобильных номерных знаков.

# 1 ОБЗОР МЕТОДОВ И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ РАСПОЗНОВАНИЕ НОМЕРОВ АВТОМОБИЛЕЙ

Компьютерное зрение продолжает занимать важное место в научных исследованиях, целью которого является глубокое осмысление цифровых изображений [1]. Процессы включают методы сбора, обработки, интерпретации и осознания цифровых изображений, которые можно категорировать на четыре группы [2]:

* Классификация – определение категории объекта на изображении;
* Семантическая сегментация – выделение пикселей, принадлежащих одному и тому же классу объектов на изображении. При этом, пересекающиеся объекты одной категории не различаются между собой;
* Детектирование объектов – определение и обрамление всех объектов определенных категорий;
* Сегментация по экземплярам – выделение пикселей, принадлежащих каждому конкретному объекту класса.

На рис. 1.1 показан пример сегментации на изображении воздушных шаров.

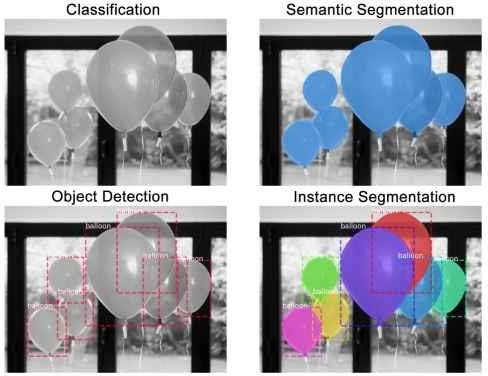


Рисунок 1.1 – Иллюстрация типов компьютерного зрения

## 1.1 Методы распознавания номерных знаков

Для выявления и идентификации автомобильных номерных знаков применяются различные техники. Давайте разберем несколько из них: метод Виолы-Джонса и машины опорных векторов.

Метод Виолы-Джонса, однако, страдает от того, что при углах наклона, превышающих 30 градусов, его точность и эффективность снижаются [3].

Опорные векторные машины, в свою очередь, ограничены в том, что они применимы только к задачам с двумя классами.

Тем временем, сверточные нейронные сети демонстрируют лучшие результаты при распознавании объектов на изображениях. В связи с этим в данном исследовании было решено прибегнуть к использованию сверточных нейронных сетей для обнаружения и идентификации номерных знаков автомобилей при различных углах наклона и в условиях шума.

Рассмотрим вышеуказанные методы более подробно.

### 1.1.1 Метод Виолы-Джонса

Метод, придуманный П. Виолой и М. Джонсом в 2001 году и известный как метод Виолы-Джонса, представляет собой инновационный алгоритм для обнаружения объектов в реальном времени. Главной его задачей является выявление лиц, однако он также успешно применяется для определения различных классов изображений [3].

Этот метод можно разбить на две ключевые части: алгоритм обучения и алгоритм распознавания. Основные принципы, лежащие в его основе, включают:

* Использование целостного представления изображений, что значительно ускоряет вычисления, необходимые для поиска объектов.
* Применение знаков Хаара для точного обнаружения интересующих объектов, в данном случае - угловых точек.
* Строительство классификатора, использующего алгоритм boosting (что переводится с английского как "усиление"), для выбора наиболее подходящих признаков в заданной части изображения.
* Все извлеченные признаки подвергаются анализу классификатором, который выдает результат "истина" или "ложь".
* Использование каскадов функций для эффективного отсеивания ненужных окон без объекта, что ускоряет процесс обнаружения.

Перечисленные принципы способствуют поиску объектов в реальном времени, однако следует учитывать, что основным недостатком данного подхода является длительное время обучения и необходимость большого объема обучающих данных для классификатора. Если угол наклона объекта на изображении превышает 30 градусов, процент успешного обнаружения снижается. Кроме того, при изменениях в освещении или сильном затемнении объекта, алгоритм может не давать нужных результатов.

Однако у метода есть и свои преимущества:

* возможность обнаружения более чем одного объекта;
* применение простых классификаторов;
* высокая скорость поиска;
* возможность использования в видеопотоке.

Давайте рассмотрим экспериментальные данные, связанные с этим методом.

Общее количество символов x = 1620

Количество сегментированных областей s=2247

Количество правильно сегментированных у = 1348

Средняя точность сегментации = 

Средняя ошибка сегментации = 

В таблице 1.1 представлен отчет по отдельным группам чисел. Точность определяется процентом правильно распознанных символов по отношению к общему количеству символов. [4].

Таблица 1.1 – Расчет групповой точности

|  |  |
| --- | --- |
| Группа | Точность сегментации Метод Виолы-Джонса, % |
| Нормальный | 90 |
| Под углом | 80 |
| Искаженный | 73 |

### 1.1.2 Машины опорных векторов

Метод опорных векторов (SVM) представляет собой управляемый алгоритм обучения, связанный с линейными классификаторами, применяемый для задач регрессии и классификации. Тихоновская регуляризация может быть рассмотрена как частный случай SVM [5]. Этот метод классификации с максимальным разрывом постоянно уменьшает эмпирическую ошибку классификации и увеличивает разрыв.

Вспомогательная векторная машина представляет интересную идею: преобразование исходных векторов в пространство большей размерности, где ищется разделительная гиперплоскость. Полученная гиперплоскость имеет две параллельные гиперплоскости по бокам и максимальный зазор в этом пространстве. Разделяющая плоскость является плоскостью с максимальным расстоянием до этих параллельных гиперплоскостей. Средняя погрешность результатов метода зависит от разницы или расстояния между этими параллельными гиперплоскостями; с увеличением расстояния погрешность уменьшается [6].

Классификация данных - это распространенная задача в мире алгоритмов машинного обучения. В этой задаче каждый элемент данных представлен вектором чисел в многомерном пространстве. Каждый элемент данных принадлежит одному из двух классов, и наша цель - найти гиперплоскость, которая разделяет эти классы максимально четко. Это достигается путем поиска гиперплоскости такой, что расстояние от нее до ближайшего элемента данных наибольшее. Эта оптимальная разделяющая гиперплоскость, а также соответствующий ей линейный классификатор, играют ключевую роль в успешной классификации данных.

Рассмотрим экспериментальные данные по методу:

Количество правильно сегментированных у = 1561

Количество SVM, распознанных алгоритмом svm = 1601

Средняя точность распознавания для алгоритма:



В таблице 1.2 представлен отчет об отдельных группах чисел для алгоритма SVM.

Таблица 1.2 – Расчет групповой точности

|  |  |
| --- | --- |
| Группа | Точность сегментации Метод Виолы-Джонса, % |
| Нормальный | 99 |
| Под углом | 99 |
| Искаженный | 98 |

### 1.2.3 Использование искусственных нейронных сетей

Человеческий мозг проявляет выдающиеся способности в решении задачи распознавания образов, превосходя в этом аспекте даже самые мощные цифровые компьютеры. Это обусловлено особенностями информационной обработки мозга, которые существенно отличаются от применяемых в обычных цифровых компьютерах методов. Мозг человека представляет собой сложную, нелинейную, параллельную систему обработки информации. Он способен так организовать свои структурные компоненты, чтобы успешно выполнять конкретные задачи, такие как распознавание образов, с многократным преимуществом в скорости по сравнению с самыми быстрыми компьютерами [7].

В данной работе предлагается использовать искусственные нейронные сети (ИНС). Согласно [7], нейронную сеть можно определить следующим образом:

"Искусственная нейронная сеть представляет собой масштабный распределенный параллельный процессор, состоящий из элементарных блоков для обработки информации, называемых нейронами. Эти нейроны накапливают экспериментальные знания и предоставляют их для последующей обработки".

Преимущества нейронных сетей были описаны в [7] и [8]. Давайте перечислим некоторые из них:

* Нелинейность, что позволяет моделировать сложные отношения в данных.
* Преобразование входящей информации в выходные результаты.
* Адаптивность, позволяющая сети настраиваться под различные задачи.
* Понятность ответов, делая выводы из сети более интерпретируемыми.
* Учёт контекстной информации для более точных прогнозов.
* Способность к отказоустойчивости, что позволяет сохранять работоспособность даже при некоторых повреждениях.
* Единообразный подход к анализу и проектированию.

Процесс обучения сети, или настройки, представляет собой процесс модификации свободных параметров нейронной сети, осуществляемый путем моделирования её взаимодействия с окружающей средой. Тип обучения определяется методом выбора характеристик [7].

