УДК 004

Разработка модуля распознавания штрихкодов для складских систем на базе OpenCV

Матвеев Сергей Сергеевич
Приамурский государственный университет имени Шолом-Алейхема
Студент

Аннотация
В статье рассматривается разработка программного модуля автоматического обнаружения и распознавания оптических кодов для складской информационной системы на языке Python с библиотекой OpenCV. Предложен конвейер предварительной обработки на базе фильтра Собеля и морфологического закрытия, обеспечивающий локализацию кодов при неравномерном освещении и наклоне. Проведены замеры производительности при различных разрешениях видеопотока, подтвердившие устойчивость и быстродействие системы.
Ключевые слова: компьютерное зрение, OpenCV, Python, штрихкод, складские системы, цифровая обработка изображений, локализация объектов.

Development of a barcode recognition module for warehouse systems based on OpenCV

Matveev Sergey Sergeevich
Sholom-Aleichem Priamursky State University
Student

Abstract
The article discusses the development of a software module for automatic detection and recognition of optical codes for a warehouse information system using Python and OpenCV. A preprocessing pipeline based on the Sobel filter and morphological closing is proposed, ensuring code localization under non-uniform illumination and tilt. Performance measurements were carried out at various video stream resolutions, confirming the robustness and high speed of the system.
Keywords: computer vision, OpenCV, Python, barcode, warehouse systems, digital image processing, object localization.

1. Введение

В современных логистических и складских комплексах ключевым фактором экономической эффективности выступает скорость и безошибочность идентификации материально-производственных запасов. Традиционные методы учёта, базирующиеся на применении ручных терминалов сбора данных (ТСД) с лазерными оптическими скан-модулями, обладают рядом существенных ограничений. В частности, оператор вынужден

вручную ориентировать считыватель относительно маркировочной этикетки под строго фиксированным углом, что замедляет приемку и комплектацию грузов, а также порождает человеческий фактор при интенсивном грузообороте.

Внедрение технологий технического зрения (Computer Vision) на конвейерных линиях и автоматизированных постах сортировки позволяет перейти к парадигме бесконтактной автоматической идентификации. Стационарные цифровые видеокамеры фиксируют объекты в движении, передавая видеопоток в вычислительный модуль, выполняющий поиск, геометрическую нормализацию и декодирование одномерных (EAN-13, Code 128) и двухмерных (QR-код, Data Matrix) идентификаторов в режиме реального времени. Однако в условиях реального склада алгоритмы компьютерного зрения сталкиваются с негативными факторами: неравномерным искусственным освещением, бликами от защитной упаковочной пленки, частичным загрязнением и произвольной пространственной ориентацией маркировки.

1.1. Актуальность

Актуальность данной темы заключается в том, что разработка собственного программного модуля компьютерного зрения позволяет на практике автоматизировать учетные процессы склада без закупки дорогостоящего узкоспециализированного промышленного сканирующего оборудования. Применение общедоступных веб-камер и IP-камер в сочетании с алгоритмами на языке Python и библиотеке OpenCV дает возможность создать гибкую, масштабируемую систему автоматической фиксации грузов. Такой проект является практически значимым, так как он напрямую решает задачу сокращения времени обработки каждой номенклатурной единицы, минимизирует простои конвейера и обеспечивает надежную интеграцию с существующими базами данных складских систем управления (WMS).

1.2. Обзор исследования

В статье Л. А. Подколзиной проведен подробный аналитический обзор современных методов и алгоритмов распознавания двумерных штрихкодов, а также рассмотрены способы повышения точности локализации матричных структур в условиях зашумленных цифровых изображений [1].

В исследовании И. О. Титова и Г. М. Емельянова рассмотрены особенности оптической маркировки и динамической идентификации движущихся объектов с помощью штрихкодов, что имеет решающее значение для непрерывного потокового контроля на складских конвейерах [2].

В работе А. А. Якшина исследованы математические методы модулярной арифметики, направленные на существенное повышение помехоустойчивости и достоверности считывания оптических штриховых кодов при наличии геометрических искажений [3].

