2) Перед автором была поставлена задача определения и выделения

текстовой информации на изображении, для последующего распознавания

изображенного текста. Одна из подзадач процесса распознавания текста из изображения - локализация текстовых блоков и фильтрация нетекстовой информации. Эта задача активно решалась в начале XXI столетия, и большинство описанных в зарубежной литературе методов используют аналогичный подход выделения границ и морфологических операций для локализации текстовых блоков, что и описан ниже. Однако в русскоязычных источниках информации по данной тематике найдено не было.

3) Существует множество OCR программ - программ оптического распознавания символов. Они отлично работают на изображениях, на которых присутствует только текст. Однако на изображениях, в которых текст комбинируется с картинками, многие, в виду внутренней структуры, работают непредсказуемо. В частности, бесплатная библиотека распознавания символов tesseract в сложных условиях может выдавать некорректный результат. Описанный алгоритм требует малого количества вычислений, значительно увеличивая точность распознавания. Кроме того, выделение текстов в блоки позволяет определить координаты текста, занимаемый объем изображения и так далее.

4) Постановка задачи:

На данном изображении выделить

5) текстовую информацию в прямоугольные блоки.

6) Задача разбивается на три этапа

…

7) Подготовка заключается в переводе изображения в черно-белый цвет, использование фильтра Гауса для сглаживания контуров и пороговая бинаризация с высоким порогом. Бинаризация просто закрашивает пиксель черным, если значения яркости выше или равно пороговой и белым иначе. Этот этап значительно упрощает этап выделения границ

8) Следующий этап требует использования свертки. Свертка - математическая операция над изображениями. На вход подается изображение как матрица коэффициентов яркостей и ядро операции свертки. В ядре один из элементов определяется как ядро. Результатом операции свертки является изображение,

9) каждый элемент которого равен сумме произведений элементов исходного изображения с ядром, наложенным на исходную матрицу. Якорь ядра сопоставляется с текущим элементом исходной матрицы, и в сумму произведений входят все элементы, попавшие в наложенное окно. Если элементы ядра выходят за границы исходного изображения, то слагаемые обычно зануляются.

Эту операцию можно значительно ускорить, используя Фуье преобразование

10) Для выделения вертикальных границ удобно использовать результаты сверток с изображенными ядрами.

11) Полученные изображения смешиваются поэлементно логическим оператором или

12) Следующий этап обработки изображения - морфологические операции. Как и в случае свертки, морфологическиая операция определена на изображении и имеет ядро. Существует две основные операции - наращивание и эрозия. Суть операции наращивания заключается в том, что ядро операции накладывается на каждый элемент изображения, и если хотя бы один пиксель из исходного изображения и ядра равен логической единице - в результирующее изображение записывается ядро оператора целиком, якорь которого расположен на позиции текущей проверки. Операция эрозия является полной логической противоположностью операции наращивания.

13) Часто на практике используются комбинации этих операций - открытие и закрытие. Закрытие позволяет убрать дыры в цельных областях, а открытия - очистить картинку от шумов.

14) ..

15) На этом слайде видно результат фильтрации полученных границ. Шаг состоит из композиции морфологических операций и логического смешения каналов, суть которого - оставить только вертикальные границы заданной высоты и объединить области, содержащие множество рядом стоящих границ в блоки. Подробности можно узнать в тексте статьи

16) В результате, обведя полученные блоки прямоугольниками,

17) решим поставленную задачу

18) Реализации описанного алгоритма на языке питон, с использованием библиотек openCV и numpy доступны по ссылке