1. Этот процесс настройки можно описать следующей последовательностью действий:
2. Сеть получает стимулы извне.
3. Свободные параметры сети подвергаются изменениям.

После изменений во внутренней структуре нейронная сеть реагирует на входные стимулы по-новому.

Алгоритм обратного распространения, известный как один из основных методов обучения, находит свое место среди разнообразных алгоритмов обучения. Наиболее известным и широко применяемым из них является алгоритм обратного распространения. Этот метод нацелен на минимизацию разницы между реальными выходными значениями нейронной сети (ANN) и желаемыми выходами. Его детальное описание можно найти в источниках [7], [8], [9], [10], [11] и [12].

Суть обучения через алгоритм обратного распространения заключается в переводе задачи отображения от входных данных к выходным значениям (путем использования набора примеров) в установку оптимальных синаптических весов и порогов для многослойного персептрона [7]. Процесс настройки сети можно рассматривать как выбор наилучшей модели из набора "кандидатов" структур, основанный на определенных критериях.

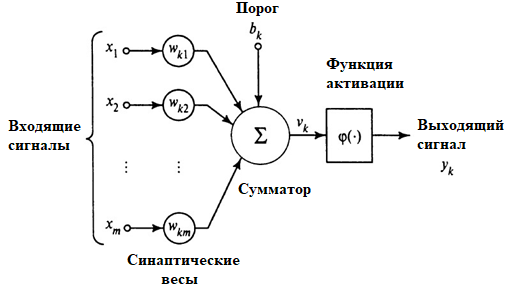


Рисунок 1.2 – Нелинейная модель нейрона [7]

Нейрон, в контексте нейронных сетей, представляет собой ключевую единицу обработки информации. На иллюстрации 1.2 мы можем увидеть модель нейрона, которая лежит в основе искусственных нейронных сетей. Эта модель включает в себя три основных компонента:

Ансамбль синапсов (или связей), каждый из которых имеет свой собственный вес или силу.

Сумматор, который складывает взвешенные входные сигналы, учитывая их веса, связанные с соответствующими синапсами нейрона.

Функция активации, которая ограничивает амплитуду выходного сигнала нейрона.

Модель нейрона также может включать пороговый элемент, обозначаемый символом , который влияет на функцию активации, регулируя входной сигнал.

Математически, работу нейрона можно описать следующими уравнениями [7]:





где , , … ,  – входные сигналы; , , … ,  — синоптические весы нейрона k;  - линейная комбинация входных действий (выход линейного комбинатора);  - порог;  – функция активации;  является выходным сигналом нейрона.

Постсинаптический потенциал рассчитывается следующим образом:



Функции активации - это специальные инструменты, которые формируют выход сигнала нейрона, в зависимости от воздействия на него локального поля. Давайте рассмотрим несколько разнообразных функций активации.

Одной из таких функций является сигмовидная функция. Она выделяется своей способностью стремительно переходить от линейного к нелинейному поведению. Логистическая функция является примером сигмовидной функции и может быть представлена следующим образом:



Другой функцией активации является линейный выпрямитель (ReLU), заданный следующим выражением:



Сверточные нейронные сети (CNN), разработанные Яном Лекуном в 1988 году, представляют собой инновационную архитектуру в мире искусственного интеллекта. Их первоначальная цель заключалась в улучшении процесса распознавания объектов на изображениях, и сегодня они остаются неотъемлемой частью глубокого обучения. CNN считаются одной из самых важных технологических новшеств в области компьютерного зрения.

Структура CNN включает в себя разнообразные слои, такие как сверточные, нелинейные, объединяющие и полностью связанные. Процесс обработки изображения через эти слои завершается формированием ответа, который может представлять собой класс объекта или вероятность принадлежности к определенным классам, наиболее точно описывающим изображение. CNN являются однонаправленными, то есть они не имеют обратной связи и представляют собой многослойную нейронную сеть.

В процессе обучения CNN используются стандартные методы, часто опираясь на алгоритм обратного распространения ошибки. Отмечается, что функция активации в таких сетях может быть выбрана исследователем в зависимости от конкретных задач и требований.

Для выполнения задачи сегментации используется U-Net, одна из наиболее распространенных архитектур сверточных нейронных сетей. Она широко применяется в сегментации изображений, где требуется выделить различные области и создать маски для классификации. U-Net состоит из двух основных частей: сжимающего пути для захвата контекста и симметричного пути для более точной локализации [15], [16], [17].

Что делает эту архитектуру выдающейся, так это ее способность превзойти предыдущие методы, такие как сверточные сети с скользящим окном, даже при ограниченном объеме обучающих данных. Например, в соревновании ISBI по сегментации нейронных структур в стеках электронной микроскопии, эта архитектура, обученная на изображениях микроскопии пропускания света (фазовый контраст и DIC), заняла первое место в 2015 году. Ее выдающееся преимущество - скорость выполнения. Даже на современных графических процессорах, сегментация изображений размером 512×512 занимает менее секунды [15], [16], [17].

Таблица 1.3 – Результаты сегментации (IOU) на конкурсе отслеживания клеток ISBI 2015

|  |  |  |
| --- | --- | --- |
| Название | PhC-U373 | DIC-HeLa |
| IMCB-SG (2014) | 0.2669 | 0.2935 |
| KTH-SE (2014) | 0.7953 | 0.4607 |
| HOUS-US (2014) | 0.5323 | - |
| Second-best (2015) | 0.83 | 0.46 |
| U-Net | 0.9203 | 0.7756 |

В исследовании оптических микроскопических изображений для сегментации клеток на конкурсах ISBI по наблюдению за ячейками в 2014 и 2015 годах успешно применялась архитектура U-Net. Сравнивая ее с другими сетями, можно заметить, что U-Net продемонстрировала выдающиеся результаты. Подробности о данном исследовании можно найти в [15].

## 1.2 Обзор систем распознавания номерных знаков

Существует обширный ассортимент индустриальных решений в сфере определения автомобильных номеров. В данном разделе предлагается сжатый обзор нескольких программных решений, где каждое из них представляет уникальные характеристики, включая название, сферу применения и используемые технологии для распознавания.

### 1.2.1 «Автомаршал»

Автомаршал — это система, предназначенная для распознавания номерных знаков. Он используется для контроля проезда автомобилей, проверки их по разным базам данных и ограничения доступа автомобилей на территорию [18] (рис. 1.3).



Рисунок 1.3 – Автомаршал

Область применения:

* пункты пропуска предприятий;
* платные стоянки;
* парковка возле торговых центров;
* закрытые зоны;
* жилые комплексы;
* коттеджные поселки;
* автомойки;
* автоматический поиск автомобиля.

Возможности:

* выявление транспортных средств в зоне контроля и определение направлений их движения;
* признание государственных регистрационных номеров автомобилей;
* ведение базы проезжающих транспортных средств;
* контроль шлагбаумов и светофоров в автоматическом и автоматизированном режимах при организации проезда транспортных средств на закрытые территории (предприятия, автостоянки, ТСЖ, коттеджные поселки и т.п.);
* поддержка баз данных пользователей на транспортных средствах: тип «Клиенты», «VIP», «Разыскивается», «Черный список» и т.д.
* поиск распознанных номеров автомобилей в базах пользовательских данных и, если они совпадают, автоматический ответ по сценарию, определенному пользователем;
* передача данных распознавания в посторонние учетные системы;
* подготовка, печать, рассылка по SMS и электронной почте различных отчетов, формируемых как автоматически, так и по желанию пользователя;
* осуществление контроля за перемещением транспортных средств на территории предприятия и контроль за соблюдением установленных регламентов такого перемещения;
* дифференциация доступа различных категорий пользователей к сгенерированным данным по паролям или картам;

Методы и технологии распознавания

Распознавание номерных знаков осуществляется с помощью ядра распознавания RecarKernel (разработка научно-производственной компании Mullen). В системе распознавания номерных знаков

«Автомаршал» для максимально надежной работы в различных условиях (день, ночь, осадки, яркое солнце, фары) использует два параллельно работающих алгоритма. Первый алгоритм базируется на нейронных сетях, второй алгоритм классификатор на основе многомерной интерполяции и аппроксимации [18].

### 1.2.2 "Spetslab-Traffic"

"SL-Traffic" — программный модуль для считывания и распознавания номерных знаков в режиме реального времени. Модуль работает как видеофильтр в системе или может быть интегрирован в любую систему видеозаписи, в том числе посторонних (рис. 1.4).



