

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СЕГМЕНТАЦИИ ТЕКСТОВЫХ ИЗОБРАЖЕНИЙ

Попова Е.С.

Научный руководитель: Болотова Ю.А.

Томский политехнический университет, Институт кибернетики
esp9@tpu.ru

Введение

В настоящее время существует необходимость интеллектуальной обработки данных, так как она позволяет автоматизировать многие сферы человеческой деятельности. Одной из таких задач является распознавание текста на изображении. Одним из этапов распознавания является сегментация фрагмента текста, результат которой представляет изображение в виде множества областей, которые соответствуют строкам, словам и символам сегментируемого текстового изображения.

Процесс распознавания текстовых изображений условно можно разделить на два этапа, это процесс сегментирования изображения на строки, слова и символы и процесс распознавания образов. В данной статье будут рассмотрены методы и средства выделения областей расположения отдельных символов на изображении.

Описание алгоритма

Для решения задачи сегментации текстового изображения было разработано приложение Windows Forms в среде Microsoft Visual Studio на языке C#. Выявление областей с символами осуществляется в несколько этапов: предобработка исходного изображения, выделение строк, выделение слов, выделение символов на изображении.

На этапе предобработки исходное изображение обрабатывается серией алгоритмов, для улучшения качества последующей сегментации.

Подавление шума

Подавление шума осуществляется за счет использования медианного фильтра, данный фильтр применяется для минимизации шума и смазывания острых краев букв (засечек и т.п.).

Компенсация разности освещений

Для компенсации разности освещений, используется алгоритм Single-Scale Retinex. Коррекция происходит по формуле:

$$R(x, y) = \log [I(x, y)] - \log [I(x, y) * G(x, y)],$$

где G — фильтр Гаусса, "*" — оператор свертки.

Данный алгоритм для выравнивания освещенности входного изображения получает компоненту освещения путем низкочастотной фильтрации изображения. В данной работе, для размытия изображения применяется двумерный фильтр Гаусса.

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

где σ — стандартное отклонение распределения Гаусса.

После вычисления новых значений яркости, основная часть полученных пикселей будет лежать в диапазоне $[-1; 1]$, для визуализации изображения, значения необходимо нормализовать по формуле:

$$I = 255 * I + 127$$

Бинаризация изображения

Прежде чем произвести бинаризацию изображения, его необходимо перевести в градации серого, яркость пикселя вычисляется по формуле:

$$I = R * 0.299 + G * 0.587 + B * 0.114$$

где R, G, B — красный, зеленый и синий каналы соответственно.

Для последующей бинаризации изображения яркость каждого пикселя $I(x, y)$ сравнивается с некоторым пороговым значением P . Если яркость пикселя больше порогового значения, то цвет пикселя принимается равным 1, иначе 0.

Глобальный порог бинаризации вычисляется по формуле:

$$P = (I_{max} + I_{min}) / 2,$$

где I_{max} — максимальное значение яркости изображения, I_{min} — минимальное значение яркости изображения.

Поворот изображения методом гистограмм

Для автоповорота исходного изображения, анализируется его гистограмма, построенная таким образом, что по оси Y будет располагаться шкала распределения черных пикселей по количеству, а на оси X размещены номера строк, работа алгоритма сводится к поиску угла поворота, при котором гистограмма будет иметь повторяющиеся минимумы и максимумы распределения черных пикселей.

На рисунке 1 представлен результат работы блока предобработки.

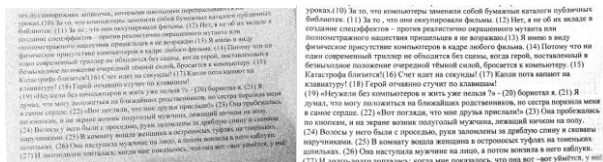


Рисунок 1 - Результат работы этапа предобработки

Сегментирование текстового изображения на строки и слова

Для сегментации фрагмента текста на строки и слова используется метод гистограмм.

Метод предполагает построение гистограммы для черных пикселей бинарного изображения. По оси Y располагается шкала распределения пикселей по количеству, а на оси X размещены номера строк, в случаи сегментирования изображения на строки.

