МЕТОДЫ ОБРАБОТКИ ИЗОБРАЖЕНИЙ И ИСПОЛЬЗОВАНИЕ КОМПЬЮТЕРНОГО ЗРЕНИЯ В OCR

IMAGE PROCESSING TECHNIQUES AND COMPUTER VISION IN OCR

УДК: 004.93

Доронин Юрий Дмитриевич, магистрант, Московский Государственный Технический Университет имени Н.Э. Баумана, г. Москва

Doronin Yuri Dmitrievich, master student, Moscow State Technical University named after N.E. Bauman, Moscow

Аннотация

С появлением современных методов машинного обучения часть интерпретации окружающего мира была возложена на компьютер. Оптическое распознавание символов - простейший из примеров, где это реализовано. Оптическое распознавание символов или OCR - хорошо изученная область в сфере машинного обучения. Программный продукт OCR принимает изображение текста, проводит определенные манипуляции над ним для повышения эффективности алгоритмов распознавания, и преобразует его в цифровой формат.

Annotation

With the advent of modern machine learning methods, part of the interpretation of the surrounding world was assigned to the computer. Optical character recognition is the simplest example where this is implemented. Optical character recognition, or OCR, is a well-studied field in machine learning. The OCR software product takes an

image of a text, performs certain manipulations on it to improve the efficiency of recognition algorithms, and converts it to a digital format.

Ключевые слова: оптическое распознавание символов, обработка изображения, машинное обучение, рукописный текст

Key words: optical character recognition, image processing, machine learning, handwritten text

Введение

Одной из проблем в современном мире, является скорость получения информации. Одна из причин этого - недоступность старых данных в цифровом формате. Для ввода данных в электронную таблицу или цифровой документ требуется ручной труд. Если данные имеют определенную структуру, к примеру, заданы в табличном формате, на это может уходить десятки минут, что в рамках тысяч документов превращается в месяцы работы. Изображение документа занимает гораздо больше места для хранения, чем тот же документ в цифровом формате.

Стандартное программное обеспечение OCR использует модель нейронной сети для распознавания символов на основе предыдущего обучения. Модель сначала обучается с использованием тысяч выборок каждого возможного символа, чтобы она фактически «узнала», как выглядит этот конкретный символ. Только когда погрешность модели достаточно низка для точного распознавания каждого символа, полученную модель можно использовать на реальных задачах.

1 Цели исследования

Основная цель состоит в том, чтобы проанализировать существующие методы распознавания текста из отсканированных изображений. Другая цель – распознавание табличных данных, определение табличной структуры входного изображения, а затем сохранение его в формате CSV или в виде электронной таблицы. Для достижения наилучших результатов необходимо учитывать множество факторов. Эти факторы не касаются конечного пользователя системы, но исследуемые методы должны их решать. Некоторые из факторов:

1 Качество входного изображения: Входное изображение может быть низкого разрешения, содержать шум, иметь неподходящие и неравномерные условия освещения во время получения изображения. Человеческий глаз может понять и объяснить более низкое качество изображения, потому что миллионы лет эволюции помогают легко распознавать закономерности, но для компьютера изображение более низкого качества может оказаться неузнаваемым. Проблема может быть решена путем предварительной обработки изображения для компенсации низкого качества изображения.

2 Язык текста: Система OCR может поддерживать нескольких языков. Количество поддерживаемых языков зависит от набора обучающих данных, который будет использоваться в процессе машинного обучения. Тренировка модели может быть выполнена на нескольких языках только при наличии большого количества данных для обучения.

3 Поддержка рукописных данных: Система может поддерживать распознавание рукописных данных, только в том случае, если модель, обученная на рукописных данных, имеет результаты не хуже, чем на печатных. Качество результата распознавания зависит от типов почерка, используемых в обучающих наборах.

4 Наличие границ в табличных данных: если входное изображение содержит табличные данные, наличие границ может помочь идентифицировать всю таблицу, а также каждую ячейку таблицы. Если границы отсутствуют, может потребоваться помощь человека, чтобы найти таблицу на изображении. Ячейки таблицы можно идентифицировать по пробелу между текстовыми блоками.

5 Скорость: Система должна осуществлять разбор символов за небольшой промежуток времени, особенно в случае табличных данных, где весь процесс распознавания символов должен повторяться для каждой ячейки. Если система работает медленнее, чем человек, который расшифровывает текст изображения в цифровой форме, тогда использование системы теряет смысл.

2 Получение изображения

Система OCR требует, чтобы на входе было изображение, содержащее текст, который необходимо извлечь. Разрешение и общее качество изображения играют важную роль в общем процессе. Существуют следующие способы получения изображения:

1 Отсканированный документ: Документ можно отсканировать и сохранить как изображение.

2 Фотографии: Распознаваемый документ или текст можно сфотографировать с помощью мобильного телефона или фотоаппарата.

3 Цифровые изображения и искусство: Изображения, созданные в цифровом виде и содержащие текст, могут быть распознаны системой OCR.

4 Скриншоты: иногда приложения на мобильных телефонах и компьютерах одинаково содержат текст, который нельзя выделить и скопировать. Этот текст можно распознать, если сделать снимок экрана при отображении текста.

Технические характеристики устройства получения изображения должны быть достаточно высокими, чтобы все детали изображения были сохранены. Следует позаботиться о таких вещах, как фокусировка на тексте, уменьшение шума и правильное освещение.

Изображения можно интерпретировать, как двумерные массивы с множеством слоев (каналов), представляющих цвет, т. е. существует столько двумерных массивов, сколько существует поддерживаемых цветов. Каждый элемент двумерного массива каждого канала сохраняет интенсивность этого пикселя в канале. Комбинация соответствующих пикселей из каждого канала дает уникальный общий цвет пикселя (рисунок 1). Размеры 2D-массива — это разрешение изображения.