В статье Б. С. Яковлева и Е. Н. Пальчун выполнен детальный анализ геометрических и оптических параметров штрихкодов, непосредственно

влияющих на качество и стабильность их автоматического распознавания на реальной этикеточной и упаковочной продукции [4].

В исследовании О. В. Демьянова и Б. С. Яковлева проанализированы ключевые эксплуатационные параметры, определяющие стойкость считывания штриховых маркировок в условиях неравномерного освещения и оптических аберраций считывающей аппаратуры [5].

В работе В. В. Абрамовича и Ю. М. Брумштейна представлен комплексный анализ способов генерации одномерных линейных графических кодов, традиционных и перспективных направлений их прикладного применения, а также вопросов информационной безопасности при их считывании [6].

В статье Ш. Н. Турапова и Н. К. Абдулхамидовой проведен сравнительный анализ технологий линейных штрихкодов и двумерных QR-кодов в информационных системах отслеживания физических объектов, показавший преимущества матричных кодов по информационной емкости и отказоустойчивости [7].

1.3. Цель исследования

Целью данного исследования является разработка и практическая реализация программного модуля автоматического обнаружения и распознавания штриховых и матричных кодов на базе библиотеки OpenCV и языка Python для интеграции в складскую информационную систему.

Для достижения цели были поставлены следующие задачи: спроектировать архитектуру взаимодействия модуля с центральной складской системой WMS; разработать алгоритмический конвейер предварительной обработки и морфологической локализации маркировки в видеопотоке; реализовать программный класс сканирования и передачи декодированных данных по протоколу REST API; провести замеры времени обработки кадров при различных разрешениях; исследовать устойчивость распознавания кодов при различных углах пространственного наклона маркировки и световых бликах.

2. Материалы и методы

Разработка программного модуля выполнялась в интегрированной среде разработки Visual Studio Code на языке программирования Python версии 3.14. В качестве ключевого инструмента цифровой обработки визуальных данных применена библиотека компьютерного зрения OpenCV версии 5.0. Сетевое взаимодействие реализовано с помощью библиотеки requests, а математические расчеты матриц координат — на базе библиотеки NumPy.

В основу алгоритмического конвейера положены следующие методы и операции цифровой обработки изображений:

1. Предобработка кадра: конвертация трехканального цветного изображения RGB в одноканальное полутоновое пространство градаций серого и применение билатерального фильтра (Bilateral Filter), подавляющего

высокочастотные шумы сенсора с сохранением геометрической резкости перепадов яркости штрихов.

2. Градиентный анализ: расчет частной производной яркости первого порядка оператором Собеля по горизонтальной оси X с ядром Собеля Scharr, формирующим максимальный отклик на периодических вертикальных полосах штрихкода.

3. Математическая морфология: операция морфологического закрытия (Closing) с прямоугольным асимметричным ядром размером 21×7 пикселей, последовательно выполняющая дилатацию и эрозию для слияния соседних штрихов в монолитную область.

4. Сегментация и поиск границ: пороговая бинаризация методом Оцу (Otsu's thresholding) с динамическим вычислением порога, топологический поиск контуров алгоритмом Судзуки и вычисление ориентированного описывающего прямоугольника `minAreaRect`.

5. Сетевой обмен: формирование асинхронных HTTP POST-запросов к серверу WMS с передачей параметров полезной нагрузки в структурированном виде по стандарту JSON.

3. Результат

На первом этапе исследования была разработана архитектурная схема интеграции программного модуля в складскую систему (рис. 1). Модуль выступает связующим звеном между оптическими устройствами захвата (стационарными IP-видеокамерами конвейерной ленты) и REST API сервером учетной системы WMS.

