

Санкт-Петербургский государственный университет

Кафедра системного программирования

Группа 22.Б15-мм

Распознавание штрихкодов по видео для автоматической инвентаризации

Мелашенко Ксения Викторовна

Отчёт по учебной практике
в форме «Решение»

Научный руководитель:
старший преподаватель кафедры СП, к.т.н., Ю.В. Литвинов

Консультант:
инженер-исследователь Mobile Robotics Lab, Skoltech, А.В. Корнилова

Санкт-Петербург
2023

Оглавление

1. Введение	3
2. Постановка задачи	5
3. Обзор	6
3.1. Подходы для локализации штрихкодов	6
3.2. Подходы для распознавания штрих-кодов	8
3.3. Подходы для улучшения распознавания штрихкодов . .	8
3.4. Метрики	9
3.5. Датасеты	10
3.6. Вывод	11
4. Реализация	12
4.1. BoundingBox	13
4.2. Localizer и Decoder	13
4.3. LocalizeIyyun и LocalizeYolo	13
4.4. DecodePyzbar и DecodeZxing	14
5. Сравнение результатов	15
5.1. Используемые наборы данных	15
5.2. Метрики	15
5.3. Алгоритмы локализации	15
5.4. Алгоритмы декодирования	16
5.5. Результаты	16
6. Заключение	18
Список литературы	19

1. Введение

Автоматизация инвентаризации значительно ускоряет процесс подсчета и проверки товаров на складе за счет использования специального программного обеспечения, оборудования и технологий, одной из которых является использование штрих-кодов. Штрих-кодирование представляет собой очень дешевый и надежный способ маркировки продуктов и, вероятно, останется отраслевым стандартом в ближайшие десятилетия. 1D-штрих-коды (их еще называют линейными) — наиболее распространенный тип идентификаторов, который используется на большинстве складов. Он вмещает в себя разное количество символов в зависимости от используемого формата. Например, штрих-код формата EAN-13 может содержать 13 цифр, а штрих-код формата Code 128 может вместить до нескольких сотен символов, чего обычно хватает для простой метки товара.

Штрих-коды могут быть считаны как лазерными сканерами, так и встроенными камерами. Использование лазерных сканеров считается уже довольно развитой технологией считывания штрих-кодов [9], поэтому в последнее время исследователи активно изучают потенциал цифровых камер в качестве средства идентификации штрих-кодов. С учетом широкого использования камер в повседневных ситуациях, возникает растущий интерес к распространению этой технологии на цифровые камеры. Здесь возникают следующие трудности: камера сделала размытое изображение, плохое освещение, повреждения или искажения штрих-кодов, штрих-код на изображении слишком маленький или его пропорции сильно искажены [9]. В итоге, в современных решениях входные данные должны быть хорошо оформленными изображениями, больше похожими на сканы, чем на фотографии [22] [13]. Отсюда процесс сканирования медленный, подверженный ошибкам и все еще требующий, чтобы пользователь прилагал свои усилия. Идеальной была бы система, которая позволяла бы обнаруживать и сканировать штрих-коды без необходимости взаимодействия с человеком.

Если окажется возможным устойчиво автоматически распознавать

объекты по видеопотоку, то эта технология может быть внедрена в различные процессы автоматизации, такие как использование на погрузочных машинах или квадрокоптерах.

2. Постановка задачи

Цель работы — повысить количество распознанных штрих-кодов за счет анализа видео последовательной съемки для улучшения автоматической инвентаризации.

Для достижения данной цели были поставлены следующие задачи.

- Задачи на осенний семестр:

1. Выполнить обзор существующих методов локализации штрих-кодов и их распознавания, а также метрик и датасетов, используемых для их оценки.
2. Реализовать инструмент для унификации существующих алгоритмов, который комплексно решает задачи обнаружения штрих-кодов и декодирования штрих-кодов.
3. Сравнить алгоритмы локализации и декодирования штрих-кодов.

- Задачи на весенний семестр:

1. Выполнить обзор существующих методов улучшения распознавания штрихкодов.
2. Внедрить методы улучшения распознавания штрихкодов в инструмент.
3. Реализовать метод анализа видео последовательной съемки, внедрив решения, лидирующие в сравнении.
4. Провести экспериментальное сравнение полученных результатов.