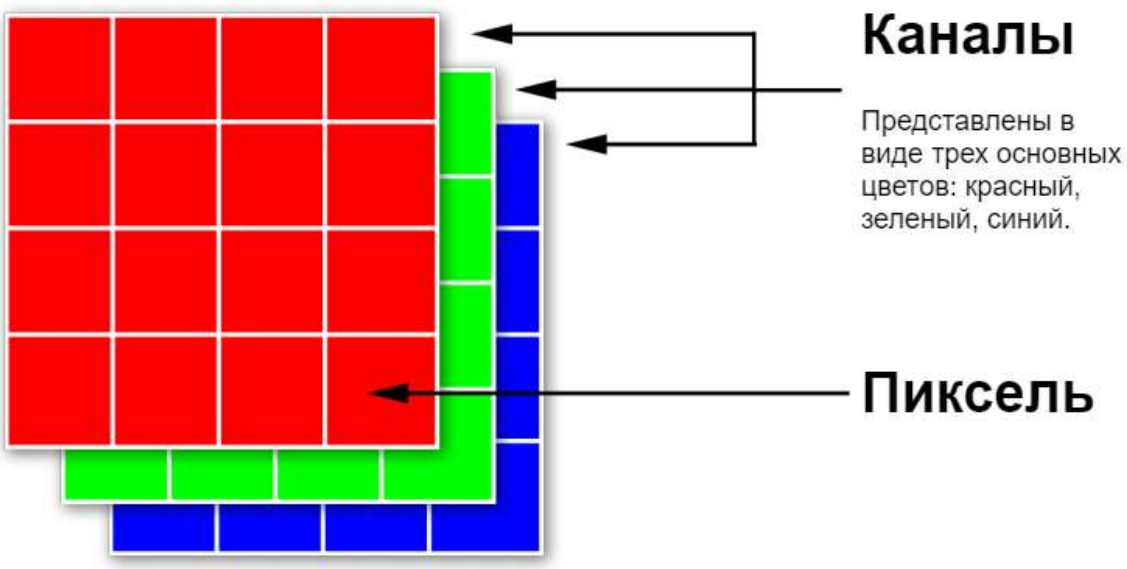


Рисунок 1 – 2D изображение

Как правило, приложения для обработки изображений выполняются с использованием изображений в градациях серого, которые имеют только один канал. Это делает обработку более эффективной.

3 Предварительная обработка

Используемые для обучения изображения, которые содержатся в готовых наборах данных, могут иметь некоторые недостатки, такие как шум и неравномерное освещение, даже после тщательного отбора. Эти недостатки необходимо устранить перед применением алгоритма OCR. Шаги предварительной обработки, которые необходимо предпринять, рассматриваются далее.

3.1 Размытие по Гауссу

Это метод размытия изображения, который использует функцию Гаусса (она также выражает нормальное распределение в статистике) для вычисления значения измененного пикселя для каждого пикселя изображения. Формула функции Гаусса в одном измерении:

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \quad (1)$$

где x - расстояние от начала координат по горизонтальной оси, y - расстояние от начала координат по вертикальной оси, и σ - стандартное отклонение распределения гаусса.

В двух измерениях используется та же формула, но как произведение двух функций Гаусса для каждого измерения:

$$G(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2)$$

Размытие по Гауссу используется для сглаживания изображения и уменьшения шума [1]. Можем наблюдать эффект размытия по Гауссу на рисунках 2 и 3.

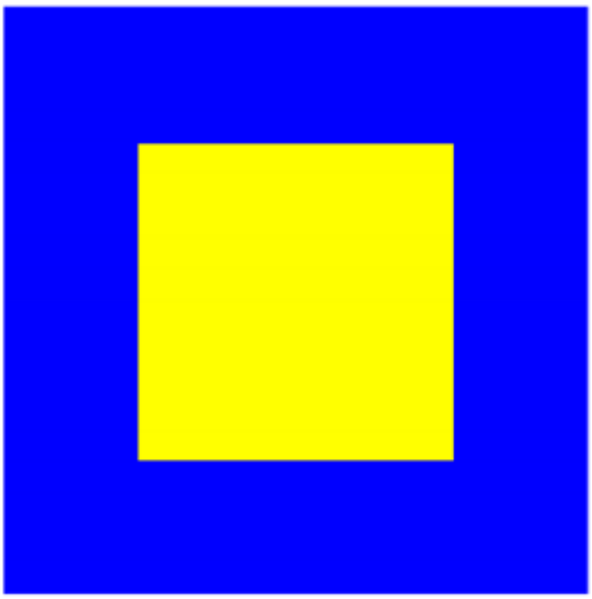


Рисунок 2 – Исходное изображение

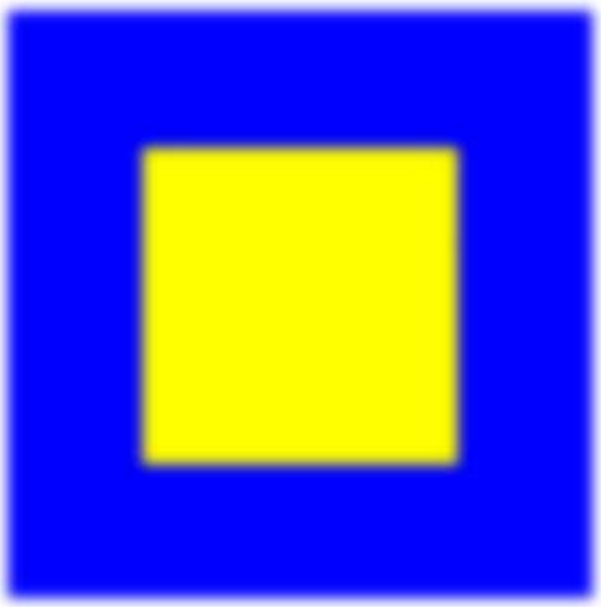


Рисунок 3 – Размытое изображение с использованием размытия по Гауссу

3.2 Пороговая обработка и бинаризация

В идеале текст должен быть написан или напечатан с использованием цвета на контрастном фоне. Другими словами, если текст более темный по цвету, предполагается, что фон будет более светлого цвета (например, черная доска с написанием мелом) и наоборот (например, текст на белой бумаге).

