

Исследование возможностей языка программирования Python для создания интеллектуальных систем сканирования багажа

А.В. Каляшина¹, Ю.Н. Смирнов²

¹*Казанский национальный исследовательский технический университет им. А.Н. Туполева – КАИ, Казань, Россия*

²*Казанский государственный энергетический университет, Казань, Россия*

Аннотация: Данное исследование посвящено анализу возможностей языка программирования Python при создании информационных систем обнаружения опасных предметов в багаже. В результате разработана архитектура системы распознавания, включающая следующие основные компоненты: модуль обработки изображений, модуль машинного обучения, базу данных и интерфейс пользователя. В качестве программного обеспечения выбран Python с библиотеками PySide6, Numpy, YOLO. Проведена реализация и тестирование информационной системы на реальных данных, что подтвердило правильность выбранных возможностей и технологий языка Python для разработки информационных систем безопасности.

Ключевые слова: информационная система, безопасность, нейронная сеть, машинное обучение, распознавание образов, быстродействие.

Приоритетной задачей в местах массового скопления людей является обеспечение безопасности. Для этого активно ведутся разработки систем обнаружения опасных предметов. На сегодняшний день существуют разнообразные информационные системы, предназначенные для сбора, анализа и обработки данных, связанных с потенциальными угрозами и обеспечением безопасности людей и объектов инфраструктуры. В системах транспортной безопасности, подобные системы используются для проверки багажа и пассажиров, выявления опасных предметов и предотвращения их провоза.

Автоматическое распознавание подозрительных объектов является одной из самых передовых технологий в области безопасности [1,2]. Системы, основанные на машинном обучении [3] и искусственном интеллекте, могут анализировать изображения и видео в реальном времени, выявляя потенциально опасные предметы. Такие системы обучаются на больших объемах данных и способны распознавать различные типы оружия, взрывчатых веществ и иных опасных предметов с высокой точностью [4]. К

методам распознавания опасных предметов на изображениях относятся: сверточные нейронные сети (CNN), рекуррентные нейронные сети (RNN), методы детекции объектов и сегментации изображений, методы локализации объектов на изображении, методы машинного обучения [5].

В настоящее время активно используются следующие программные средства [4, 5]:

- X-Ray Screener –система, предназначенная для автоматического анализа рентгеновских снимков багажа и обнаружения опасных предметов. Она использует методы глубокого обучения для классификации и локализации объектов внутри багажа;

- Smiths Detection iCMORE –комплексное решение для автоматического анализа и классификации изображений, полученных с помощью рентгеновских и КТ-сканеров. Разработано для обнаружения опасных предметов в багаже и грузах;

- ClearScan –система автоматического анализа изображений, полученных с помощью рентгеновских сканеров, предназначенная для проверки багажа и грузов. Использует передовые алгоритмы машинного обучения для распознавания опасных предметов;

- DeepScan –программное обеспечение, основанное на использовании глубоких нейронных сетей для анализа рентгеновских изображений багажа. Разработано для автоматического обнаружения и классификации опасных предметов.

Отдельной задачей необходимо выделить проблемы, возникающие при получении и обработке изображений. В работе [6] исследована задача повышения разрешающей способности изображений с помощью метода Super-Resolution, а в работе [7] уделено внимание нейросетевому подходу на основе методов глубокого обучения.

Цель данного исследования – анализ возможностей языка программирования Python при разработке системы определения потенциально опасных предметов в багаже [8]. В данной статье представлена архитектура и реализация базовых модулей системы, без описания нейронной сети. Предварительно были установлены следующие требования к информационной системе:

- 1) обнаружение различных типов опасных предметов в багаже;
- 2) функционирование в режиме реального времени;
- 3) вывод результатов сканирования на экран оператора;
- 4) автоматическая регистрация опасных предметов в базе данных;
- 5) гибкая архитектура, позволяющей адаптироваться к различным условиям эксплуатации и типам угроз;
- 6) высокий уровень защиты данных и конфиденциальности личной информации пассажиров;

Структура разработанной системы включает следующие основные компоненты:

- 1) блок предварительной обработки изображений: осуществляет предварительную обработку, включая уменьшение шума, коррекцию яркости и контрастности; принимает входные изображения с интроскопов или других устройств сканирования багажа; подготавливает изображения для последующего анализа и распознавания.