Рисунок 1.4 – Spetslab-Traffic

Область применения:

* в сочетании с программой SpetLab – Registration Check, разработанной специально для обеспечения эффективной работы сотрудников ГАИ, SL-Traffic позволяет автоматически регистрировать транспортные средства, проходящие через пункт пропуска ГАИ без участия человека.
* в сочетании с программой «СпецЛаб – Досье» SL-Traffic служит для обеспечения безопасности, контроля въезда/выезда транспортных средств со стоянок, коллективных гаражей и запрещенных зон;
* контроль движения автомобилей, как в черте города, так и на трассах.

Возможности:

* распознавание всех однострочных номеров, выпущенных в России и ряде других стран ближнего и дальнего зарубежья;
* одновременное распознавание до 20 номерных знаков на одном кадре (большее количество номеров просто не поместится в кадре с необходимым качеством изображения);
* оптическое определение скорости транспортных средств по скорости смены рамки номерного знака;
* нахождение наилучшего числа из последовательности определенных чисел для одного автомобиля (в случае, если номер автомобиля определялся несколько раз);
* ускоренная работа на поисковых зонах;
* использование четырех алгоритмов распознавания чисел для повышения эффективности;
* регулирование скорости распознавания;
* работать как с собственным архивом, так и с другими базами данных.

Методы и технологии распознавания

В пункте пропуска устанавливается видеокамера в соответствии с вышеперечисленными требованиями. Видеосигнал с этой камеры поступает в компьютер с помощью установленной в нем платы видеозахвата и в режиме реального времени обрабатывается программным модулем для считывания и распознавания номерных знаков SL-Traffic в следующей последовательности:

* в зоне охвата камерой обнаружено движущееся транспортное средство;
* производится поиск номерного знака (или нескольких номеров);
* выявленный номер распознается;
* делаются снимки проезжающего автомобиля (по желанию пользователя);
* полученная информация передается для регистрации и архивирования в выбранную по усмотрению пользователя базу данных;
* архив хранится.

Пользователю предлагается 4 типа алгоритмов распознавания чисел [19]. Выбор лучшего из них пользователь делает самостоятельно для достижения наилучшего результата в конкретных условиях применения.

Работа программного модуля SL-Traffic в сочетании с программным обеспечением «SpetLab – Проверка регистрации» также производится в режиме реального времени и практически мгновенно (в течение одной секунды) выполняет следующие операции:

* осуществляет распознавание чисел;
* осуществляет поиск во всех базах данных, существующих в ГАИ;
* при обнаружении преступной информации выдает звуковое оповещение и тревожное сообщение на экран монитора;
* вся полученная информация записывается в журнал.

### 1.2.3 "SecurOS Auto"

Система интеллектуального видеоанализа SecurOS Auto обеспечивает распознавание номерных знаков транспортных средств. SecurOS Auto работает на базе интеграционной платформы управления видео SecurOS, позволяющей создать комплекс безопасности с необходимым заказчиком функционалом (рис.1.5).



Рисунок 1.5 – SecurOS Auto

Область применения:

Сфера деятельности SecurOS Auto охватывает значительный спектр задач от обеспечения безопасности на стоянках до контроля потоков автомобилей по городу.

Возможности

* распознавать номерные знаки более 50 стран мира;
* распознавать номерные знаки по отдельным кадрам без использования видео;
* быстро настроить обновления для распознавания ряда новых стандартов;
* сформировать базу данных распознанных номерных знаков с сохранением соответствующей информации о дате, времени и месте обнаружения
* транспортное средство, скорость и направление его движения, а также ссылка на видеоролик;
* организовать поиск ранее распознанных номерных знаков по заданным параметрам;
* поддерживать одновременно внешние и внутренние списки номеров (информационный, белый, черный);
* автоматически сверять признанный знак государственной регистрации (ОРЗ) по внутренним и внешним спискам (базам данных);
* настроить необходимые реакции системы на распознавание номерных знаков, на результаты поиска;
* оперативно информировать оператора (поддерживается также голосовое оповещение) и/или отправлять уведомления (e-mail, SMS и т.д.) внешним службам о результатах распознавания номерных знаков и/или результатах сравнения данных транспортного средства со списками номеров;
* создать необходимое количество рабочих мест оператора;
* автоматически формировать отчеты разного типа по результатам распознавания, поиска в базе данных.

Методы и технологии распознавания

Модуль локализации генерирует гипотезы о вероятном присутствии номерного знака на изображении (в кадре) и правильно обрабатывает ситуации, когда в кадре имеется несколько номерных знаков. Анализ базируется на поиске участков кадра, имеющих текстовую структуру разностей яркости. В настройках режима распознавания можно указать, производить ли поиск номерных знаков в фиксированных участках рамки.

Положение номерного знака в кадре прогнозируется не только на основе анализа текущего изображения, но и путем экстраполяции результатов распознавания предыдущих кадров.

Масштабирование и бинаризация изображения. Распознавание GRZ осуществляется в несколько этапов анализа двухградационных (бинаризированных) и серых изображений. Область кадра, заданная локализатором, масштабируется, а затем бинаризуется с помощью собственного алгоритма. В результате получается изображение, удобное для быстрого анализа и состоящее только из белых и черных пикселей.

Распознавание чисел. На этапе анализа двоичного изображения происходит обнаружение и распознавание символов номерных знаков, их выравнивание в последовательность символов, а также поиск отсутствующих символов на основе использованных номерных шаблонов определенной страны.

Сравнение изображения с набором шаблонов позволяет избежать ошибочного распознавания «артефактов», определения неправильного количества символов в номерном знаке, неправильного распознавания знаков, близких друг другу и т.п.

SecurOS Auto использует шаблоны, учитывающие геометрию местоположения символов на номерных знаках. Как правило, для правильного наложения шаблона на изображение достаточно распознавания трех-четырех символов в числе. Это позволяет определить расположение остальных символов и распознать их с высокой степенью вероятности. Шаблоны содержат информацию о разрешенных последовательностях букв и цифр, их количестве, физических размерах цифр – соотношении длины и ширины кадров, наличии изображения государственного флага или других графических элементов. SecurOS Auto позволяет одновременно работать с практически неограниченным количеством наборов шаблонов GRZ из разных стран, однако для надежного распознавания по скорости поступления видео рекомендуется использовать шаблоны не более пяти-шести стран.

Реализованный анализ полутонового изображения позволяет значительно повысить точность распознавания. Это особенно важно, если необходимо выбрать между несколькими «конкурирующими» вариантами, которые имеют схожие варианты написания, степень отличия которых может стать еще меньше из-за малого размера символов, низкой контрастности, шума и т.д.

Уточнение результата распознавания, то есть повторное распознавание, выполняется в зависимости от результатов первого прохода. Первоначальное локализованное изображение повторно обрабатывается с различными настройками контрастности, масштабированием и другими типами нормализации в зависимости от обнаруженной проблемы.

Межкадровое слияние результатов распознавания и выдача окончательного результата. Распознавание номерного знака производится на всех кадрах, где его размер и контрастность находятся в заданных пределах, а результаты передаются в модуль «Отслеживание» для межкадрового сравнения, прогнозирование траектории движения номерного знака на последующих кадрах видеоряда. , а также выдача конечного результата (фиксация числа) Модуль "Отслеживание" может одновременно "отслеживать" все числа, найденные в кадре.

Для стабильно хороших результатов достаточно, чтобы номерной знак был виден не менее трех-пяти кадров. Например, одна камера 25-30 кадров в секунду, контролирующая участок проезжей части длиной 8 метров, позволяет получить четыре и более кадров движущегося автомобиля со скоростью 120-180 км/ч. Такого количества кадров, как правило, достаточно для уверенного распознавания всех символов номерного знака.

Качество и скорость распознавания зависят от таких показателей как качество оптики и экспозиции камеры, достаточность освещенности зоны наблюдения, мощность обработки сервера видеоаналитики и т.д. Рецензирование дизайна и лабораторное тестирование перед созданием спецификации проекта сохраняет уровень распознавания на уровне 96-99,9%.

Окончательный результат распознавания выдается в виде «события», доступного всем модулям в «SecurOS». В зависимости от пользовательских настроек, SecurOS сохраняет весь фрагмент проезда транспортного средства или его «лучшую раму», определенную на основе промежуточных результатов распознавания.

SecurOS Auto имеет несколько режимов выдачи результатов распознавания, зависящих от настроек пользователя. Двое из них – «дорогая» и «парковка» предполагают, что камера установлена стационарно, а третья, «мобильная», что камера устанавливается в движущийся, например, патрульный автомобиль.