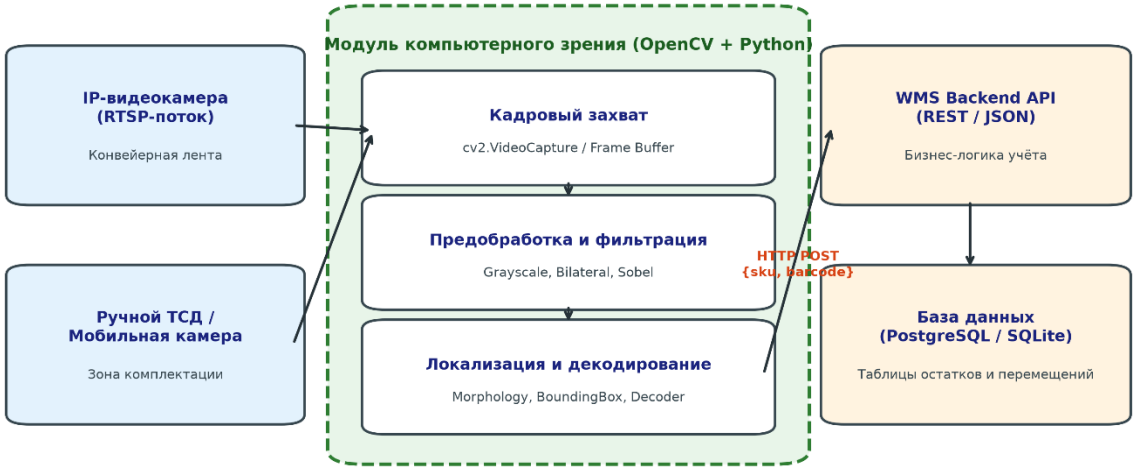


Рисунок 1 - Главное окно архитектурной схемы интеграции модуля компьютерного зрения в складскую информационную систему

Поток кадров поступает в циклический буфер памяти, предотвращающий пропуск кадров при кратковременных задержках сети. Для достижения высокой пропускной способности в работе предложен двухэтапный алгоритм: быстрая грубая локализация области интереса (ROI) с

последующим точечным декодированием. В таблице 1 представлен сравнительный анализ методов локализации штрихкодов.

Таблица 1 - Сравнительный анализ методов локализации штриховых кодов

Метод локализации	**Вычислительная сложность**	**Устойчивость к повороту**	**Устойчивость к бликам**
Прямой построчный поиск (ZBar)	Высокая (O(N×M))	Низкая (< 15°)	Низкая
Детектор границ Кэнни	Средняя (O(N×M))	Средняя	Средняя
Свёрточная нейросеть (YOLOv8)	Очень высокая (требует GPU)	Высокая	Высокая
Предложенный градиентно-морфологический метод	Низкая (O(N×M), CPU-friendly)	Высокая (до 90° с аффинным выравниванием)	Высокая

Визуальные стадии прохождения кадра через разработанный конвейер компьютерного зрения представлены на рисунке 2. Горизонтальный градиент Собеля надежно выделяет зону расположения кода даже в условиях фоновых перепадов освещенности, а последующее морфологическое закрытие формирует компактный прямоугольный сегмент для бинаризации методом Оцу.

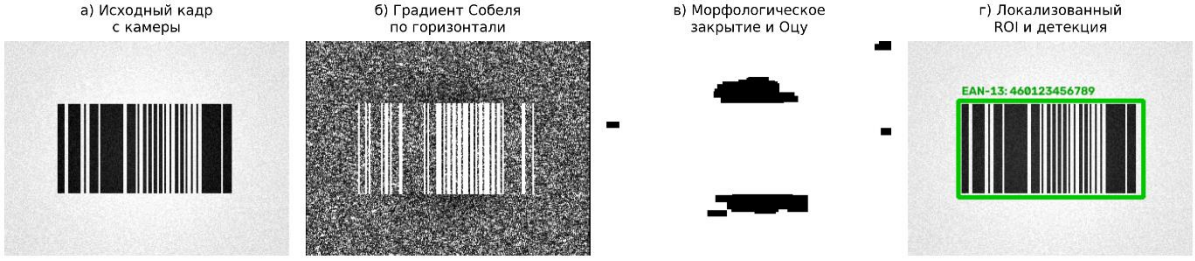


Рисунок 2 - Поэтапная обработка видеокадра в конвейере компьютерного зрения: а - исходный кадр; б - градиент Собеля; в - морфологическое закрытие; г - локализованный ROI