Но, в реалиях, изображение может иметь неравномерное освещение и шум, что негативно сказывается на качестве распознавания. С помощью методов

пороговой обработки и бинаризации можно решить проблему неравномерного освещения и шума.

В качестве решений пороговой обработки в основном используют следующие методы:

5 Ручная установка порога.

6 Адаптивная установка пороговых значений.

7 Метод Оцу.

Каждый из этих методов имеет свои особенности и недостатки, комбинирую которые можно добиться высокого качества выходного изображения, позволяющее более точно распознать символы.

3.2.1 Ручная установка порога

Сначала вручную определяем фиксированное пороговое значение интенсивности и фильтруем текст. Это означает, что для более светлого текста на более темном фоне интенсивность пикселя выше порогового значения, оставляем его и отбрасываем остальные пиксели, заменяя интенсивность нулем и наоборот для более темного текста и более светлого фона. Этого также можно достичь, просто реализуя алгоритм для одной из форм (из темного текста, светлый фон и светлый текст, и темный фон) и инвертируя изображение и используя тот же алгоритм, если в качестве входных данных задана другая форма. Если передаваемое значение интенсивности при установлении порога заменяется максимально возможным значением интенсивности (белый), а другие заменяются самым низким значением (черным), то есть нулевым, мы получаем изображение только в двух цветах.

3.2.2 Адаптивная установка пороговых значений

Этот метод определения порога не принимает фиксированное пороговое значение для всего изображения, но пороговое значение вычисляется для меньших областей, так что оно имеет разные пороговые значения для разных областей изображения. Это особенно полезно, когда изображение содержит участки с неравномерным освещением. Постоянное значение в этом типе

пороговой обработки — это размер блока соседних пикселей, которые принимаются во внимание при вычислении порогового значения.

3.2.3 Метод Оцу

Метод Оцу [2], названный в честь Нобуюки Оцу, используется для выполнения автоматической пороговой обработки изображений. Он автоматически определяет оптимальный порог для изображения. Этот порог определяется путем минимизации дисперсии интенсивности внутри класса или, что эквивалентно, путем максимизации дисперсии между классами. Это эквивалентно глобально оптимальному алгоритму k-средних, выполняемому на гистограмме интенсивности. Этот метод использует как простой, так и адаптивный порог, имея единое пороговое значение для всего изображения и удаляя нежелательный фон, который может возникнуть в адаптивном пороге. Его применение зависит от входного изображения. Мы должны использовать адаптивное определение порога или пороговое значение Оцу, в зависимости от того, что лучше подходит для текущего изображения.

На рисунках 4-7 показаны различия в выходных данных рассмотренных методов определения порога.

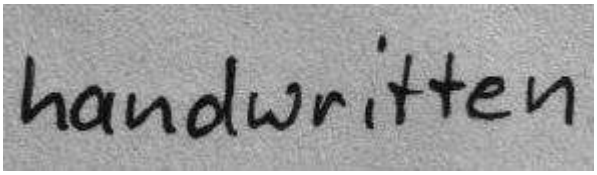


Рисунок 4 – Исходное изображение



Рисунок 5 – Результат ручного определения порога с пороговым значением 127

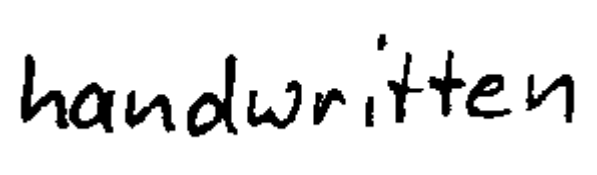


Рисунок 6 – Адаптивная обработка



Рисунок 7 – Результат метода Оцу

Применимость этих методов зависит от сценария использования. Один может работать в одном случае, а другой - в другом. Как правило, адаптивная

установка пороговых значений работает лучше всего, если используются правильные параметры.

3.3 Морфологические операции

Изображение с пороговой обработкой или преобразованием в двоичную форму может содержать нежелательные промежутки и трещины, которые можно заполнить с помощью растяжения соответствующего элемента структурирования. Это особенно необходимо для реконструкции сетки таблицы, когда изображение содержит таблицу, которую необходимо извлечь с сохранением структуры.

Один из способов реконструировать структуру таблицы или сеток - использовать операцию морфологического выделения [3] с достаточно большим структурирующим элементом отдельно как в вертикальном, так и в горизонтальном направлениях. Морфологическая операция выделения представляет собой эрозию с последующим расширением. Это удалит весь текст с изображения, но сохранит и улучшит все вертикальные и горизонтальные линии соответственно. Можно объединить как вертикальные, так и горизонтальные линии, чтобы получить сеточную структуру таблицы (рисунки 8-11).

После реконструкции сетки можно выполнить морфологическую операцию закрытия (расширение с последующей эрозией), чтобы заполнить оставшиеся зазоры или несоответствия в структуре. Существуют и другие, более эффективные методы для достижения лучших результатов, некоторые из них включают машинное обучение.