3. Обзор

В данной главе рассматриваются существующие алгоритмы локализации и декодирования штрихкодов, а также метрик и датасетов, используемых для оценки этих алгоритмов.

3.1. Подходы для локализации штрихкодов

Задача локализации штрих-кода состоит в обнаружении области, на которой располагается штрих-код на данном изображении.

В большинстве случаев, выходные данные алгоритма локализации штрих-кодов состоят из набора ограничивающих рамок (bounding boxes), обрамляющих эти штрих-коды, как, например, в алгоритмах из статей [12, 23].

Для исследования существующих алгоритмов для локализации штрих-кодов, использующих классические методы их области компьютерного зрения или обученные нейросети, были проанализированы публикации за последние несколько лет.

3.1.1. Алгоритмы без обучения

Большинство существующих методов обнаружения штрих-кодов основаны на классических методах компьютерного зрения, таких как морфологические операции, чтобы найти структуру на изображении, подобную штриху. Морфологическая обработка включает в себя методы обработки изображений, область применения которых связана с формой объектов на изображении. Обычно эти методы применяются, чтобы сократить количество «недостатков» на бинарном изображении (обработка и устранение шума), также сюда входят выделение краев, описание формы, сегментация и др. Примерами таких морфологических преобразований являются вычисление градиента с использованием оператора Собеля (Sobel kernel), бинаризация, использование структурирующего элемента и т.д. Подробнее об этом можно узнать из работы [8]. Авторы статьи также предлагают альтернативный подход с использованием

bottom-hat filtering и distance map в своей следующей работе [7]. В расширенной версии добавлена карта евклидовых расстояний. Используя эту информацию, объекты, находящиеся далеко от других объектов, можно легко удалить, и останутся только близлежащие объекты (группа сегментов полос).

Методы с использованием морфологических операций хорошо подходят в случаях, когда важна скорость вычислений, а не точность [15], и всегда зависят от правильного выбора структурирующего элемента, на что ссылаются в статье [2].

Maximally stable extremal regions (MSER) используется в статье [2] для обнаружения темных полос штрих-кодов с последующим преобразованием Хафа, чтобы найти перпендикулярную линию полосы штрих-кода, проходящую через центр этой полосы. В следующей своей статье [3] авторы использовали детектор параллельных сегментов (PSD), который основан на детекторе линейных сегментов (LSD). Однако эти методы требуют высокого качества изображения, а MSER очень чувствителен к размытию (об этом утверждается в работе [1]), что не позволяет обнаружить штрих-коды в большинстве кадров, полученных в быстро движущемся видео.

В статье [23] локализация штрих-кода может происходить в реальном времени, а также метод устойчив к перевернутому штрихкоду и масштабированию. Здесь авторы метода сначала использовали гистограмму направленных градиентов (orientation histogram) входных изображений для удаления шума, а затем с помощью вычисления энтропии выбирали область, которая, скорее всего, представляет собой штрих-код.

3.1.2. Алгоритмы с обучением

В области локализации штрих-кодов, статья [24] была одной из первых работ, использовавших глубокое обучение вместе с преобразованием Хафа. Благодаря возможностям нейронных сетей находить объекты на изображении, позднее были представлены несколько работ, использующих различные модели, такие как R-CNN, ANNs [18], RPN [12], Y-

Net, YOLO [14], а также YOLOv4, YOLOv5 и YOLOv7.

Например, в работах [17, 5] для обучения модели обнаружения штрих-кода используется Faster R-CNN, а в исследовании [13] авторы используют алгоритм обнаружения объектов глубокого обучения YOLO. В последнем случае алгоритм дает хорошие результаты только на качественном не размытом изображении.

3.2. Подходы для распознавания штрих-кодов

Задача декодирования штрих-кода состоит в извлечении информации из штрих-кода, который расположен на данном изображении, используя результаты алгоритма локализации.

3.2.1. Pyzbar и Zxing

Pyzbar и Zxing — открытые библиотеки по декодированию штрих-кодов. Они обладают хорошей поддержкой различных форматов штрих-кодов, таких как QR-коды, штрих-коды EAN и др. и являются самыми распространенными библиотеками в этой области.