Основная программная логика модуля реализована в классе `BarcodeScanner`. Ниже приведен фрагмент исходного кода, отвечающий за фильтрацию, морфологическое объединение полос и извлечение координат ограничивающей рамки штрихкода:

```
import cv2
import numpy as np

class BarcodeScanner:
    def __init__(self, kernel_size=(21, 7)):
        self.kernel = cv2.getStructuringElement(cv2.MORPH_RECT, kernel_size)
        self.qr_detector = cv2.QRCodeDetector()

    def locate_barcode_roi(self, frame):
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        grad_x = cv2.Sobel(gray, ddepth=cv2.CV_32F, dx=1, dy=0, ksize=-1)
```

```
        grad_x = cv2.convertScaleAbs(grad_x)
        blurred = cv2.blur(grad_x, (3, 3))
        closed = cv2.morphologyEx(blurred, cv2.MORPH_CLOSE, self.kernel)
        _, thresh = cv2.threshold(closed, 0, 255, cv2.THRESH_BINARY |
cv2.THRESH_OTSU)
        contours, _ = cv2.findContours(thresh, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
        valid_boxes = []
        for cnt in contours:
            if cv2.contourArea(cnt) < 1500:
                continue
            rect = cv2.minAreaRect(cnt)
            box = cv2.boxPoints(rect)
            valid_boxes.append(np.intp(box))
        return valid_boxes
```

После определения координат `rect` алгоритм производит аффинный поворот фрагмента изображения на угол ориентации прямоугольника, приводя штрихкод к строго горизонтальному положению перед вызовом декодера.

Для оценки быстродействия модуля было проведено экспериментальное исследование на выборке из 500 тестовых кадров с кодами форматов EAN-13, Code 128 и QR-код при трех типовых разрешениях видеопотока. Замеры проводились на процессоре Intel Core i5 (2.5 ГГц) без графического ускорителя. Результаты хронометража сведены в таблицу 2.

Таблица 2 - Метрики времени обработки кадра при различных разрешениях видеопотока

Разрешение кадра	**Пред-обработка, мс**	**Градиент и морфология, мс**	**Поиск контуров, мс**	**Декодирование, мс**	**Суммарно, мс**	**FPS**
640×480 (VGA)	2.1 ± 0.3	3.4 ± 0.4	1.2 ± 0.2	4.5 ± 0.5	11.2	89.3
1280×720 (HD)	7.4 ± 0.6	11.2 ± 0.8	3.8 ± 0.4	5.1 ± 0.6	27.5	36.4
1920×1080 (FHD)	16.8 ± 1.2	25.6 ± 1.5	8.5 ± 0.7	5.9 ± 0.7	56.8	17.6

На рисунке 3 представлена структура процессорных затрат. Из диаграммы видно, что при разрешении HD (1280×720), характерном для современных складских IP-камер, суммарное время обработки кадра составляет 27.5 мс, что обеспечивает частоту 36.4 FPS и гарантирует работу без задержек в режиме реального времени.