Employee ID	Name	Salary
101	Adam	1000
102	Drake	2000
103	Smith	3500
104	Lois	10000
105	Freddy	5200
106	Mark	6000
107	Nick	2000
108	Michael	2000

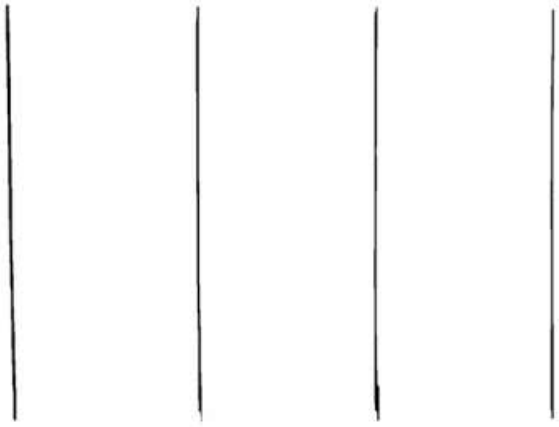


Рисунок 8 – Исходное изображение после определения порога

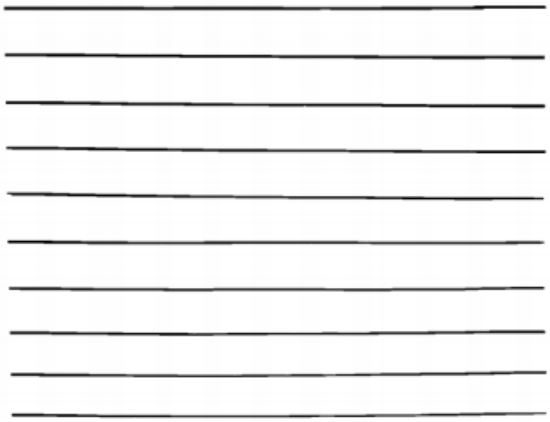


Рисунок 9 – Результат открытия в вертикальном направлении

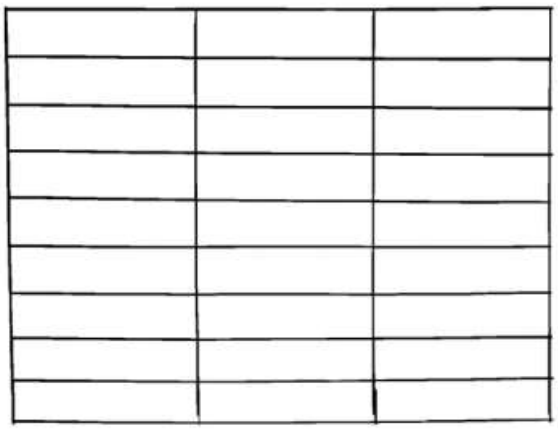


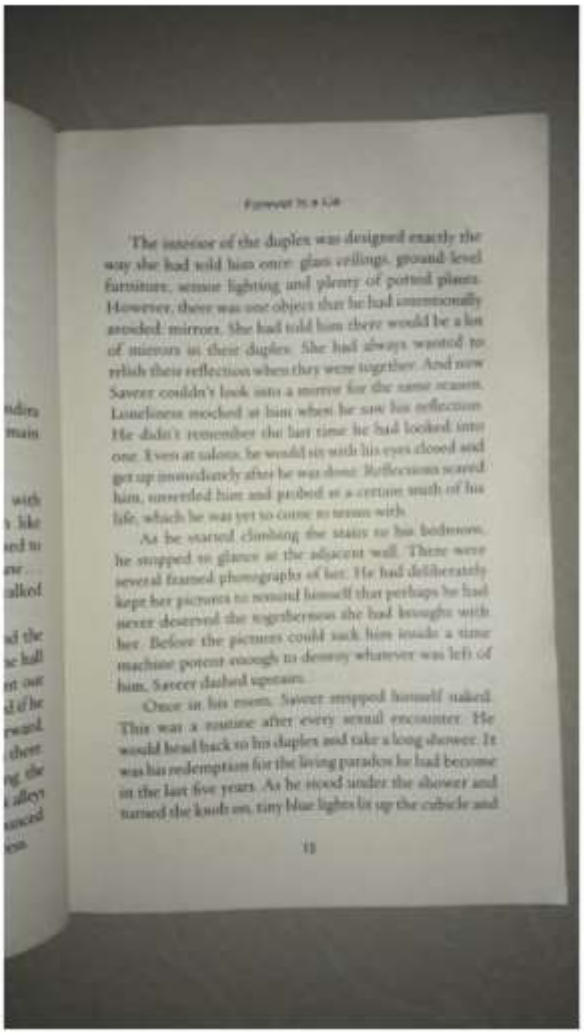
Рисунок 10 – Результат открывания в горизонтальном направлении

Рисунок 11 – Результат после объединения обоих результатов

Этот подход может не работать, если изображение было снято под косым углом. Чтобы решить эту проблему, может потребоваться устранение перекоса и коррекция перспективы.

4 Извлечение страниц

Входное изображение может быть отсканированным документом или фотографией. Если оно содержит всю страницу, необходимо преобразовать ее таким образом, чтобы текст был правильно выровнен, а изображение в целом, оказалось в структуре сверху вниз, как если бы оно было отсканировано. Реализуется это, обнаруживая края с помощью оператора Кэнни [4] на изображении, чтобы получить границы страницы. Затем находится самый большой контур, имеющий четыре стороны. Далее применяется к этому контуру перспективную трансформацию деформации, чтобы растянуть его до прямоугольника подходящего размера (рисунки 12-15).