В режиме "дорога" распознаются только движущиеся номера, а результат выдается, когда номерной знак выходит из рамки или если он "утерян" в кадре (обычно в результате блокировки).

В режиме «парковки» также распознаются фиксированные номера, а результат также выдается в случае остановки транспортного средства, ранее находившегося в движении. В последнем случае "SecurOS Auto" определяет, в какой момент автомобиль с распознанным номерным знаком "вышел из рамки"; это событие можно перенести в систему управления шлагбаумом.

Сохранение результатов распознавания в базе данных и совпадение со списками номерных знаков, и проверка на превышение скорости

Событие распознавания GRZ обрабатывается, среди прочего, в модуле «Контроль» номерной знак принадлежит той или иной полосе, скорость проезжающего транспортного средства определяется по данным радара, результат распознавания сохраняется в базе данных SecurOS Auto и проверяется по внутреннему и внешнему спискам. Результат проверки, например, наличие этого номера в базе краж или превышение разрешенной скорости на этой полосе, также сохраняется в собственной базе данных, и, кроме того, отправляется как событие в SecurOS. В результате в пользовательском интерфейсе «SecurOS Auto» появляется строка «тревожная».

## 1.3 Проверка систем распознавания номерных знаков

Сотрудники испытательной лаборатории журнала CCTV PROSYSTEM [20] протестировали системы распознавания номерных знаков, предоставленные разными компаниями.

Методика проведения тестов, достаточное количество тестового материала, возможность присутствия представителей фирм-конкурентов во время тестирования – исключают субъективность или неоднозначность в интерпретации результатов.

В статье [20], где представлены результаты анализа, вводится понятие "качество системы", K. Качество системы пропорционально количеству правильно распознанных номерных знаков T и обратно пропорционально количеству нераспознанных номерных знаков (100 - T) и "мусор", состоящий из ложных и неправильных распознаваний – F. Таким образом, показатель качества системы можно рассчитать по формуле:



В таблице 1.4 представлен сравнительный анализ популярных систем на рынке программного обеспечения для распознавания номерных знаков.

Таблица 1.4 – Обобщенная таблица результатов из [20]

|  |  |  |  |
| --- | --- | --- | --- |
| Название системы | Количество правильно распознанных номеров (T ), % | «Мусор» (F), % | Качество системы (K) |
| SecurOS Auto | 83,7 | 6,5 | 79 |
| Автомаршал | 89,8 | 14,9 | 59 |
| Spetslab-Traffic | 80,2 | 12,1 | 33 |

Тестирование проводилось на выборке из 1000 чисел, погрешность измерения составляла 3%. О вероятности распознавания можно судить по первому столбцу таблицы. О недостатках систем можно судить по второму столбику.

## 1.4 Итог камеральной проверки

Для парковки и сбора информации лучшим выбором является система SecurOS Auto: 6,5% мусор; 83,7% правильных распознаваний.

Важной отличительной чертой системы SecurOS Auto является наличие специализированных настроек и настроек параметров распознавания, влияющих на результаты распознавания, что позволяет использовать ее как днем, так и ночью, с разными погодными условиями.

Для целей поиска автомобиля лучшая система " Автомаршал": 89,8% правильно зарегистрированных номеров.

## 1.5 Требования к разработанному алгоритму

По результатам испытаний на рынке систем распознавания номерных знаков можно сформулировать основные требования к алгоритму распознавания. Основным показателем качества алгоритма является вероятность распознавания.

Средняя вероятность распознавания составляет более 80%. Таким образом, разработанный алгоритм распознавания должен давать результаты с вероятностью распознавания 80% и выше.

Скорость определения одного номерного знака в существующих программных системах составляет 30–150 мс. Разработанный алгоритм должен иметь скорость распознавания, сравнимую с заданным диапазоном, или работать быстрее.

Элемент классификации должен быть изготовлен на основе аппарата нейронных искусственных сетей.

# 2 АЛГОРИТМ РАЗОЗНАЧЕНИЯ АВТОМОБИЛЬНЫХ НОМЕРОВ НЕЙРОСЕТЬЯМИ

## 2.1 Общее решение задачи

Процесс решения задачи распознавания автомобильного номерного знака при въезде в автостоянку в общих чертах можно представить в последовательности следующих шагов, показанных на рисунке 2.1.

Сегментация номерной пластины

Удаление фрагмента с номерной пластиной

Предварительная обработка вырезанного фрагмента

Классификация сегментов номерной пластины

Рисунок 2.1 – Общая схема решения задачи распознавания автомобильного номерного знака

В данной работе мы предлагаем алгоритм удаления и распознавания цифрового изображения номерного знака.

## 2.2 Поэтапная работа с изображением

Рассмотрим более подробно этапы решения задачи.

### 2.2.1 Сегментация номерных знаков

Для сегментации номерных символов используется U-Net SNS, архитектура сети показана на рис. 2.2.

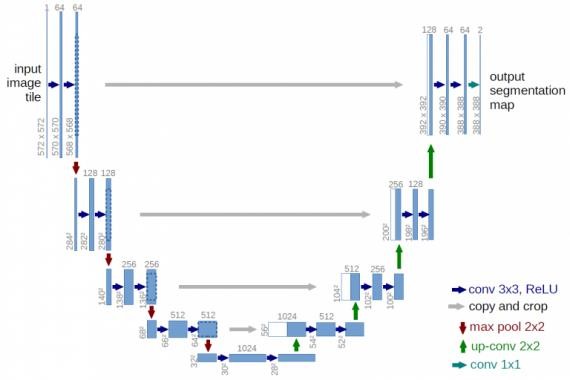


Рисунок 2.2 – Архитектура сети U-Net.

На рис. 2.2 каждый синий квадрат соответствует карте многоканальных свойств. Количество каналов показано в верхней части квадрата. Размер указан в нижнем левом краю квадрата. Белые квадраты являются копиями карты собственности. Стрелки представляют разные операции.

Архитектура сети состоит из пути сужения (слева) и пути расширения (справа). Путь сужения является типичной архитектурой сверточной нейронной сети [13]. Он состоит из повторного применения двух сверток 3x3 с последующей функцией активации ReLU и максимальным объединением (2x2 мощность 2) для снижения разрешения.

На каждом шаге понижения дискретизации функциональные каналы удваиваются. Каждый шаг в пути расширения состоит из операции по повышению дискретизации карты свойств, за которой следуют:

* свертка 2×2, что уменьшает количество функциональных каналов;
* объединены с соответственно обрезанной картой свойств по сокращению пути;
* две складки 3×3, а затем ReLU.

Обрезка необходима из-за потери предельных пикселей с каждой сверткой.

Последний слой использует свертку 1x1 для отображения каждого 64-компонентного вектора признаков на желаемое количество классов [15], [16], [17].

Настройка сети осуществляется путем вычисления коэффициента Диса (также называемого коэффициентом Соренсена-Дайса [23]) или коэффициента подобия Жакара [24], показывающего меру подобия [22] — в этом случае показывает меру площади правильно обозначенные отрезки (отношение площади пересечения к площади, в дальнейшем IoU (Intersection over union) [25].

Вход – это изображение реальной сцены, на которой находится номерной знак автомобиля. Пример входящего изображения на рис. 2.3.



Рисунок 2.3 – изображение, поступающее на вход

Для каждого изображения была создана маска, закрывающая номерной знак.

Пример маски для изображения на рис. 2.3 показано на рис. 2.4.



Рисунок 2.4 – Пример маски

На выходе получаем маску, затем, накладывая ее на исходное изображение, получаем изображение с сегментированным номерным знаком, рис. 2.5.



Рисунок 2.5 – Изображение, полученное на выходе

### 2.2.2 Удаление фрагмента с номерным знаком

Для дальнейшей работы сегментированный номерной знак следует вырезать, получив его координаты.

Для получения координат дальше будем работать с сегментированным фрагментом, маской, полученной на выходе (рис. 2.6).



Рисунок 2.6 – Маска, покрывающая номерной знак, полученная на выходе

К маске применяется двусторонний фильтр, являющийся нелинейным фильтром сглаживания изображений, сохраняющий края. Он заменяет интенсивность каждого пикселя средневзвешенным значением интенсивности соседних пикселей. Этот вес может быть основан на распределении Гаусса. Важно, что весы зависят не только от евклидового расстояния в пикселях, но и от радиометрических отличий (например, разницы в диапазоне, такие как интенсивность цвета, расстояние глубины и т.п.). Это дает возможность сохранить острые края [26], [27], [28], изменения практически незаметны глазу.