Алгоритм основан на том, что количество черных пикселей в межстрочных интервалах существенно меньше чем в текстовых строках. Основываясь на этом предположении, определим каким должно быть наименьшее количество черных пикселей в строке, чтобы отнести ее к текстовой строке. Рассчитаем значение по формуле:

$$N = 0.1 * Str_{max},$$

где Str_{max} – максимальное число черных пикселей в строке изображения. Следовательно, используя найденный порог разделим все изображение на строки.

Работа алгоритма сегментации строк заключается в последовательном просмотре массива содержащего количество черных пикселей для каждой строки сравнение их с минимальным количеством N и выявлении множества пар индексов (s^1_i, s^2_i) строк, соответствующих границам печатных строк.

Аналогично осуществляется сегментация строк на слова, только в данном случае условием выделения пробелов является последовательность из K белых пикселей в строке изображения.

Методом связанных компонент

Для сегментации слов на символы в работе используется двухпроходной метод связанных компонент (МСК).

Под выделением связанных компонент понимают присвоение уникальной метки каждому объекту изображения. Приследующем анализе данные метки служат в качестве идентификаторов при обращении к объектам.

Для описания алгоритма введем некоторые понятия. Обозначим через I матрицу изображения. Если $I(i, j) = 0$, то пиксель является фоновым. Если $I(i, j) = 1$, то пиксель принадлежит объекту. Через L обозначим двумерную матрицу, скан-маска, которая используется для хранения информации о метках и имеющую размеры, равные размерам изображения.

Первый проход по изображению осуществляется из верхнего левого угла, слева на право и сверху вниз. На внешнем цикле – перебор строк, на внутреннем – перебор столбцов строки, и анализируются только соседи сверху и слева.

Каждый раз при обнаружении черного пикселя его соседи, принадлежащие скан-маске, исследуются для определения метки, которая будет присвоена рассматриваемому пикселю. Если в скан-маске содержатся только фоновые пиксели, то рассматриваемый пиксель получает новую промежуточную метку, если скан-маска содержит только один пиксель интереса, то

рассматриваемый пиксель получает метку соседа. Если скан-маска содержит несколько точек интереса, то их промежуточные метки являются эквивалентными, среди них выбирается метка с наименьшим значением и пикселю присваивается значение выбранной метки-представителя.

После окончания первого прохода каждый объектный пиксель получает временную метку, на втором проходе значение метки уточняется. Второй проход осуществляется в противоположенном направлении, снизу-вверх и с права на лево, на втором проходе исследуются все соседние пиксели.

Значение меток после первого прохода алгоритма представлено на рисунке 2а, б.

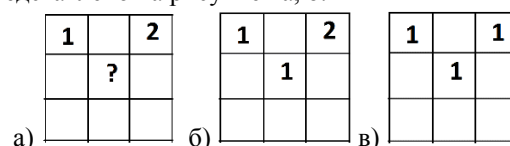


Рисунок 2 - Пример работы МСК на первом проходе (а,б). Результат работы МСК после второго прохода (в)

Следовательно, на втором проходе ищутся связи между маркированными пикселями различных категорий. Если связь найдена, то все «старшие» метки заменяются на «младшие». Результат работы второго прохода представлен на рисунке 2в.

Заключение

В результате проведенных исследований были выявлены факторы имеющее наибольшее влияние на качество сегментирования, к таким факторам относятся:

1. Качество исходного изображения
2. Тип шрифта на изображение
3. Размер шрифта

Были сделаны выводы, что метод связанных компонент не подходит для корректного распознавания всех возможных шрифтов, так как если шрифт предполагает наличие соединения между символами, то алгоритм посчитает эти символы за одну букву, однако несмотря на что метод связанных компонент не является универсальным, в большинстве случаев он справляется с поставленной задачей.

Список использованных источников

1. Поршнева С.В., Левашкина А. О. Универсальная классификация алгоритмов сегментации изображений [Электронный ресурс] // Журнал научных публикаций аспирантов и докторантов. URL: <http://jurnal.org/articles/2008/inf23.html> (дата обращения: 05.03.2015).
2. Цифровая обработка изображений / Гонсалес Р., Вудс Р., 2012. – 1072 с.
3. Компьютерное зрение / Л. Шапиро, Дж. Стакан, 206. – 53 с.