Pyzbar¹ — это библиотека Python, которая считывает одномерные штрих-коды и QR-коды с использованием библиотеки Zbar — пакета программного обеспечения с открытым исходным кодом для считывания штрих-кодов из видеопотока и изображений.

ZXing² (Zebra Crossing) — это тоже библиотека обработки изображений 1D/2D штрих-кодов с открытым исходным кодом, реализованная на Java и имеющая порты на другие языки.

3.3. Подходы для улучшения распознавания штрих-кодов

Одним из подходов является операция поворота. Например, в статье [6] вычисляют угол поворота наклонного штрих-кода и поворачивают

¹<https://github.com/NaturalHistoryMuseum/pyzbar> (дата обращения: 2023-12-14)

²<https://github.com/zxing/zxing> (дата обращения: 2023-12-14)

наклонную область штрих-кода в горизонтальное состояние с помощью аффинных преобразований, чтобы штрихкод можно было легче распознать.

Еще одним подходом является устранение размытия (deblurring). В статье [20] делают устранение размытия изображений штрих-кода для повышения скорости декодирования с помощью DeblurGAN-v2 [4], а в статье [16] применяется постобработка с двусторонним фильтром, называемым фильтром Adaptive Manifold (AM).

3.4. Метрики

Метрикой можно численно оценить производительность различных методов локализации и декодирования штрих-кодов и понять, какой метод более эффективен в определенных сценариях использования.

- **Jaccard index**

$$J(R, G) = \frac{|R \cap G|}{|R \cup G|}, \text{ где}$$

R — результат алгоритма, G — ожидаемый результат алгоритма

- **Intersection over union (IoU)**

$$IoU = \frac{TP}{TP + FP + FN}, \text{ где}$$

TP — True Positive, FP — False Positive, FN — False Negative

$$R = \frac{\text{correct}}{\text{possible}} = \frac{TP}{TP + FN}$$

- **Precision**

$$P = \frac{\text{correct}}{\text{actual}} = \frac{TP}{TP + FP}$$

- **Recall**

$$R = \frac{\text{correct}}{\text{possible}} = \frac{TP}{TP + FN}$$

где `correct` — количество штрих-кодов, успешно распознанных системой, `possible` — это общее количество штрих-кодов, которые мы ожидаем иметь возможность декодировать, а `actual` — общее количество штрих-кодов, распознаваемых системой.

`Precision` можно интерпретировать как долю объектов, которые распознала система и распознала их правильно, а `recall` показывает, какую долю правильно найденных объектов из всех объектов нашел алгоритм.

3.5. Датасеты

- **WWU Muenster Barcode Database** [19] — датасет из 1055 изображений штрих-кодов формата EAN и UPC-A. Только пару снимков из датасета имеют более одного экземпляра штрих-кода на изображении.
- **ArTe-Lab 1D Medium Barcode Dataset** [11] — включает в себя два датасета одномерных штрихкодов формата EAN13. Первый датасет содержит изображения штрих-кодов, снятые с устройств с автофокусом, а второй датасет содержит снимки, сделанные устройствами без автофокуса.
- **ParcelBar** — датасет из 844 изображений штрих-кодов формата Code128
- **Synthetic Barcode Datasets (SBD)** [12] — 100 000 синтетических наборов данных штрих-кодов низкого качества и 100 000 наборов данных синтетических штрих-кодов высокого качества вместе с `ground truth masks` ограничивающих рамок, что позволяет оценивать качество алгоритма локализации. Датасеты включают в себя множество различных форматов штрих-кодов, таких, как Code 39, Code 128, UPC, EAN, QR и много других.

3.6. Вывод

Среди рассмотренных статей, посвященных локализации и декодированию штрих-кодов, было найдено несколько методов с открытым исходным кодом. Некоторые из них удалось запустить локально, а другие оказались сильно устаревшими.

Можно заметить, что современные подходы сначала используют классические методы обработки изображений, а далее прибегают к использованию нейронных сетей для нахождения объектов, таких как штрих-коды. Также, они используют перечисленные выше подходы для улучшения распознавания. Согласно опросу из [21], большинство существующих работ решают каждую задачу локализации и декодирования штрих-кодов независимо. Отсюда следует потребность в подходах, которые комплексно решают обе эти задачи с возможностью внедрения технологии для улучшения распознавания штрихкодов.