The interior of the duplex was designed exactly the way she had told him once: glass ceilings, ground-level furniture, sensor lighting and plenty of potted plants. However, there was one object that he had intentionally avoided: mirrors. She had told him there would be a lot of mirrors in their duplex. She had always wanted to relish their reflection when they were together. And now Sawyer couldn't look into a mirror for the same reason. Loneliness mocked at him when he saw his reflection. He didn't remember the last time he had looked into one. Even at salons, he would sit with his eyes closed and get up immediately after he was done. Reflections scared him, unveiled him and probed at a certain truth of his life, which he was yet to come to terms with.

As he started climbing the stairs to his bedroom, he stopped to glance at the adjacent wall. There were several framed photographs of her. He had deliberately kept her pictures to remind himself that perhaps he had never deserved the togetherness she had brought with her. Before the pictures could suck him inside a time machine potent enough to destroy whatever was left of him, Sawyer dashed upstairs.

Once in his room, Sawyer stripped himself naked. This was a routine after every sexual encounter. He would head back to his duplex and take a long shower. It was his redemption for the living paradox he had become in the last five years. As he stood under the shower and turned the knob on, tiny blue lights lit up the cubicle and

13

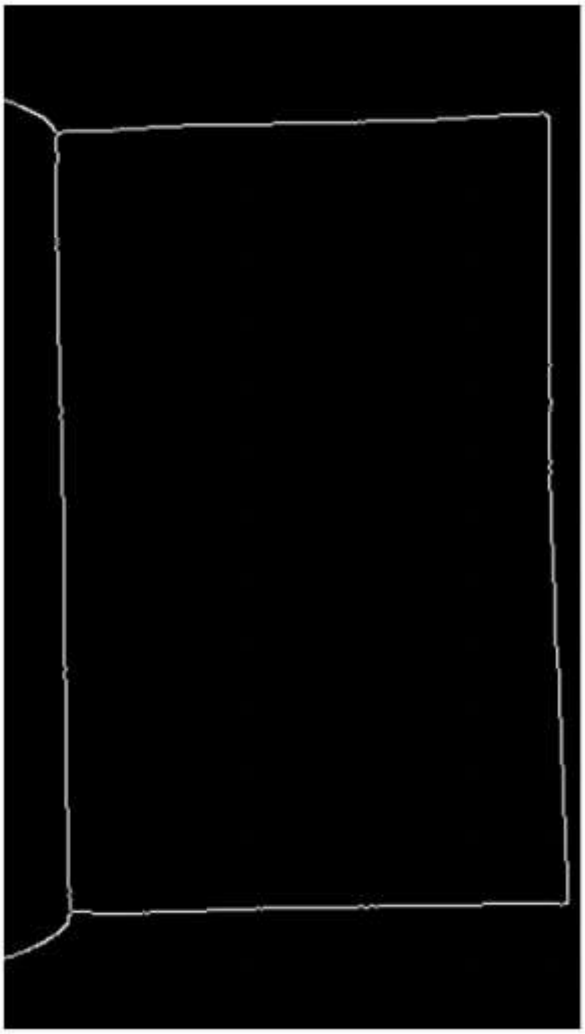


Рисунок 12 – Исходное изображение, снятое камерой телефона

Рисунок 13 – Обнаружение границ с помощью оператора Кэнни

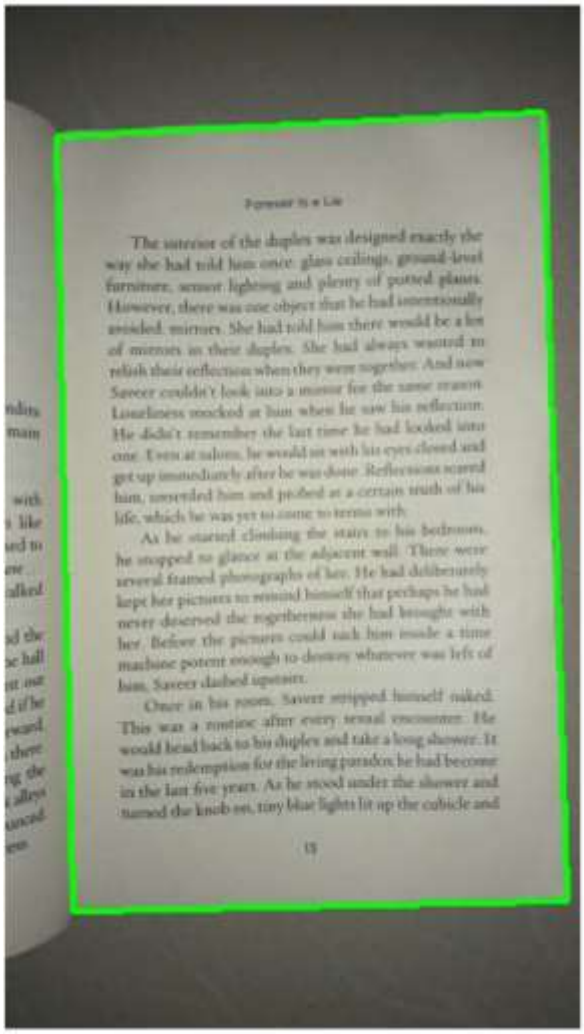


Рисунок 14 – Самый большой контур, имеющий четыре стороны

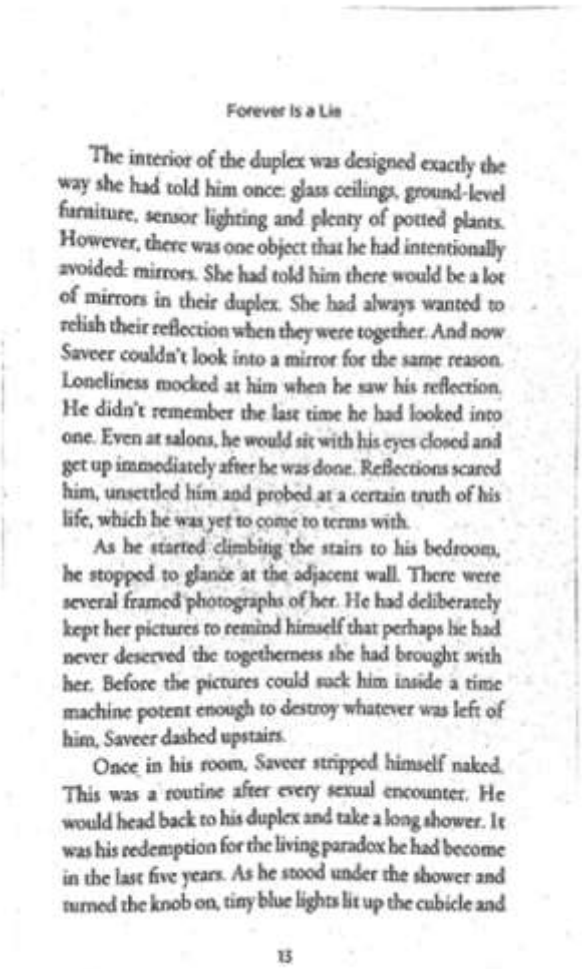


Рисунок 15 – Исправленное изображение, растягивая самый большой контур, имеющий четыре стороны

5 Определение таблицы

Если изображение содержит табличную структуру и имеет видимые границы, то эти границы будут прямоугольной формы. Есть много подходов к идентификации таблиц, но исследуемый метод использует контуры.

5.1 Контуры

Контуры представлены, как кривые, соединяющие все непрерывные точки (вдоль границы), имеющие одинаковый цвет или интенсивность. По сути, это идентификация границ областей изображения схожего цвета.

В бинаризированном изображении сетка таблицы либо черная, либо белая, а фон имеет противоположный цвет. Это дает четкую границу контурам, чтобы покрыть всю площадь изображения. Многие контуры могут быть обнаружены на изображении, представляющем каждую ячейку таблицы, отметки и текста, но в первую очередь необходимо найти всю таблицу. Самый большой контур или контур, покрывающий наибольшую площадь, считается таблицей, и по сути, любой контур внутри таблицы является ячейкой, за исключением текста.

Текст можно игнорировать, игнорируя более мелкие контуры, потому что текст занимает лишь небольшую область по сравнению с ячейками и всей таблицей. В данный момент нужна только вся таблица, а не текст, потому что таблица может нуждаться в правильной ориентации для дальнейшей обработки.