Затем к отфильтрованной маске применяют метод нахождения особых точек – детектор Харриса, являющийся оптимальным детектором связанных углов [29, 30]. Затем создается новое изображение и полученные точки накладываются на найденные координаты, результат на рисунке 2.7.

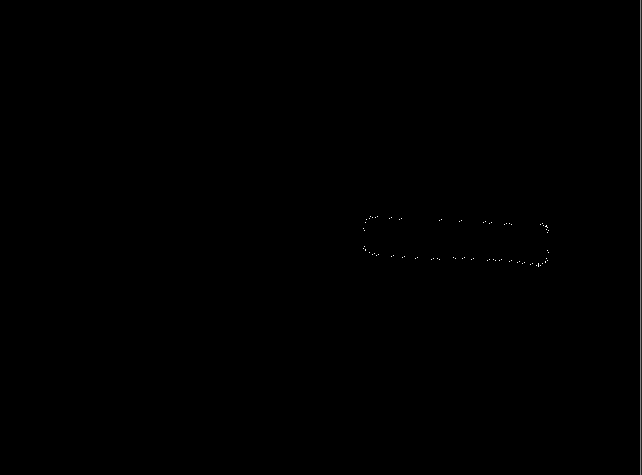


Рисунок 2.7 – Результат поиска особых точек

Белые пиксели – это области возможных углов. Вы можете найти много углов, примыкающих друг к другу. Чтобы найти четыре необходимых, вычисляется расстояние каждого угла друг к другу. Сохраняются только те углы, расстояние которых превысило определенный порог, который меняется до тех пор, пока не будут найдены лишь четыре необходимые точки.

После получения углов они могут быть не в правильном порядке, сортируем в правильном порядке и строим новое изображение, используя полученные точки – перспектива [31], рис. 2.8, с помощью набора инструментов библиотеки OpenCV [33], [33].



Рисунок 2.8 – Вырезанный номерной знак

### 2.2.3 Предварительная обработка вырезанного фрагмента

Далее следует получить каждый отдельный буквенно-цифровой символ.

Превратить в градации серого. Grayscale (grayscale, grayscale [англ. Grayscale]) – цветовой режим изображений, отображаемых в оттенках серого, размещены в виде таблицы как эталоны яркости белого цвета. Чаще используется ступенчатое изображение равномерного ряда оптических плотностей нейтральных серых полей. Он широко используется в цветоведении и колористике, для оценки и измерения качества воспроизведения тона в фотосъемке, процессах сканирования, копирования и печати (печать).

Для преобразования цветных изображений в градации серого используется набор инструментов библиотеки OpenCV [34]. Результат преобразования показан на рис. 2.9.

****

Рисунок 2.9 – Результат изменения цветового пространства (изображение в оттенках серого)

Пороговая обработка и бинаризация.

Далее изображение ограничивается и бинаризируется. Результат бинаризации изображения номерного знака приведен на рис. 2.10.



Рисунок 2.10 – Результат порогового значения

Исключение неинформативных областей.

Не все области представляют интерес для проблемы распознавания номерных знаков. С помощью метода связанных компонентов мы отделяем сегменты друг от друга, заменяя значения пикселей каждого отдельного сегмента определенными значениями [35], а затем удаляем объекты с площадью меньше заданного значения, рис. 2.11.



Рисунок 2.11 – Результат исключения неинформативных областей

Извлечение сегментов.

На следующем этапе все отдельные сегменты вырезаются для дальнейшего распознавания. На рис. 2.12 показан результат извлечения сегментов из изображения номерного знака.



Рисунок 2.12 – Результат извлечения отрезков

Каждый извлеченный сегмент уменьшается до 28×28 пикселей, чтобы сформировать вектор признаков каждого буквенно-цифрового символа номерного знака.

### 2.2.4 Классификация сегментов номерных знаков

Задачей классификации является задача отнесения изображения к одному из представленных множеств [36]. Каждая область изображения номерного знака, полученная в результате сегментации, представляет собой алфавитно-цифровое изображение из алфавита действительных символов.

Архитектура искусственной нейронной сети для классификации буквенно-цифровых символов представлена следующим образом:

* 1 слой свертки с окном свертки 3×3, с функцией активации ReLU
* 1 скрытый полностью подключенный уровень с функцией активации ReLU
* полностью подключенный уровень, с функцией логистической активации на выходе сети.

На вход подается набор буквенно-цифровых изображений, на выходе каждое из которых относится к определенному классу, затем формируется конечный результат, представляющий номер автомобиля в текстовом формате.

Для увеличения точности употребляется ансамбль искусственных нейронных сетей (ШНМ).

Для обучения ANS (комитета) используется алгоритм мешивания. Алгоритм является одним из основных алгоритмов, при котором решение комитета принимается путем усреднения решений отдельных классификаторов.

Для обучения классификаторов используется метод разбиения обучающей выборки. Суть метода состоит в том, чтобы разделить обучающую выборку на M равных неперекрывающихся частей в соответствии с количеством использованных классификаторов, причем каждый классификатор обучается без одной части данных. Таким образом, первый классификатор учится без первой части, второй без второй и т.д. Такой подход гарантирует обучение каждого классификатора на разных данных.

Исходный сигнал ансамбля БНМ при использовании алгоритма беггинга рассчитывается по формуле:



где  – выход i-й ИНС, M – количество ИНС, образующих ансамбль.

В целевую функцию вводится дополнительный штраф за коррелирование выходных сигналов ИНС:



где  – ошибка i-й ИНС на n-м учебном примере,  – необходим выходной сигнал для n-го учебного примера, N – размер обучающей выборки,  – параметр, регулирующий влияние штрафа.

Второе слагаемое вычисляется следующим образом:



Минимизация целевой функции (ЦФ) для ансамбля соответствует минимизации целевой функции каждой сети в отдельности. Формирование ансамбля происходит с использованием однородных архитектур.

# 3 РЕАЛИЗАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

В настоящее время существует огромное количество языков программирования, а также программных библиотек (далее – библиотеки) для работы с искусственными нейронными сетями (ШНМ) и генетическими алгоритмами (ГА), реализуемыми на разных языках программирования, например, C++, C#, Python и т.д.

## 3.1 Язык разработки

В качестве инструмента решения задач и реализации алгоритмов был выбран язык программирования Python.

Python – высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и восприятие кода. Синтаксис ядра Python минималистичен. В то же время, стандартная библиотека включает в себя большой объем полезных функций [15].

Python поддерживает структурное, обобщенное, объектно-ориентированное, функциональное и аспектно-ориентированное программирование. Основные архитектурные черты – динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений, высокоуровневые структуры данных и т.д. Поддерживается разбиение программ на модули, которые могут объединяться в пакеты.

Разработка языка Python началась в конце 1980-х годов сотрудником голландского института CWI Гвидо ван Россумом [16]. Для распределенной ОС Amoeba нужно было создать расширяемый скриптовый язык, и Гвидо начал писать Python как хобби, позаимствовав некоторые наработки для языка ABC (Гвидо участвовал в разработке этого языка, ориентированного на обучение программированию). В феврале 1991 года разработчик опубликовал исходный текст в новостной группе alt.sources. С самого начала Python проектировался как объектно-ориентированный язык.

Развитие языка происходит согласно четко регламентированному процессу создания, обсуждения, отбора и реализации документов PEP (Python Enhancement Proposal) – предложений по развитию Python.

Следует также отметить, что название языка программирования произошло не от названия семейства пресмыкающихся. Автор назвал язык в честь популярного британского комедийного телешоу 1970-х «Летающий цирк Монти Пайтона»[17]. Впрочем, все равно название Python чаще связывают именно со змеей, чем с передачей — значки файлов в KDE или Microsoft Windows и даже эмблема на сайте python.org (до выхода версии 2.5) изображают змеиные головы. Важная цель разработчиков Python – создать язык программирования, интересный для использования. Это отражено в его названии, пришедшем из Монти Пайтона.

Пример программы на языке программирования Python приведен на рис. 3.1.

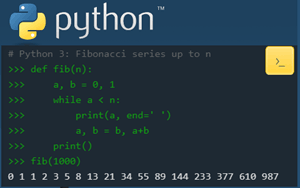


Рисунок 3.1 – Пример программы на языке Python

OpenCV (Open Source Computer Vision Library) – библиотека с открытым кодом функций и алгоритмов для компьютерного зрения, обработки изображений и числовых алгоритмов общего назначения.