4. Реализация

Большинство алгоритмов локализации и декодирования в оригинальной реализации были написаны на разных языках программирования, таких как Java, C++ и Python. Для удобства реализации инструмента был выбран язык Python, так как он предоставляет возможность работать с библиотекой NumPy³ — удобной и эффективной библиотекой для работы с данными. Также, была использована библиотека OpenCV⁴ для работы с изображениями. В реализацию попали методы, код кото-

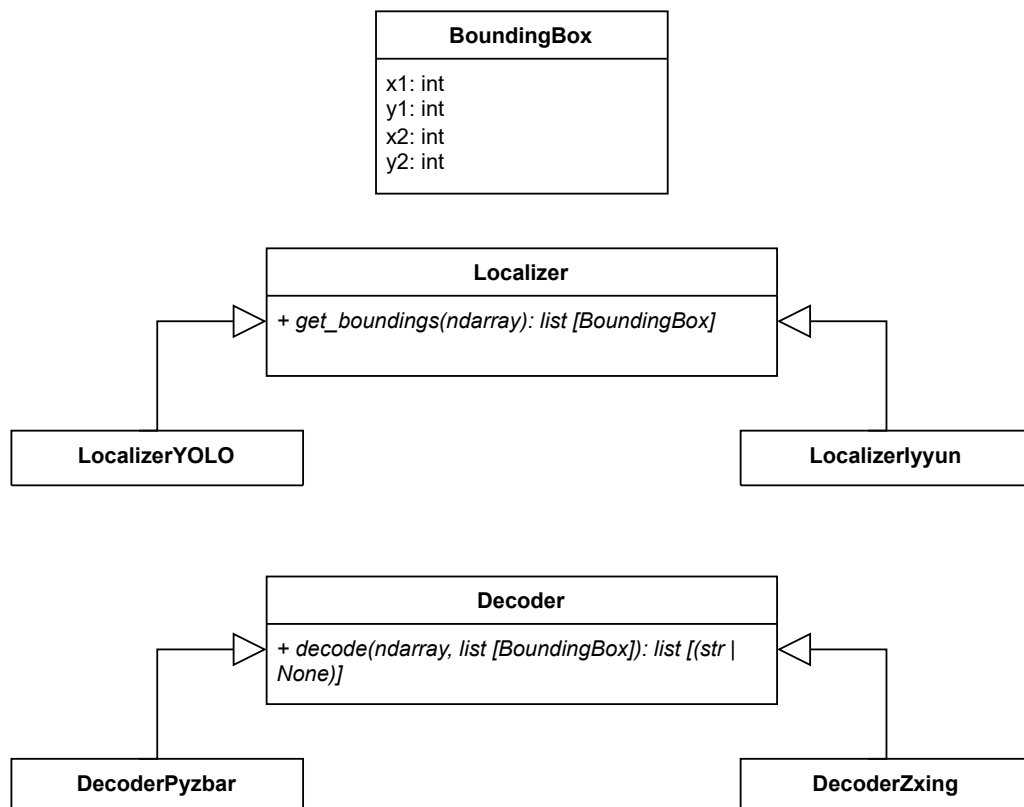


Рис. 1: UML-диаграмма реализованного инструмента

рых был в открытом доступе, а именно: метод от автора Iyyun [23], метод с использованием YOLOv7⁵, Pyzbar и Zxing.

Как видно из UML-диаграммы на рис. 1, основными классами инструмента являются: **LocalizerIyyun**, **LocalizerYolo**, **DecoderPyzbar**, **DecoderZxing**, **Localizer**, **Decoder** и **BoundingBox**. Далее описаны дета-

³<https://numpy.org/> (дата обращения: 2023-12-14)

⁴<https://opencv.org> (дата обращения: 2023-12-14)

⁵<https://github.com/kirill-ivanov-a/onnx-yolov7>

ли того, за что отвечают эти классы.

4.1. BoundingBox

Как было сказано ранее, выходные данные алгоритма локализации штрих-кодов — это набор ограничивающих рамок (bounding boxes) каждого найденного штрих-кода на изображении. `BoundingBox` описывает координаты каждой такой рамки, где `x1`, `y1` — координаты нижнего левого угла, а `x2`, `y2` — координаты верхнего правого угла.