Процесс получения самого большого контура может служить цели преобразования изображения так, чтобы казалось, что изображение было снято под идеальным углом, в то время как на самом деле оно было снято под косым углом, чтобы поток текста и таблицы оставался в правильном направлении (рисунки 16-18).

Alice	12	1000
Bob	13	2000
Carl	14	3000
Smith	14	500

Рисунок 16 – Исходное изображение после операции определения порога

Alice	12	1000
Bob	13	2000
Carl	14	3000
Smith	14	500

Рисунок 17 – Красные границы представляют каждый контур

Alice	12	1000
Bob	13	2000
Carl	14	3000
Smith	14	500

Рисунок 18 – Контуры с меньшими контурами игнорируются, чтобы игнорировать текст

6 Механизм оптического распознавания символов

Механизм оптического распознавания символов или механизм OCR — это реализация алгоритма, которая берет предварительно обработанное изображение и возвращает текст, написанный на нем. Это собственно программа, распознающая текст. Существует множество стандартных подходов глубокого обучения к проблеме распознавания текста. Одними из самых популярных являются YOLO [5], SSD [6], Mask RCNN [7] и Faster RCNN [8]. Эти архитектуры в основном представляют собой идентификаторы объектов, которые можно обучить распознаванию символов. Такие алгоритмы, как Mask RCNN и Faster RCNN, являются идентификаторами на основе области. Это означает, что алгоритм сначала ищет объекты (в данном случае текст) на изображении, а затем классифицирует объекты (в данном случае символы). Этот двухэтапный процесс делает его медленным, но более точным.

Такие алгоритмы, как YOLO и SSDare Single Shot Detectors, одновременно ищут объекты и классифицируют их. Одноэтапный процесс делает их быстрее, но эффективность хуже для небольших объектов, например текста в нашем случае. Любой из ранее упомянутых наборов данных может использоваться для обучения этих моделей, а обученная модель может использоваться для предсказания или распознавания текста в любом входном изображении. Одно из решений в области OCR Tesseract — это средство распознавания текста, которое изначально было разработано HP и было сделано с открытым исходным кодом в 2005 году, а теперь его разработка

спонсируется Google. Он поддерживает более 100 языков с большой точностью благодаря последней версии (4.1.0), основанной на LSTM.

7 Проблема с табличными данными

Табличные данные необходимо вводить в ячейку механизма OCR, чтобы сохранить табличную структуру. Изображение может содержать сотни ячеек, и для их обработки всю процедуру OCR необходимо запускать снова и снова. На это уходит много времени. Если в качестве входных данных используется изображение без табличной структуры, выходные данные могут быть получены в течение нескольких секунд. Но если изображение содержит таблицу с множеством ячеек, получение вывода может занять несколько минут. Скорость обработки - основная проблема табличных данных. Обе таблицы с границами и без них страдают от этой проблемы.

8 Параллельный рабочий поток

Скорость обработки можно повысить, обрабатывая каждую ячейку параллельно, используя параллельную архитектуру ЦП и ГП, которые очень эффективны при распараллеливании небольших и похожих задач. Концепция многопоточности может использоваться для распараллеливания процесса и разделения рабочей нагрузки между несколькими потоками. Общий алгоритм кратко представлен на рисунке 19.

