

УДК 004.42:004.93

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ СКАНЕРА QR-КОДОВ СРЕДСТВАМИ PYTHON

Лыткина Е.А., ученица 10-Б класса

Научный руководитель: Рябко Е.В. к.т.н., доцент

Муниципальное бюджетное общеобразовательное учреждение
«Отраденская СОШ №2» Новоусманского района, Воронежской области

В настоящей статье рассматривается процесс разработки приложения для сканирования и декодирования QR-кодов с использованием языка программирования Python. Программа предоставляет пользователю два основных режима работы: сканирование в реальном времени с веб-камеры и распознавание кодов из загруженных графических файлов форматов PNG и JPG. В работе описываются дополнительные функции, такие как копирование ссылок в буфер обмена и открытие веб-страниц, а также рассматриваются потенциальные направления для дальнейшего улучшения программного продукта.

Введение

В современном мире двумерные матричные штрихкоды (QR-коды) стали неотъемлемой частью повседневной коммуникации, маркетинга и логистики. Они позволяют быстро передавать текстовую информацию, ссылки на веб-ресурсы и другие данные. Разработка собственного программного обеспечения для чтения таких кодов является актуальной задачей, позволяющей понять принципы компьютерного зрения и взаимодействия программных компонентов [1]. В данной работе представлено описание приложения, созданного на Python, которое совмещает в себе функции сканера и декодера, а также обладает удобным графическим интерфейсом [2].

Предметом исследования являются программные и алгоритмические средства распознавания QR-кодов в среде Python.

Объект исследования: десктопное приложение для сканирования QR-кодов, интегрирующее модули компьютерного зрения, графический интерфейс пользователя и инструменты обработки данных.

Цели и задачи: целью данной работы является описание архитектуры и функциональных возможностей приложения-сканера QR-кодов.

Для достижения поставленной цели решаются следующие задачи:

1. Анализ ключевых библиотек, используемых для распознавания и обработки изображений.
2. Описание логики работы приложения при сканировании с веб-камеры и из файлов.
3. Характеристика графического интерфейса и его элементов управления.

4. Определение векторов дальнейшего развития и модернизации программы.

Функциональность приложения обеспечивается за счет интеграции нескольких специализированных библиотек, каждая из которых выполняет строго определенную задачу. Основные библиотеки и их назначение представлены в таблице 1.

Библиотеки	Назначение
OpenCV(cv2)	Работа с камерой и изображениями
PyZBar	Декодирование QR-кодов
Tkinter	Графический интерфейс (GUI)
Pillow (PIL)	Обработка изображений
Webbrowser	Открытие ссылок в браузере

Таблица 1 – Основные библиотеки и их назначение

OpenCV выступает в роли основного инструмента для получения данных. При сканировании с камеры библиотека захватывает каждый кадр видеопотока. В случае работы с файлом, OpenCV или Pillow отвечают за его загрузку. PyZbar, построенная на основе библиотеки ZBar, принимает подготовленное изображение и ищет в нем области, содержащие QR-код, возвращая декодированную информацию [3, 4].

Основная функциональность приложения.

Приложение предлагает пользователю два ключевых режима работы, покрывающих большинство сценариев использования.

Сканирование с веб-камеры в реальном времени (рисунок 1).