- 2) модуль распознавания опасных предметов: включает в себя нейронную сеть, обученную на размеченных данных опасных предметов, таких как колюще-режущие, огнестрельные, взрывчатые, патроны; проводит классификацию объектов на изображениях идентифицированных предметов в соответствии с заданными категориями опасных предметов; определяет вероятность присутствия каждого опасного предмета на сканируемом изображении.

3) Система управления и принятия решений: анализирует результаты работы модуля распознавания опасных предметов; принимает решения о дальнейших действиях на основе служб; может включать в себя механизмы автоматического оповещения служб безопасности или операторов о выявленных предметах.

4) модуль персонализации и настройки: позволяет администраторам системы конфигурировать параметры обнаружения в соответствии с требованиями конкретной локации или типа установки; обеспечивает возможность адаптации системы к различным условиям эксплуатации и требованиям безопасности.

Для разработки системы используются следующие библиотеки языка программирования Python [9,10]:

- cv2 - OpenCV - использованы инструменты для обработки изображений и видеопотоков в реальном времени;
- YOLO – наиболее эффективная и быстрая модель для обнаружения объектов на изображениях, а также для детекции объектов;
- PIL – конвертация форматов, изменение размера и сохранение изображений;
- ImageQt - ImageQt интегрирование изображений в графический интерфейс, созданный с использованием PySide6;
- Numpy - выполнение численных операций и обработки массивов данных (изображений и результатов детекции);
- SQLite (через собственный модуль db) - SQLite позволяет сохранять данные в формате SQL. При этом не требуется настройка сервера базы данных, что особенно важно для небольших приложений, таких как данное.

Для обеспечения высокой эффективности системы уделено внимание масштабируемости и производительности. Система должна быть способна обрабатывать большие объемы данных в реальном времени и оперативно

реагировать на обнаруженные угрозы, что требует оптимизации алгоритмов и использования высокопроизводительных вычислительных ресурсов.

Таким образом, проектирование и разработка системы распознавания опасных предметов в багаже требует комплексного подхода, учитывающего различные аспекты, начиная от выбора алгоритмов и технологий до тестирования, внедрения и обучения персонала. Ниже описаны основные этапы разработки информационной системы и их результаты:

1) Разработка системы распознавания.

Использован метод `process_stream`, который обрабатывает видеопоток с камеры и выполняет распознавание объектов с использованием модели YOLO. Результаты распознавания затем отображаются в таблице интерфейса и сохраняются в базе данных. Для визуализации обнаруженных объектов использован метод `plot`.

2) Разработка интерфейса системы распознавания.

В окне интерфейса отображаются скрины найденных опасных или запрещенных предметов, их класс, уникальный идентификатор (ID) и вероятность уверенности, предоставленная нейронной сетью. Эта информация позволяет оператору быстро и точно определить наличие опасных предметов в багаже. При создании интерфейса использован метод `show_table`, который создает таблицу с четырьмя колонками для отображения ID, изображения объекта, класса и уверенности модели в распознавании

Для создания окна для отображения видеопотока с камеры и настройки его свойств использован метод `show_scanner`. На рис.1 представлено главное окно разработанного интерфейса пользователя.

3) Разработка базы данных системы распознавания.