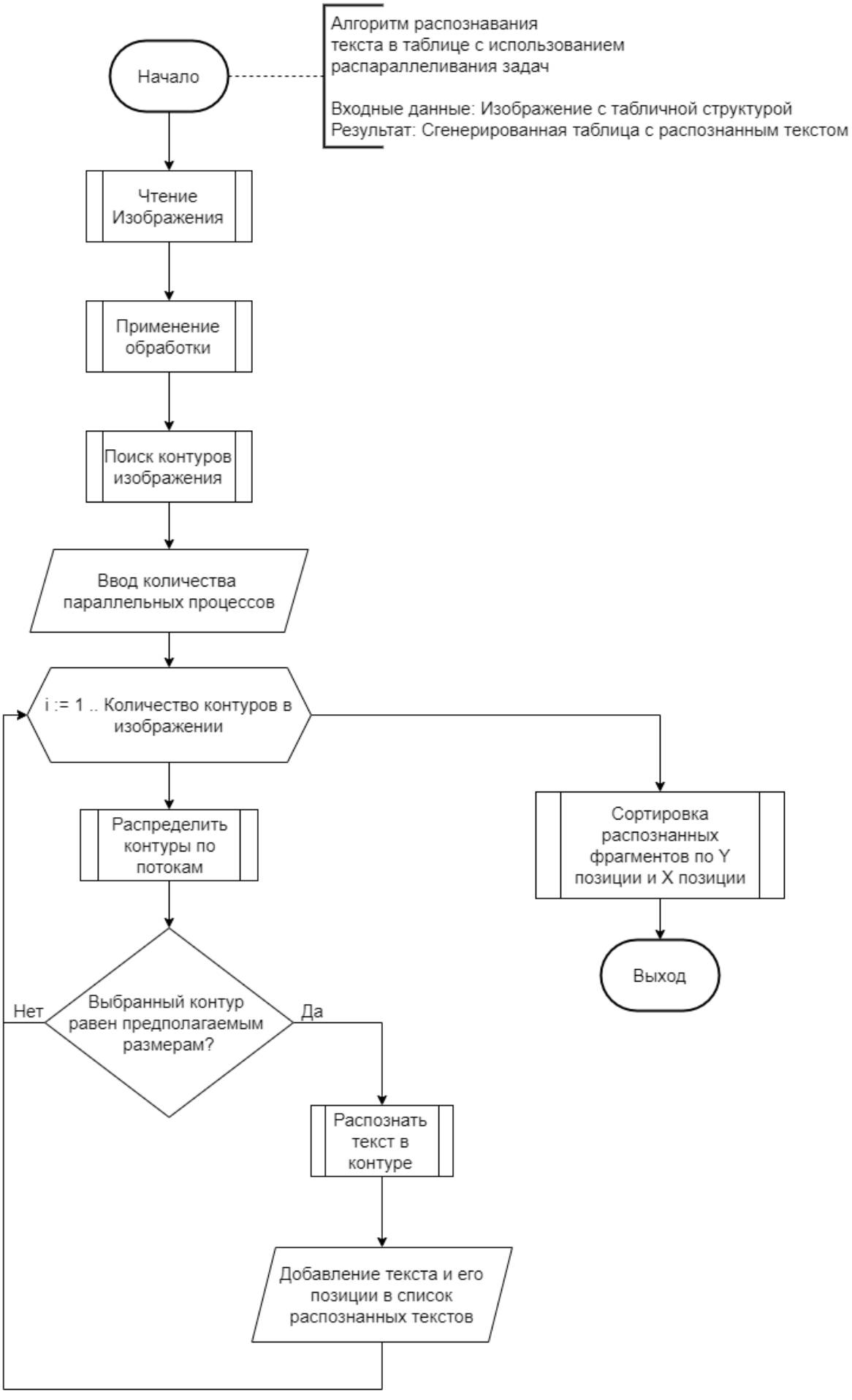

Алгоритм распознавания
текста в таблице с использованием
распараллеливания задач
Входные данные: Изображение с табличной структурой
Результат: Сгенерированная таблица с распознанным текстом
Начало
Чтение Изображения
Применение обработки
Поиск контуров изображения
Ввод количества параллельных процессов
i := 1 .. Количество контуров в изображении
Распределить контуры по потокам
Сортировка распознанных фрагментов по Y позиции и X позиции
Выход
Выбранный контур равен предполагаемым размерам?
Нет
Да
Распознать текст в контуре
Добавление текста и его позиции в список распознанных текстов

Рисунок 19 – Алгоритм распознавания табличных данных в изображении с использованием распараллеливания задач

В некоторых случаях параллельный рабочий поток может быть расширен. Если существует огромное количество изображений, для которых необходимо выполнить оптическое распознавание текста, то несколько изображений можно обрабатывать параллельно, чтобы ускорить общий процесс. Рабочий процесс показан на рисунке 20.