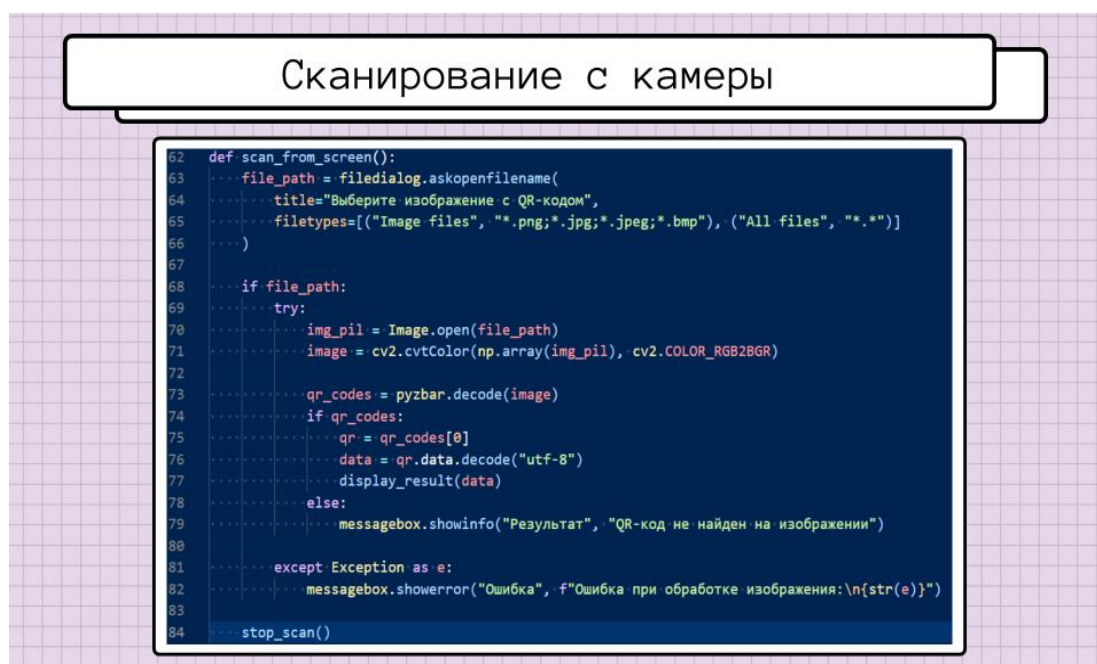


Рисунок 1 – Диалоговое окно, отображающее данные, полученные с камеры

Данный режим инициирует захват видео с камеры устройства. Каждый полученный кадр преобразуется в оттенки серого (grayscale), что повышает эффективность работы детектора. Затем кадр передается в PyZBar для анализа. Как только на кадре обнаруживается QR-код, система мгновенно его декодирует. Если распознанная строка является ссылкой (начинается с `http://` или `https://`), приложение автоматически открывает её в браузере. В противном случае содержимое выводится на экран в виде обычного текста.

Сканирование из загруженного изображения. Пользователь может выбрать на диске файл в формате PNG или JPG. Программа загружает его с помощью Pillow, при необходимости конвертирует в формат, понятный для PyZbar, и выполняет декодирование. Логика обработки результата (текст или ссылка) идентична режиму работы с камерой.

Графический интерфейс, реализованный на Tkinter, спроектирован с акцентом на минимализм и удобство. Для обеспечения аккуратного внешнего вида в проекте строго соблюдаются правила верстки: отступы между всеми кнопками и элементами управления унифицированы и составляют 5 пикселей (рисунок 2). Исключение сделано для кнопки «Стоп», отступ от которой увеличен до 10 пикселей для визуального выделения и предотвращения случайных нажатий. Горизонтальные отступы между кнопками внутри логических групп также симметричны и равны 5 пикселям, что создает ощущение упорядоченности и структурированности [2, 4].