4.2. Localizer и Decoder

`Localizer` и `Decoder` представляют собой классы, которые описывают абстракцию методов классов `LocalizerIyyun`, `LocalizerYolo`, которые наследуются от `Localizer`, и абстракцию методов классов `DecoderPyzbar` и `DecoderZxing`, которые наследуются от класса `Decoder`.

`Localizer` содержит в себе абстракцию метода `get_boundings`. Метод принимает изображение, преобразованное в массив `numpy`, и возвращает список из координат обрамляющих рамок `list[BoundingBox]`, полученный с помощью одного из алгоритмов локализации.

`Decoder` содержит в себе абстракцию метода `decode`. Он принимает список из координат обрамляющих рамок `list[BoundingBox]`, по которым входное изображение обрезается и далее передается алгоритму для декодирования штрих-кода. Метод возвращает список строк декодированного штрих-кода или значение `None`, если штрих-код не удалось декодировать.

4.3. LocalizeIyyun и LocalizeYolo

Для воспроизводимости алгоритмов с их зависимостями на локальной машине, они были обернуты в `Docker` контейнера при помощи инструкций в `Dockerfile`. `LocalizerIyyun` — класс, который содержит в себе реализацию метода, предложенного в статье [23]. `LocalizerYolo`

— класс, содержащий в себе обертку модели YOLOv7, сделанную с целью упрощения работы с YOLO.

4.4. DecodePyzbar и DecodeZxing

Методы здесь также были обернуты в Docker контейнера при помощи инструкций в Dockerfile. **DecodePyzbar** - класс, воспроизводящий алгоритм декодирования из библиотеки Pyzbar, а **DecodeZxing** — из библиотеки Zxing.

5. Сравнение результатов

В данной главе приводятся результаты запуска и визуального сравнения алгоритмов локализации и декодирования штрих-кодов.

5.1. Используемые наборы данных

В качестве небольшого экспериментального датасета было выбрано несколько снимков из открытых датасетов WWU Muenster Barcode Database и ArTe-Lab 1D Medium Barcode Dataset, так как именно они чаще всего фигурировали в сравнениях. Также, было выбрано несколько снимков, сделанных исследователями из Mobile Robotics Lab, Skoltech [10]. Выбранные снимки представляют собой как качественно сделанные изображения, так и «экзотические случаи» — размытые, перевернутые и деформированные.

5.2. Метрики

Для простоты сравнения результатов существующих работ, мы используем ту же метрику, что и в статье [2], а именно метрику Jaccard, которая в случае локализации даст оценку, насколько правильно пересекаются области, найденные алгоритмом, с действительными областями, на которых находится штрихкод. В случае декодирования, метрика даст оценку количества правильно распознанных штрихкодов из общего количества штрихкодов на изображении.

5.3. Алгоритмы локализации

В таблице 1 представлены численные результаты алгоритмов локализации, запущенных на тестовом датасете. Видно, что на данных из Mobile Robotics Lab, Skoltech метод с использованием модели YOLOv7 дал очень хорошие результаты, значительно обогнав метод Iyulun. Оба метода отлично справились с локализацией штрих-кодов на качественных изображениях, сделанных в хорошо освещенном помещении, что

можно увидеть на рисунке 2. Из рисунка также можно увидеть, что иногда алгоритмы захватывали область, на которой не присутствует штрих-кода. На данных WWU Muenster и ArTe-Lab метод с YOLOv7 распознаёт очень плохо. Так как модель YOLOv7 была предобучена на выборке со стикерами, она не всегда распознает именно сами штрих-коды, из которых и состоят оба перечисленных датасета выше, и в основном обнаруживает только стикеры со штрихкодами. В плохо освещенном помещении YOLOv7 также сработал хорошо, но Iyuuu с этой задачей не справился. К тому же, Iyuuu дает множество неправильных результатов на датасетах WWU Muenster и ArTe-Lab, где в основном изображения довольно приемлемого качества, а штрихкоды ровные и четкие.

5.4. Алгоритмы декодирования

Метод `decode` из библиотеки `Zxing` отлично декодирует качественно сделанные изображения, однако декодировать штрих-коды, которые перевернуты, размыты или с какими-либо дефектами, он не смог.