В данном исследовании использована SQLite - компактная библиотека, которая не требует настройки сервера или сложного администрирования. SQLite хранит данные в виде единого файла базы данных, что упрощает

резервное копирование и восстановление, что важно для безопасности данных, так как сканы опасных предметов могут содержать критически важную информацию для обеспечения безопасности публичных мест. Для создания таблицы сканов использован метод `create_scans_table`, для создания таблицы сканов использован метод `create_objects_table`, для создания базы данных использован метод `create_db`. На рис. 2 представлена схема разработанной базы данных

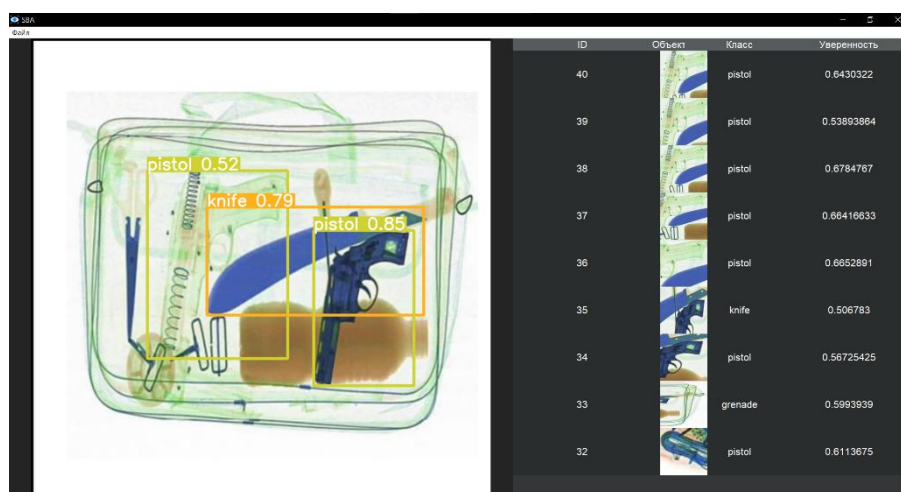


Рис. 1. - Интерфейс программы

4) Разработка таблицы объектов системы распознавания - таблица включает в себя идентификаторы объектов, изображения (скриншоты) обнаруженных предметов, их классификацию и уровень уверенности системы в правильности этой классификации.

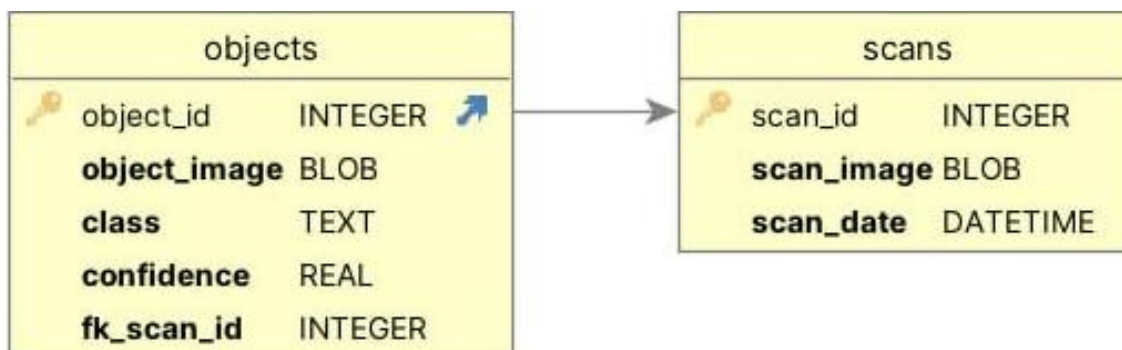


Рис. 2. - Схема разработанной базы данных

Главное окно приложения создает класс `MainWindow`. Таблица настраивается для удобного отображения данных, а окно приложения имеет фиксированные минимальные размеры, иконку и заголовок.

В результате данного исследования можно сделать вывод о том, что язык программирования Python позволяет создавать практически все модули информационной системы сканирования багажа. В ходе исследования была разработана архитектура системы распознавания с использованием библиотек языка Python, включающая ключевые компоненты: модуль обработки изображений, модуль машинного обучения, базу данных для хранения результатов и интерфейс для операторов. Реализация и тестирование этих модулей показали высокую точность (97%) работы системы. Тестирование проводилось на реальных наборах данных и подтвердило правильность выбранных методов и технологий, продемонстрировав высокие показатели точности распознавания опасных предметов.