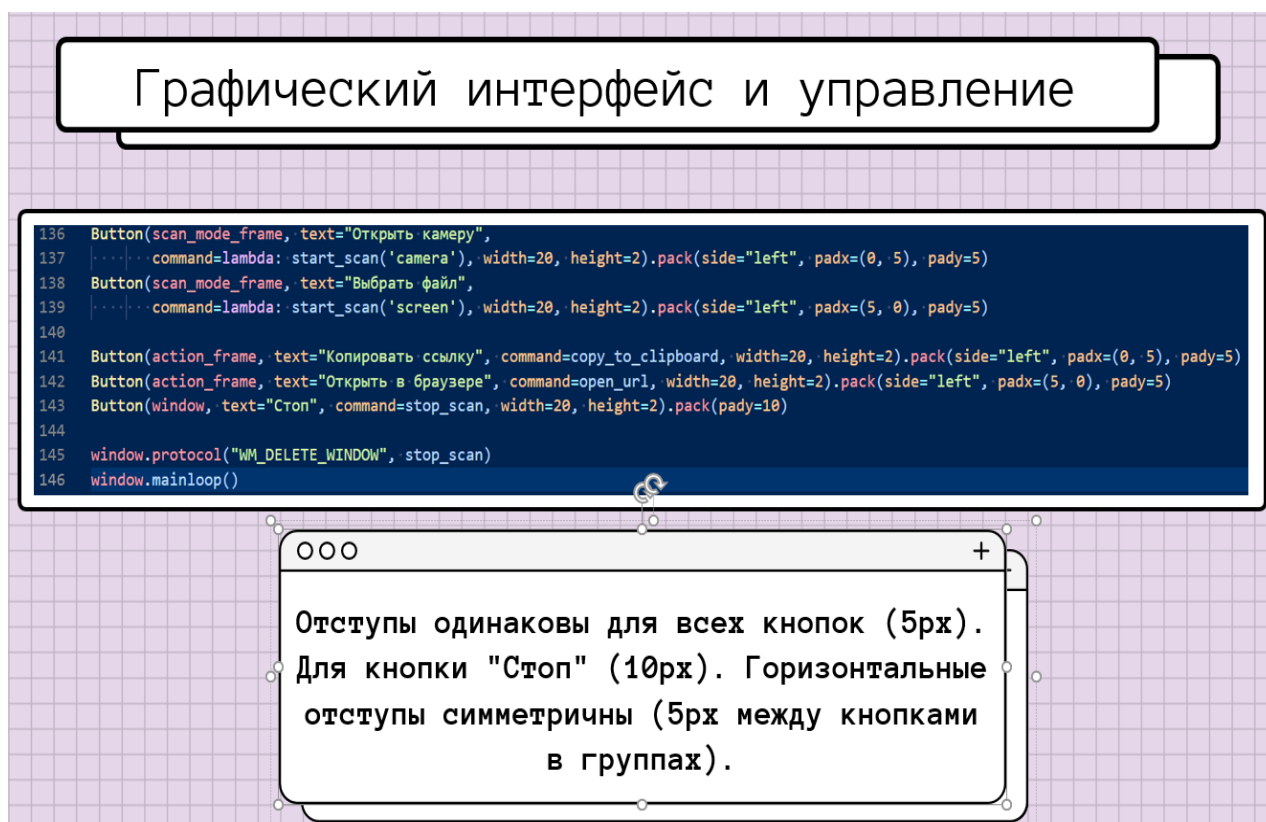


Рисунок 2 – Диалоговое окно, отображающее графический интерфейс и управление

Помимо основного функционала, интерфейс поддерживает дополнительные удобные функции. Например, реализована возможность копирования распознанной ссылки в буфер обмена одним нажатием кнопки, что избавляет пользователя от необходимости выделять и копировать текст вручную. Управление процессом сканирования интуитивно понятно и требует минимального обучения [2].

Перспективы и возможные улучшения.

Разработанное приложение представляет собой полноценный Minimum Viable Product (MVP), который может быть расширен и улучшен в нескольких направлениях. Анализ текущих тенденций и аналогичных проектов позволяет выделить следующие векторы развития:

- сохранение истории сканирований. Добавление локальной базы данных или простого текстового лога для хранения всех ранее распознанных кодов с привязкой ко времени;
- сканирование нескольких QR-кодов. Модернизация алгоритма для поиска и декодирования всех кодов, присутствующих на одном изображении или в кадре видеопотока;
- генератор QR-кодов. Добавление в приложение обратной функции — создания собственных QR-кодов из текста или ссылок, что превратит программу в универсальный инструмент;
- поддержка штрих-кодов. Расширение функционала PyZbar для чтения линейных штрихкодов (EAN-13, Code 128 и др.), что актуально для торговли и складского учета;
- запуск на Android. Адаптация кода для мобильных платформ с использованием фреймворков вроде Kivy или BeeWare, что позволит использовать приложение вне десктопной среды [5].

Разработанное в рамках данной работы приложение представляет собой завершенное программное решение, демонстрирующее практическую реализацию алгоритмов компьютерного зрения и обработки графической информации средствами Python. Разработанный программный конвейер демонстрирует эффективность взаимодействия данных компонентов, обеспечивая корректную идентификацию типа распознанных данных (URL-адреса или текстовая информация) и соответствующую реакцию системы. Отдельного внимания заслуживает реализация пользовательского интерфейса, выполненная с соблюдением принципов эргономики и унификации отступов, что подтверждает ориентацию на комфортное взаимодействие с программой [5]. Предложенные направления дальнейшего развития, включающие расширение функционала с целью обеспечения множественного сканирования, генерации кодов и адаптации под мобильные платформы, позволяют рассматривать разработанное приложение не как конечный продукт, а как основу для создания многофункциональной системы обработки графически закодированных данных.

Список литературы:

1. Лутц М. Программирование на Python / М. Лутц. – 4-е изд. – СПб.: Символ-Плюс, 2011. – Т. 1-2. – 2200 с. – ISBN 978-5-93286-159-6.
2. Титов А. Н. Введение в Tkinter. Разработка графических интерфейсов в Python: учебно-методическое пособие / А. Н. Титов // Казанский национальный исследовательский технологический университет. – Казань: Издательство КНИТУ, 2023. – 100 с. – ISBN 978-5-7882-3340-6.
3. Шакирьянов Э. Д. Компьютерное зрение на Python: первые шаги / Э. Д. Шакирьянов. – Москва: Лаборатория знаний, 2021. – 160 с. – ISBN 978-5-00101-318-1.
4. Хорст Я. Компьютерное зрение с OpenCV и Python: практическое руководство / Я. Хорст. – Москва: Литрес, 2025. – 180 с.
5. Vasilev I. Python Deep Learning / I. Vasilev. 3rd ed. Birmingham: Packt Publishing, 2023. 1 online resource. ISBN 978-1-83863-345-6.