`Pyzbar` тоже хорошо работает с качественно сделанными изображениями. Кроме того, он показал неплохие результаты на снимках, где штрих-коды были с небольшими дефектами или были перевернуты. Размытые штрих-коды не распознал.

5.5. Результаты

Путем визуального сравнения можно сделать вывод, что существующие методы с использованием нейронных сетей несколько превосходят методы, использующие методы из классического CV, и будут хорошо подходить в условиях инвентаризации складов. Также, в сравнении алгоритмов декодирования, библиотека `Pyzbar` превзошла библиотеку `Zxing` в общем количестве успешно декодированных штрих-кодов, так что использование `Pyzbar` будет давать больше хороших результатов.



(a) Iyyun method



(b) YOLOv7



(c) Iyyun method



(d) YOLOv7

Рис. 2: Сравнение результатов алгоритмов локализации

Таблица 1: Jaccard index алгоритмов локализации штрихкодов

Набор данных	WWU Muenster	ArTe-Lab	Sk Campus
YOLOv7	9.1	25.2	95.93
Iyyun	24.1	16.7	55.0

Таблица 2: Jaccard index алгоритмов декодирования штрихкодов

Набор данных	WWU Muenster	ArTe-Lab	Sk Campus
Pyzbar	70.0	20.0	44.1
Zxing	60.0	40.0	17.6

6. Заключение

В рамках работы были выполнены следующие задачи:

1. Выполнен обзор существующих методов локализации штрих-кодов и их распознавания, а также метрик и датасетов, используемых для их оценки.
2. Реализован инструмент для унификации существующих алгоритмов, который комплексно решает задачи обнаружения штрих-кодов и декодирования штрих-кодов. Ссылка на GitHub репозиторий — <https://github.com/ksenmel/Barcode-Detection>.
3. Было проведено сравнение алгоритмов локализации и декодирования штрих-кодов. Было установлено, что алгоритмы локализации с использованием нейронных сетей дают результаты лучше, чем алгоритмы с классическими подходами из области компьютерного зрения.

В следующем семестре планируется:

1. Реализовать метод анализа видео последовательной съемки, внедрив решения, лидирующие в сравнении.
2. Провести экспериментальное сравнение полученных результатов.

Список литературы

- [1] Brylka Robert, Schwanecke Ulrich, Bierwirth Benjamin. [Camera Based Barcode Localization and Decoding in Real-World Applications](#) // 2020 International Conference on Omni-layer Intelligent Systems (COINS). — 2020. — P. 1–8.
- [2] Creusot Clement, Munawar Asim. [Real-Time Barcode Detection in the Wild](#) // 2015 IEEE Winter Conference on Applications of Computer Vision. — 2015. — P. 239–245.
- [3] Creusot Clement, Munawar Asim. [Low-computation egocentric barcode detector for the blind](#) // 2016 IEEE International Conference on Image Processing (ICIP). — 2016. — P. 2856–2860.
- [4] DeblurGAN-v2: Deblurring (Orders-of-Magnitude) Faster and Better / Orest Kupyn, T. Martyniuk, Junru Wu, Zhangyang Wang // 2019 IEEE/CVF International Conference on Computer Vision (ICCV). — 2019. — P. 8877–8886. — URL: <https://api.semanticscholar.org/CorpusID:199543931>.
- [5] Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks / Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun // arXiv preprint arXiv:1506.01497. — 2015.
- [6] Jinwang Yi Yuanbiao Xiao. Efficient Localization of Multitype Barcodes in High-Resolution Images // Mathematical Problems in Engineering. — Vol. 2022. — 2022.
- [7] Katona Melinda, Nyúl László G. Efficient 1D and 2D Barcode Detection Using Mathematical Morphology // International Symposium on Mathematical Morphology and Its Application to Signal and Image Processing. — 2013. — URL: <https://api.semanticscholar.org/CorpusID:15768838>.
- [8] Katona Melinda, Nyúl László G. [A Novel Method for Accurate and Efficient Barcode Detection with Morphological Operations](#) // 2012

Eighth International Conference on Signal Image Technology and Internet Based Systems. — 2012. — P. 307–314.