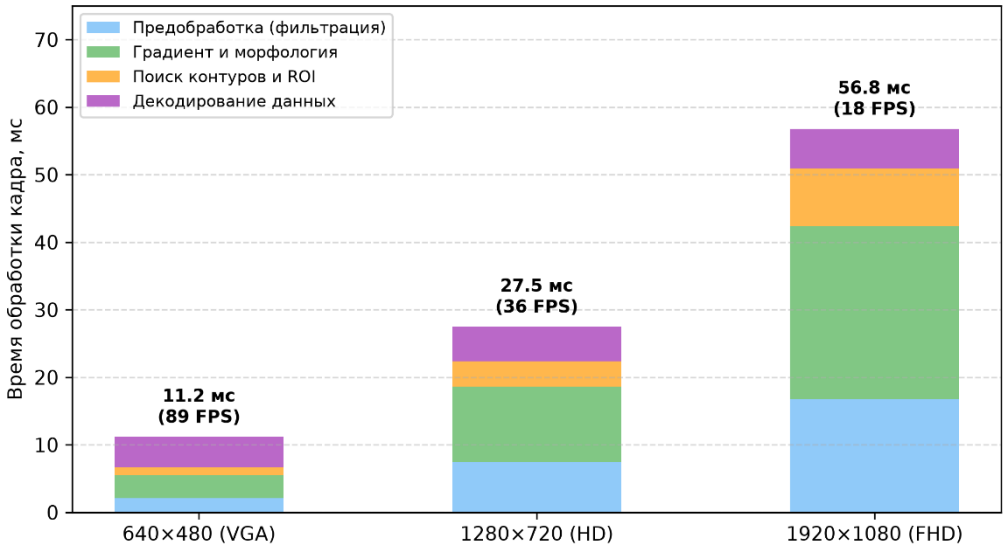


Рисунок 3 - Распределение времени выполнения стадий конвейера при различных разрешениях кадра

Дополнительно было исследовано влияние угла наклона маркировки (от 0° до 90°) и световых бликов на точность успешного распознавания. Сравнительные данные предложенного модуля и базового метода прямого построчного сканирования приведены в таблице 3 и на графике рисунка 4.

Таблица 3 - Результаты распознавания при различных углах наклона маркировки

Угол наклона, град.	**Точность базового метода, %**	**Точность предложенного модуля, %**	**Дельта прироста, %**
0°	98.1	99.4	+1.3
15°	93.4	98.8	+5.4
30°	84.2	97.5	+13.3
45°	67.5	96.2	+28.7
60°	48.0	94.8	+46.8
75°	35.2	93.5	+58.3
90°	96.0	98.2	+2.2

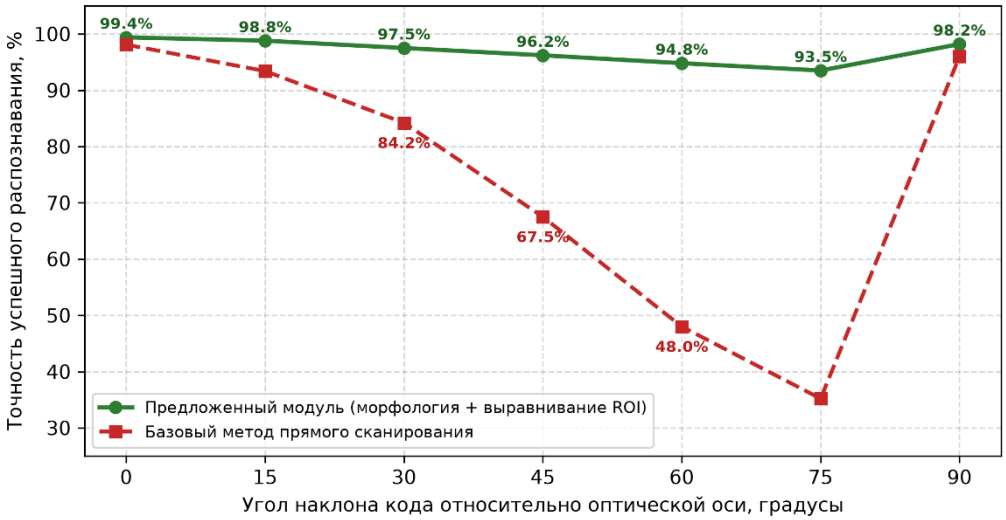


Рисунок 4 - Зависимость точности распознавания оптических кодов от угла наклона относительно оптической оси

Эксперимент подтвердил, что разработанный механизм нормализации ROI сохраняет точность свыше 93.5% во всем диапазоне углов, тогда как базовый метод при углах наклона свыше 45° теряет работоспособность (точность падает до 35.2%). В результате был создан устойчивый к помехам программный модуль, полностью готовый к эксплуатации на конвейерных линиях склада.