Литература

1. Пратик Д. Искусственный интеллект с примерами на Python. - Санкт-Петербург: Диалектика, 2019. – 448 с.
2. Барский А. Б. Искусственный интеллект и интеллектуальные системы управления: монография — Москва: Русайнс, 2022. — 185 с. — ISBN 978-5-4365-8166-8.
3. Игнатьева О. В., Сокирка А. Д., Журавлев Д. С. Применение методов машинного зрения на встраиваемых системах // Инженерный вестник Дона. – 2024. – № 1. URL: ivdon.ru/ru/magazine/archive/n1y2024/8918.
4. Нуриев М. Г., Белашова Е. С., Барабаш К. А. Конвертер Markdown-файлов в LaTeX-документ // Программные системы и вычислительные методы. – 2023. – № 1. – С. 1-12.

5. Прохоренок Н.А. OpenCV и Java. Обработка изображений и компьютерное зрение – Санкт-Петербург: БХВ-Петербург, 2018. – 320 с.
6. Смирнов Ю.Н., Каляшина А.В. Использование нейросетевых методов в задаче повышения разрешения Super Resolution // Научно-технический вестник Поволжья. – 2023. – № 10. – С. 55-57.
7. Смирнов Ю.Н., Каляшина А.В., Абдуллин А.И. Нейронные сети для анализа биомедицинских изображений // Научно-технический вестник Поволжья. – 2023. – № 6. – С. 106-109.
8. Кириченко А. А. Практикум по нейропакетам: монография. — Москва: Русайнс, 2019. — 253 с. — ISBN 978-5-4365-3507-4.
9. Жорняк А. Г., Морозова Т. А. Разработка визуального пользовательского интерфейса в программах для научных и инженерных вычислений на языке программирования Python. Часть IV. Библиотека Tkinter // Научно-технический вестник Поволжья. – 2024. – № 1. – С. 73-76.
10. Храмов В. В., Митясова О. Ю. Способ формирования ортогональных признаков распознавания объектов на плоских изображениях // Вестник кибернетики. – 2024. – Т. 23, № 2. – С. 76-80.

References

1. Pratik D. Iskusstvennyj intellekt s primerami na Python Dialektika [Artificial Intelligence with Examples in Python]. Sankt-Peterburg: Dialektika, 2019. 448 p.
 2. Barskij A. B. Iskusstvennyj intellekt i intellektual'nye sistemy upravleniya [Artificial intelligence and intelligent control systems]. Moskva: Rusajns, 2022. 185 p.
 3. Ignat'eva O. V., Sokirka A. D., Zhuravlev D. S. Inzhenernyj vestnik Dona. 2024. № 1. URL: ivdon.ru/ru/magazine/archive/n1y2024/8918.
-



4. Nuriev M. G., Belashova E. S., Barabash K. A. Programmnye sistemy i vychislitelnye metody. 2023. № 1. Pp. 1-12.
5. Prohorenok N.A. OpenCV i Java. Obrabotka izobrazhenij i komp'yuternoje zrenie [OpenCV and Java. Image processing and computer vision]. Sankt-Peterburg: BHV-Peterburg, 2018. 320 p.
6. Smirnov Yu. N., Kalyashina A. V. Nauchno-tehnicheskij vestnik Povolzhya. 2023. № 10. Pp. 55-57.
7. Smirnov Yu.N., Kalyashina A.V., Abdullin A.I. Nauchno-tehnicheskij vestnik Povolzhya. 2023. № 6. Pp. 106-109.
8. Kirichenko A. A. Praktikum po nejropaketam [Workshop on neuropackets]. Moskva: Rusajns, 2019. 253 p.
9. Zhornyak A. G., Morozova T. A. Nauchno-tehnicheskij vestnik Povolzhya. 2024. № 1. pp. 73-76.
10. Hramov V. V., Mityasova O. Yu. Vestnik kibernetiki. 2024. T. 23, № 2. Pp. 76-80.

Дата поступления: 4.06.2024

Дата публикации: 18.07.2024