- [9] Katuk Norliza, Mahamud K.-R.K., Zakaria Nur Haryani. A review of the current trends and future directions of camera barcode reading // Journal of Theoretical and Applied Information Technology. — 2018. — 04. — Vol. 97. — P. 2268–2288.
- [10] Mobile Robotics Lab, Skoltech Warehouse dataset. — <https://drive.google.com/file/d/1VMUETtXrG9jhJ8NVBJXD1QV1xXRvSJhD/view?pli=1>, https://drive.google.com/file/d/10VMI84azDB5j8_lnCesZbUwaf8IPL57K/view.
- [11] Neural Image Restoration for Decoding 1-D Barcodes using Common Camera Phones / Alessandro Zamberletti, Ignazio Gallo, Moreno Carullo, Elisabetta Binaghi // International Conference on Computer Vision Theory and Applications. — 2010. — URL: <https://api.semanticscholar.org/CorpusID:14224247>.
- [12] Quenum Jerome, Wang Kehan, Zakhor Avidel. [Fast, Accurate Barcode Detection in Ultra High-Resolution Images](#) // 2021 IEEE International Conference on Image Processing (ICIP). — 2021. — P. 1019–1023.
- [13] Real-time barcode detection and classification using deep learning / Daniel Kold Hansen, Kamal Nasrollahi, Christoffer Bøgelund Rasmussen, Thomas B Moeslund // International Joint Conference on Computational Intelligence / SCITEPRESS Digital Library. — 2017. — P. 321–327.
- [14] Redmon Joseph, Farhadi Ali. YOLO9000: Better, Faster, Stronger // 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). — 2016. — P. 6517–6525. — URL: <https://api.semanticscholar.org/CorpusID:786357>.

- [15] Sörös Gábor, Floerkemeier Christian. Blur-resistant joint 1D and 2D barcode localization for smartphones // Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia. — 2013. — URL: <https://api.semanticscholar.org/CorpusID:31145666>.
- [16] Using Deep ConvNet for Robust 1D Barcode Detection / Jianjun Li, Qiang Zhao, Xu Tan et al. — 2017. — URL: <https://api.semanticscholar.org/CorpusID:67006418>.
- [17] Using Deep ConvNet for Robust 1D Barcode Detection / Jianjun Li, Qiang Zhao, Xu Tan et al. // Advances in Intelligent Systems and Interactive Applications. — Cham : Springer International Publishing, 2018. — P. 261–267.
- [18] Ventsov N.N., Podkolzina L.A. [Localization of Barcodes Using Artificial Neural Network](#) // 2018 IEEE East-West Design Test Symposium (EWDTS). — 2018. — P. 1–6.
- [19] Wachenfeld Steffen, Terlunen Sebastian, Jiang Xiaoyi. [Robust 1-D Barcode Recognition on Camera Phones and Mobile Product Information Display](#) // Mobile Multimedia Processing: Fundamentals, Methods, and Applications / Ed. by Xiaoyi Jiang, Matthew Y. Ma, Chang Wen Chen. — Berlin, Heidelberg : Springer Berlin Heidelberg, 2010. — P. 53–69. — ISBN: 978-3-642-12349-8. — URL: https://doi.org/10.1007/978-3-642-12349-8_4.
- [20] Wang Chaoxin, Guevara Nicolais, Caragea Doina. [Using Deep Learning to Improve Detection and Decoding Of Barcodes](#) // 2022 IEEE International Conference on Image Processing (ICIP). — 2022. — P. 1576–1580.
- [21] Wudhikarn Ratapol, Charoenkwan Phasit, Malang Kanokwan. Deep Learning in Barcode Recognition: A Systematic Literature Review // [IEEE Access](#). — 2022. — Vol. 10. — P. 8049–8072.
- [22] Yi Jinwang, Xiao Yuanbiao. Efficient Localization of Multitype Barcodes in High-Resolution Images // Mathematical Problems in

Engineering. — 2022. — URL: <https://api.semanticscholar.org/CorpusID:247874665>.

- [23] Yun Inyong, Kim Joongkyu. Vision-based 1D barcode localization method for scale and rotation invariant // Region 10 Conference, TENCON 2017-2017 IEEE / IEEE. — 2017. — P. 2204–2208.
- [24] Zamberletti Alessandro, Gallo Ignazio, Albertini Simone. Robust Angle Invariant 1D Barcode Detection // In proceedings of the 2nd Asian Conference on Pattern Recognition (ACPR). — 2013.