OpenCV создан для разработки общей инфраструктуры программ компьютерного зрения и ускорения использования машинного восприятия в коммерческих продуктах. Будучи лицензированным продуктом BSD, OpenCV облегчает предприятиям использование и изменение кода [18].

Библиотека содержит более 2500 оптимизированных алгоритмов, включая полный набор как классического, так и современного компьютерного зрения, и алгоритмов машинного обучения. Их можно использовать для обнаружения и распознавания лиц, идентификации объектов, классификации действий человека на видео, отслеживания движений камеры и движущихся объектов, извлечения 3D-моделей объектов, получения точек 3D облаков из стереокамер, общего захвата изображений для получения изображения высокого разрешения. всю сцену, находите похожие изображения из базы данных изображений, удаляйте красные глаза со вспышек, отслеживайте глаза, распознавайте пейзажи и устанавливайте маркеры для покрытия дополненной реальности и многое другое.

OpenCV имеет более 47 000 пользователей сообщества и более 18 миллионов загрузок. Библиотека широко используется компаниями, научными группами и государственными учреждениями.

Наряду с такими известными компаниями, как Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota, использующими библиотеку, много стартапов, таких как Applied Minds, VideoSurf и Zeitera, которые широко используют OpenCV.

OpenCV имеет интерфейсы C++, Python, Java и MATLAB и поддерживает Windows, Linux, Android и MacOS. В основном он ориентирован на программы просмотра в реальном времени и использует инструкции MMX и SSE, если они доступны.

В настоящее время активно разрабатываются полнофункциональные интерфейсы CUDA и OpenCL. Существует более 500 алгоритмов и примерно в 10 раз больше функций, составляющих или поддерживающих эти алгоритмы.

OpenCV написан на C++ и имеет шаблонный интерфейс, который легко работает с контейнерами STL [12].

Библиотека os

Модуль os из стандартной библиотеки языка программирования Python обычно используется для работы с установленной операционной системой, а также файловой системой ПК. Он содержит множество полезных методов для взаимодействия с файлами и папками на жестком диске. Программы, работающие с модулем os, не зависят от типа ОС и легко переносятся на другую платформу.

Модуль os в Python – это библиотека функций для работы с операционной системой [12]. Методы, включенные в нее, позволяют определять тип операционной системы, получать доступ к переменным окружениям, управлять директориями и файлами:

* проверка существования объекта согласно заданному пути;
* определение размера в байтах;
* удаление;
* переименование и другое.

При вызове функций os нужно учесть, что некие из них могут не поддерживаться текущей операционной системой.

Чтобы использовать методы с os, необходимо подключить библиотеку. Для этого в Python используется import os, который нужно описать в файле до первого обращения к модулю. Рекомендуется использовать это руководство в начале файла с исходным кодом. Пример реализации модуля изображен на рис. 3.2.



Рисунок 3.2 – Пример получения информации об ОС

## 3.2 Среда разработки

Для реализации платформы для разработки Telegram-бота для обеспечения информационной поддержки распознавания текста была выбрана среда PyCharm.

PyCharm – это самая стабильная интеллектуальная Python IDE с полным набором средств для эффективной разработки приложений на языке программирования Python. Выпускается в двух вариантах – бесплатная версия PyCharm Community Edition и расширенная версия приложения по подписке, поддерживающая больший набор возможностей PyCharm Professional Edition. PyCharm выполняет инспекцию кода на лету, автодополнение, в том числе основываясь на информации, полученной при выполнении кода, навигации по коду, обеспечивает множество рефакторингов и т.д. [17].

PyCharm также представляет собой интегрированную среду разработки для языка программирования Python. Предоставляет средства для анализа кода, графический отладчик, инструмент для запуска юнит-тестов и поддерживает веб-разработку на Django. PyCharm разработан компанией JetBrains на основе IntelliJ IDEA. Более 10 лет разработки платформы IntelliJ предоставляет PyCharm более 50 самых разных плагинов, включая поддержку дополнительных VCS, интеграции с различными инструментами и фреймворками, редактором обновлений, таким как эмуляция Vim.

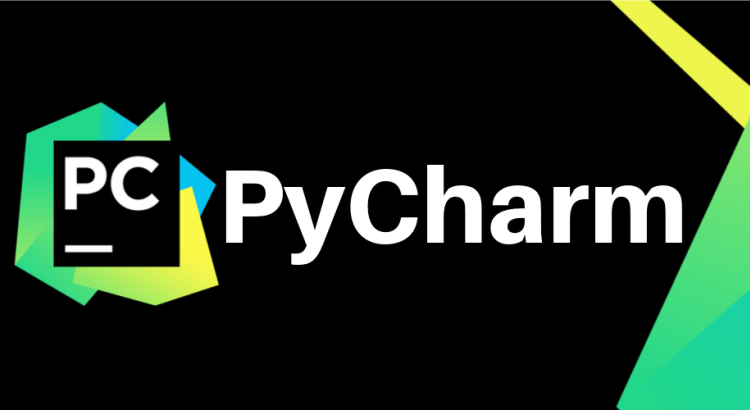


Рисунок 3.3 - PyCharm

Ключевые возможности среды разработки:

* мощный и функциональный редактор кода с подсветкой синтаксиса, авто-форматированием и авто-отступами для поддерживаемых языков;
* простая и мощная навигация в коде;
* помощь при написании кода, включающего в себя автодополнение, авто-импорт, шаблоны кода, проверка на совместимость версии интерпретатора языка и многое другое;
* быстрый просмотр документации для любого элемента прямо в окне редактора, просмотр внешней документации через браузер, поддержка docstring – генерация, подсветка, автодополнение и многое другое;
* большое количество инспекций кода;
* мощный рефакторинг кода, предоставляющий широкие возможности выполнения быстрых глобальных изменений в проекте [17].

## 3.3 Разработка программы

Для начала работы были подключены следующие библиотеки

import os  
from datetime import timedelta  
from pathlib import Path  
import cv2  
import numpy as np  
import pytesseract as tess  
  
from PyQt6 import uic  
from PyQt6.QtGui import QPixmap  
from PyQt6.QtWidgets import QApplication, QFileDialog  
from imutils import contours  
from moviepy.editor import VideoFileClip  
import json

Для разбиения видео на фрагменты была создана следующая часть кода

def split():  
 video\_clip = VideoFileClip(video\_file)  
 filename, \_ = os.path.splitext(video\_file)  
  
 if not os.path.isdir(filename):  
 os.mkdir(filename)  
  
 saving\_frames\_per\_second = min(video\_clip.fps, SAVING\_FRAMES\_PER\_SECOND)  
 step = 1 / video\_clip.fps if saving\_frames\_per\_second == 0 else 1 / saving\_frames\_per\_second  
  
 for current\_duration in np.arange(0, video\_clip.duration, step):  
 frame\_duration\_formatted = format\_timedelta(timedelta(seconds=current\_duration)).replace(":", "-")  
 frame\_filename = os.path.join(filename, f"frame{frame\_duration\_formatted}.jpg")  
  
 video\_clip.save\_frame(frame\_filename, current\_duration)  
 print("split")

После того как видео разобьется на много изображений, для распознавания изображения была создана отдельная функция.

После обработки изображений для вывода информации была написана следующая часть кода

def recognize():  
 load\_file()  
 global result  
 image = cv2.imread(image\_file)  
 height, width, \_ = image.shape  
 gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)  
 thresh = cv2.threshold(gray,0,255,cv2.THRESH\_OTSU)[1]  
 cnts = cv2.findContours(thresh, cv2.RETR\_TREE, cv2.CHAIN\_APPROX\_SIMPLE)  
 cnts, \_ = contours.sort\_contours(cnts[0])  
  
 for c in cnts:  
 area = cv2.contourArea(c)  
 x,y,w,h = cv2.boundingRect(c)  
 if area> 20000:  
 img = image[y:y+h, x:x+w]  
 gray2 = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)  
 thresh2 = cv2.threshold(gray2, 0, 255, cv2.THRESH\_OTSU)[1]  
 res = tess.image\_to\_string(thresh2)  
 if len(res)>7:  
 form.label\_7.setText(res)  
 result=res

В зависимости от количества посещений разработана соответствующая схема скидок

count = 0;  
for i in range(len(arr)):  
 if arr[i][0] == result:  
 count += 1  
if count == 0:  
 arr.append([result, 1])  
else:  
 for i in range(len(arr)):  
 if arr[i][0] == result:  
 arr[i][1] += 1  
for i in range(len(arr)):  
 if arr[i][0] == result:  
 form.label\_2.setText(str(arr[i][1]))  
 if (arr[i][1]<5):  
 form.label\_5.setText('0%')  
 elif (arr[i][1]>=5 and arr[i][1]<10):  
 form.label\_5.setText('5%')  
 elif (arr[i][1]>=10 and arr[i][1]<20):  
 form.label\_5.setText('10%')  
 else:  
 form.label\_5.setText('15%')

На каждую отдельную кнопку было создано соответствующее событие

form.pushButton.clicked.connect(load);  
form.pushButton\_2.clicked.connect(split);  
form.pushButton\_3.clicked.connect(choose);  
form.pushButton\_4.clicked.connect(recognize);

Также для кадрового разбиения была создана следующая функция

def format\_timedelta(td):  
 result = str(td)  
 try:  
 result, ms = result.split(".")  
 except ValueError:  
 return result + ".00".replace(":", "-")  
  
 ms = round(int(ms) / 10000)  
 return f"{result}.{ms:02}".replace(":", "-")

## 3.4 Работа с базой данных

Разработка корпоративных сайтов подразумевает создание БД, которые решат вопросы поиска, классификации и учета полезных сведений, делая рабочий процесс автоматизированным. Без этого ни одна программа и веб-система не смогут исправно и оперативно функционировать.