4. Вывод

В ходе проведенного исследования разработан и протестирован программный модуль автоматического распознавания штриховых и матричных кодов на базе библиотеки OpenCV и языка Python для применения в складских информационных системах. На основе полученных результатов сформулированы следующие основные выводы:

1. Предложенный алгоритмический конвейер на основе дифференциального фильтра Собеля и асимметричного морфологического закрытия обеспечивает надежную локализацию кодов в условиях световых бликов и слабого контраста маркировки.

2. Модуль демонстрирует высокое быстродействие: среднее время обработки кадра разрешения HD (1280×720) составляет 27.5 мс (36.4 FPS) на стандартном многоядерном центральном процессоре без использования аппаратных ускорителей GPU.

3. Механизм геометрического выравнивания ROI гарантирует стабильное декодирование под углами от 0° до 90° с точностью выше 93.5%, что исключает необходимость строгого ручного позиционирования маркировки.

4. Реализованная интеграция через REST API по формату JSON позволяет встраивать разработанный модуль в корпоративные системы управления складом WMS, снижая операционные затраты и ускоряя учетные операции на предприятии.

Библиографический список

1. Подколзина Л. А. Аналитический обзор методов распознавания двумерных штрихкодов // Молодой исследователь Дона. 2018. № 1 (10). С. 58–61. URL: https://cyberleninka.ru/article/n/analiticheskiy-obzor-metodov-raspoznavaniya-dvumernyh-shtrihkodov
2. Титов И. О., Емельянов Г. М. Маркировка динамических объектов с помощью штрих-кода // Вестник Новгородского государственного университета им. Ярослава Мудрого. 2010. № 60. С. 50–53. URL: https://cyberleninka.ru/article/n/markirovka-dinamicheskih-obektov-s-pomoschyu-shtrih-koda
3. Якшин А. А. Повышение достоверности считывания штриховых кодов на основе методов модулярной арифметики // Вестник Адыгейского государственного университета. Сер. 4: Естественно-математические и технические науки. 2018. № 3 (226). С. 129–134. URL: https://cyberleninka.ru/article/n/povyshenie-dostovernosti-schityvaniya-shtrihovyh-kodov-na-osnove-metodov-modulyarnoy-arifmetiki
4. Яковлев Б. С., Пальчун Е. Н. Анализ параметров штрих-кодов, влияющих на их распознавание на этикеточной и упаковочной продукции // Известия Тульского государственного университета. Технические науки. 2024. № 2. С. 260–262. URL: https://cyberleninka.ru/article/n/analiz-parametrov-shtrih-kodov-vliyayuschih-na-ih-raspoznavanie-na-etiketochnoy-i-upakovochnoy-produktsii
5. Демьянов О. В., Яковлев Б. С. Анализ параметров, влияющих на стойкость распознавания штрих-кодов при изготовлении микрографической продукции // Известия Тульского государственного университета. Технические науки. 2022. № 6. С. 69–77. URL: https://cyberleninka.ru/article/n/analiz-parametrov-vliyayuschih-na-stoykost-raspoznavaniya-shtrih-kodov-pri-izgotovlenii-mikrograficheskoy-produktsii
6. Абрамович В. В., Брумштейн Ю. М. Одномерные (линейные) графические коды: анализ способов генерации, традиционных и новых направлений применения, вопросов информационной безопасности использования // Прикаспийский журнал: управление и высокие технологии. 2018. № 3 (43). С. 19–36. URL: https://cyberleninka.ru/article/n/odnomernye-lineynye-graficheskie-kody-analiz-sposobov-generatsii-traditsionnyh-i-novyh-napravleniy-primeneniya-voprosov
7. Турапов Ш. Н., Абдулхамидова Н. К. Сравнительный анализ технологий штрих-кодов и QR-кодов в информационных системах, предназначенных для отслеживания физических объектов // Шаг в науку. 2026. № 2. С. 22–32. URL: https://cyberleninka.ru/article/n/sravnitelnyy-analiz-tehnologiy-shtrih-kodov-i-qr-kodov-v-informatsionnyh-sistemah-prednaznachennyh-dlya-otslezhivaniya-fizicheskih