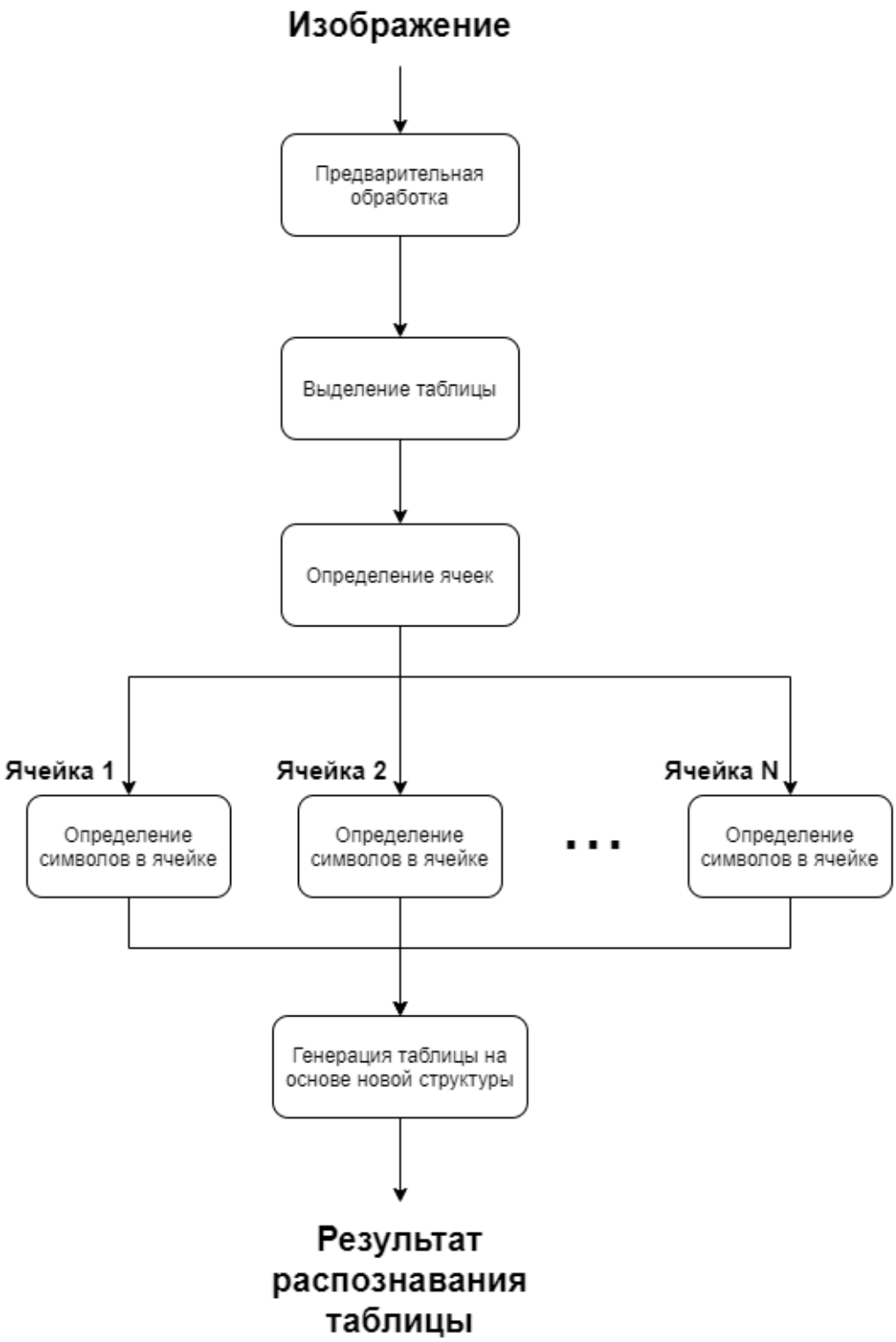


Рисунок 20 – Процесс распознавания текста в таблице

Таким образом, изображение, содержащее табличную структуру, может быть распознано на порядок быстрее с помощью распараллеливания. Это становится доступным из-за однообразия операций над ячейками таблицы.

9 Заключение

В ходе работы исследованы основные методы компьютерного зрения и машинного обучения в области оптического распознавания символов. Изучены и продемонстрированы способы обработки изображений, позволяющие повысить эффективность алгоритмов распознавания.

Для оцифровки изображений с табличной структурой предложен алгоритм распараллеливания задач с целью увеличения скорости распознавания.

Литература

1. Towards data science [Электронный ресурс]. - 2018. - URL: https://towardsdatascience.com/everything-you-ever-wanted-to-know-about-computer-vision-heres-a-look-why-it-s-so-awesome-e8a58dfb641e (дата обращения 2021-04-11).
2. Sezgin, M. and Sankur, B., 2004. Survey over image thresholding techniques and quantitative performance evaluation. Journal of Electronic imaging – P.146-166.
3. Serra, J. and Soille, P. eds., 2012. Mathematical morphology and its applications to image processing - Vol. 2.
4. Canny, J., 1986. A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence – P.679-698.
5. Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2016. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition – P.779-788.
6. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y. and Berg, A.C., 2016, October. Ssd: Single shot multibox detector. In European conference on computer vision - P.21-37.
7. He, K., Gkioxari, G., Dollár, P. and Girshick, R., 2017. Mask r-cnn. In Proceedings of the IEEE international conference on computer vision - P.2961-2969.
8. Ren, S., He, K., Girshick, R. and Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems – P.91-99.

Literature

1. Towards data science [Electronic resource]. - 2018.- URL: https://towardsdatascience.com/everything-you-ever-wanted-to-know-about-computer-vision-heres-a-look-why-it-s-so-awesome-e8a58dfb641e (date of access 2021-04-11).
2. Sezgin, M. and Sankur, B., 2004. Survey over image thresholding techniques and quantitative performance evaluation. Journal of Electronic imaging - P.146-166.
3. Serra, J. and Soille, P. eds., 2012. Mathematical morphology and its applications to image processing - Vol. 2.
4. Canny, J., 1986. A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence - P.679-698.
5. Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2016. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition - P.779-788.
6. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y. and Berg, A.C., 2016, October. Ssd: Single shot multibox detector. In European conference on computer vision - P.21-37.
7. He, K., Gkioxari, G., Dollár, P. and Girshick, R., 2017. Mask r-cnn. In Proceedings of the IEEE international conference on computer vision - P.2961-2969.
8. Ren, S., He, K., Girshick, R. and Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems - P.91-99.