В процессе задействованы специальные программные средства – системы управления БД (СУБД), позволяющие:

* создавать БД;
* производить доступ к информации;
* вести работу с информацией (добавлять, редактировать, удалять) и обеспечивать безопасность данных.

Разработка баз данных осуществляется для оперирования информацией в реляционной базе данных с помощью СУБД. Качественно структурированные хранилища, функционирующие бесперебойно, способны решать вопросы без привлечения вспомогательных ресурсов со стороны персонала.

В автоматизированных хранилищах предусматриваются вспомогательные опции:

* учет клиентов;
* отслеживание оборота средств и товаров;
* формирование детальной статистики;
* ведение документооборота и т.д.

Разработка программы базы данных позволяет владельцу бизнеса иметь доступ к актуальным сведениям о функционировании всей организации. При привлечении качественной системы управления можно сократить критические нагрузки на серверы предприятия, организуя равномерное распределение, гарантируя уменьшение затрат на электричество, а также обслуживание техники.

Этапы разработки базы данных

Проектирование и разработка базы данных осуществляется в несколько этапов:

* уточняется техническое задание;
* формируется функционирующий пример БД;
* созданный образец запускается в действие;
* в ходе квалифицированного использования корректируются и уточняются требования к базе данных;
* исходя из предоставленной информации, создается БД;
* полученный продукт проходит тестирование, в ходе которого устраняются допущенные ошибки;
* после база данных запускается, и поддерживается в дальнейшем.

При создании программ БД клиенты могут решить следующие задачи:

1. Формирование общего информпространства для работников. Рассматривается возможность подключения к базе и клиентов и партнеров.

2. Разделение доступа к данным.

3. Информирование о реальной картине развития бизнеса.

4. Сокращение временных затрат на поиск необходимых сведений. Достаточно и нескольких секунд с момента ввода запроса.

5. Гарантируется разработка и защита баз данных от вероятной потери сведений, взлома и т.д.

6. Допустим обращение к БД через любое устройство, подключенное к Интернету.

Физическая модель данных, напротив, зависит от конкретной БД, фактически являясь отражением системного каталога. В физической модели содержится информация обо всех объектах БД. Поскольку стандартов на объекты БД нет (например, нет стандарта на типы данных), физическая модель зависит от конкретной реализации СУБД. Следовательно, одной и той же логической модели могут соответствовать несколько разных физических моделей. Если в логической модели не имеет значения, какой именно тип данных имеет атрибут, то в физической модели важно описать всю информацию о конкретных физических объектах – таблицах, колонках, индексах, процедурах и т.д. Разделение модели данных на логические и физические позволяет решить несколько важных задач.

Все данные хранятся в базе данных Sqlite 3. Подключение к базе данных происходило благодаря следующему коду

import sqlite3

db = sqlite3.connect('server.db')  
sql = db.cursor()

Создание таблицы было реализовано следующим образом:

sql.execute("""CREATE TABLE IF NOT EXISTS carwash (number TEXT, countvisit INT)""")

А сама база данных выглядит следующим образом (рис. 3.4)



Рисунок 3.4 – Структура БД(cars)

## 3.5 Результат работы программы

Для запуска программы необходимо открыть файл и запусти в PyCharm. Перед вами предстанет следующая картина (рис. 3.5)

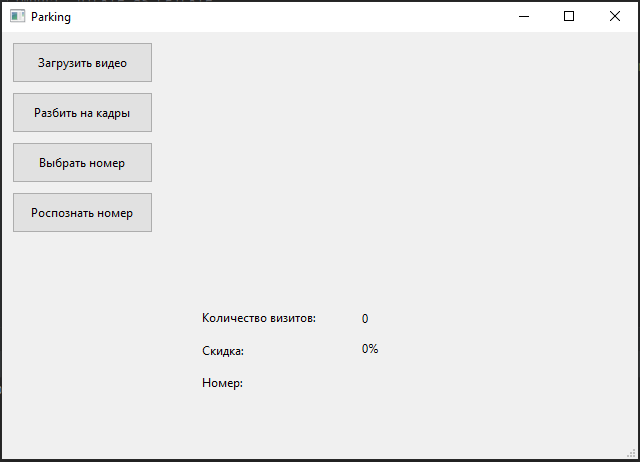


Рисунок 3.5 – начало программы

Далее нажав кнопку «Загрузить видео», и выбрать необходимое видео, пройдет загрузка видео (рис. 3.6)

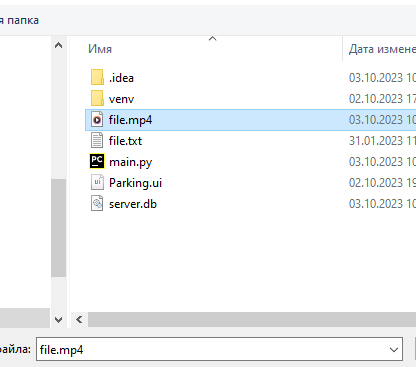


Рисунок 3.6 – загрузка видео

Затем нажав кнопку «Разбить на кадры», программа разобьет видео на кадры и сохранит их в соответствующую папку (рис. 3.7)

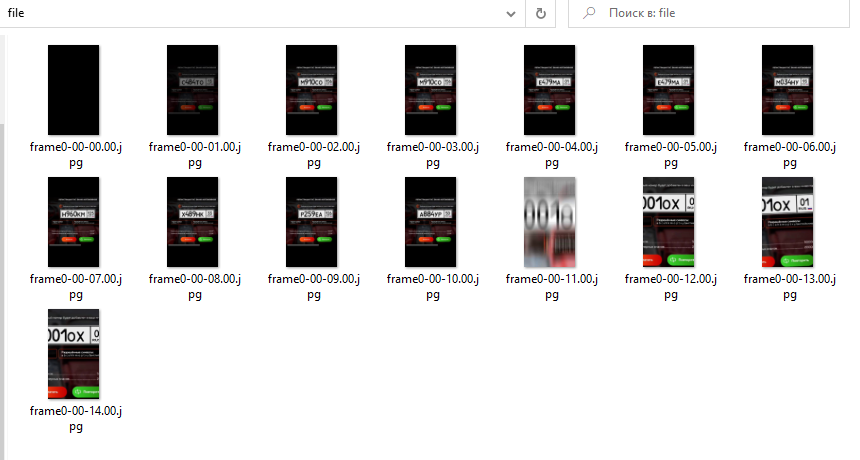


Рисунок 3.7 – Разбиение на кадры

После чего можно заняться распознаванием изображений. Для этого нажмите кнопку «Выбрать номер» и выберите изображение. Перед вами появится выбранное вами изображение (рис. 3.8)

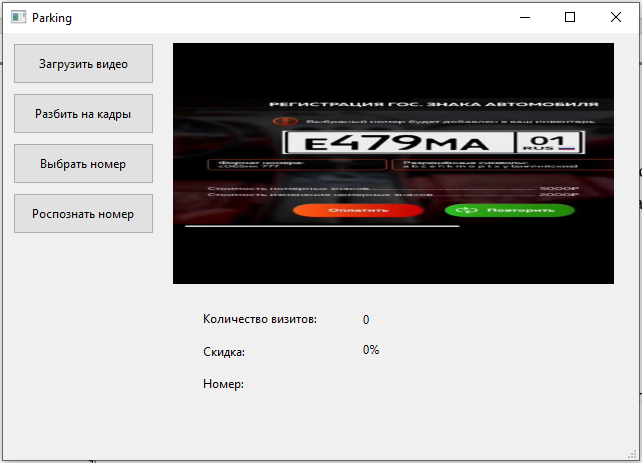


Рисунок 3.8 – загрузка изображения

Далее нажав на кнопку «Распознать номер» перед вами появится следующая информация (рис. 3.9)

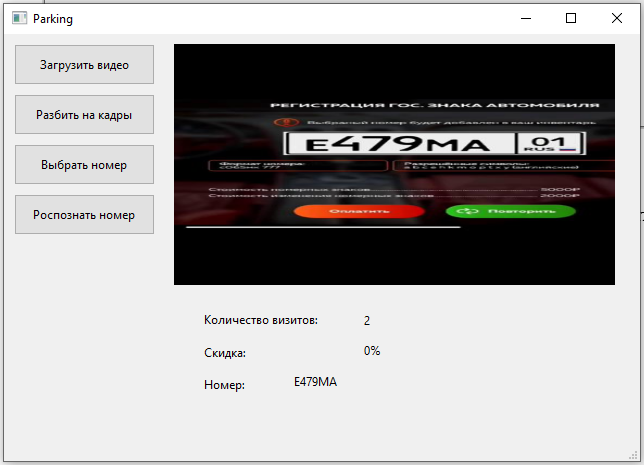


Рисунок 3.9 – Информация об автомобиле

В случае, если автомобиль был на стоянке много раз, то информация соответственно покажет об этом, и покажет скидку (рис. 3.10)

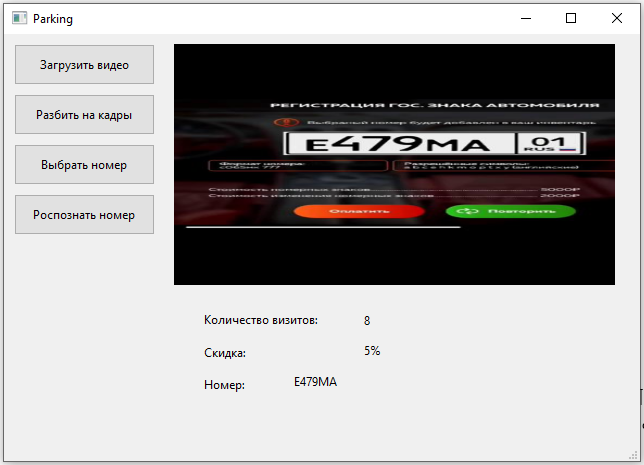


Рисунок 3.10 – Количество визитов 8, скидка 5%

# ЗАКЛЮЧЕНИЕ

Проблему автоматического распознавания номерных знаков в цифровых изображениях решали многие исследователи. В работе рассмотрен собственный алгоритм решения задачи. Системы определения номерных знаков используются для контроля за въездом и выездом транспортных средств с территории предприятий, автостоянок, контроля за транспортным потоком.

Существующие коммерческие системы распознавания номерных знаков не всегда предоставляют пользователям информацию о методах искусственного интеллекта, используемых для распознавания буквенно-цифровых символов. В данной работе был выполнен ряд аналитических задач для определения алгоритмов, используемых в системах. Данные системы получены из независимых открытых источников.

Проведен аналитический обзор существующих методов и программных систем для распознавания автомобильных номеров. Проведен их сравнительный анализ, определен перечень программных продуктов, пригодных для решения определенного круга целевых задач по распознаванию номерных знаков.

Для сегментации номерных знаков используется архитектура U-Net SNA, которая показала отличные результаты сегментации в других областях. Удалось добиться точности 94,8% при сегментации номерного знака на наборе данных теста.

В работе исследуется возможность использования искусственных нейронных сетей, а также их ансамблей в качестве элементов классификации. Сравнительный анализ различных архитектур и размеров комитетов позволяет обосновать выбор классифицирующего элемента в предложенном алгоритме распознавания.

Разработанный нейросетевой алгоритм распознавания номерных знаков позволяет добиться точности распознавания 92,1%.

Результаты получены на выбранных образцах автомобильных номерных знаков, доступных в открытом доступе. Полученная точность распознавания сопоставима с существующими аналогами.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Желтов С.Ю. Обработка и анализ изображений в проблемах машинного зрения. // Москва.: Физматкнига, 2010. – 672 с.

2. Habr Mask R-CNN: архитектура современной нейронной сети для сегментации объектов в изображениях [Электронный ресурс]: – Режим доступа: https://habr.com/post/421299 (дата обращения 27.09.2023).

3. Виола П., Джонс М. Дж. Надежное обнаружение лица в реальном времени/Международный журнал компьютерного зрения. – 2014. – Т. 57. – С. 137-154.

4. Серикова А. С. Разработка и программная реализация алгоритма сегментации и распознавания номерных знаков // Магистерская работа, ТПУ, ИШИТР, ИСТ 2017.

5. Ильин В. А., Позняк Е. Г. Линейная алгебра // Киев Физматлит 2019.

6. Нелло Кристианини, Джон Шоу-Тэйлор. Введение в поддержку векторных машин и других методов обучения на основе ядра // Cambridge University Press, 2020

7. Хайкин С. Нейронные сети: полный курс: пер. с англ. // С. Хайкин. – 2-е изд., переп. – М. [и др.]: Вильямс, 2016. – 1103 с.

8. Комарцова Л. Нейрокомпьютеры: учебник для вузов // Л. Г. Комарцова, А. В. Максимов. – 2-е изд., перераб. и доп.. – М.: Изд-во МГТУ, 2014. – 399 с.

9. Джонс М. Программирование искусственного интеллекта в приложениях // М. Джонс: Пер. с англ. Осипов А.И. – Харьков.: ГТК Пресс, 2013. – 312 с.

10. Каллан Р. Основные понятия нейронных сетей // Пер. с англ. – Издательство М. Уильямса, 2011.

11. Тадеусевич Р. Элементарный ввод в технологию нейронных сетей с примерами программ: Пер. с пола // Р. Тадеусевич [и др.]. – М.: Горячая линия-Телеком, 2011. – 408 с.

12. Осовский С. Нейронные сети для обработки информации: пер. с польского. // С. Осовский. – М.: Деньги и статистика, 2012. – 344 с.

13. Дж. Лонг, Э. Шелхамер, Т. Даррелл: Полностью сверточные сети для семантической сегментации 2014 г.

14. Хабр Что такое свертка нейронная сеть [Электронный ресурс]: - Режим доступа: https://habr.com/post/309508 (дата обращения 27.09.2023).

15. U-Net: нейронная сеть для сегментации изображений [Электронный ресурс]: — Режим доступа: https://neurohive.io/ru/vidy-nejrosetej/u-net-image-segmentation. (дата обращения 27.09.2023).

16. Олаф Роннебергер, Филипп Фишер, Томас Брокс U-Net: сверточные сети для сегментации биомедицинских изображений

17. Глубокое обучение U-Net. [Электронный ресурс]: - Режим доступа: http://deeplearning.net/tutorial/unet.html (дата обращения 27.09.2023)

18. Автомаршал [Электронный ресурс]: уч. веб-сайт. 2014. – Режим доступа: http://avtomarshal/, свободный. Заголовок с экрана (дата обращения 27.09.2023).

19. SL Трафик. Безопасный город [Электронный ресурс]: оф. веб-сайт – Режим доступа: http://goal.ru/, (дата обращения 27.09.2023).

20. CVS Computer Video Security [Электронный ресурс]: офиц. веб-сайт – Режим доступа: http://www.cvsnt.ru (дата обращения 27.09.2023).

21. Адрон. Распознавание номерных знаков: «Автоинспектор» – руководитель тестирования [Электронный ресурс] – Режим доступа: http://www.adron-perm.ru/test\_leader, (дата обращения 27.09.2023).

22. Коэффициент сходства [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/ Коэффициент сходства (дата обращения 27.09.2023).

23. Коэффициент Соренсена [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Sorensen\_coefficient (дата обращения 27.09.2023).

24. Коэффициент Жакара [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org//wiki/Jaccard\_coefficient (дата обращения 27.09.2023).

25. Evaluating image segmentation models [Электронный ресурс] / Blog Jeremy Jordan – Режим доступа: https://www.jeremyjordan.me/evaluating-image-segmentation-models (дата обращения 27.09.2023)