Обучение с неполной информацией от учителя при распознавании текстовых изображений *

Савчинский Б.Д., Олефиренко С.А.

Вступление

Задача распознавания текста, несмотря на свою популярность и кажущуюся изученность, все еще содержит значительное пространство для научных исследований. Одновременно распознавание текстовой строки является ярким и, наверное, одним из простейших примеров в структурном распознавании изображений. Именно по этой причине эта задача является удобным полигоном для внедрения и экспериментальной проверки новых методов в структурном распознавании.

Одной из областей структурного распознавания, которая, на наш взгляд, требует значительных исследований, является обучение и оценка параметров распознающих алгоритмов на основе обучающей выборки. Как известно, обучающая выборка содержит определенное количество изображений и соответствующих им результатов распознавания. В случае распознавания изображения текстовой строки результатом является не только последовательность букв, которая ему отвечает, но и сегментация этой строки на изображения отдельных букв. При этом к сегментации обычно выдвигаются довольно жесткие требования: одинаковые буквы в пределах разных сегментов, содержащих эти буквы, должны быть одинаковым образом отцентрованны. Это значит, что координаты соответствующих пикселей одних и тех же букв на разных сегментах должны быть одинаковыми. Построение такой сегментации требует значительных усилий и времени учителя.

В данной работе мы предлагаем постановку и алгоритм решения задачи оценки параметров алгоритма распознавания текстовой строки на основе обучающей выборки, которая содержит лишь примеры изображений текстовых строк и соответствующих им последовательностей букв и не содержит сегментации изображений на изображения отдельных букв. Такая формулировка задачи обучения позволяет значительно упростить и ускорить построение обучающей выборки, сведя ее к простому набору текста, который соответствует обучающим изображениям.

Работа состоит из четырех разделов, первый из которых посвящен основным определениям и постановке задачи, второй — ее решению, третий и четвертый, соответственно — экспериментальной проверке алгоритмов и выводам.

*Работа выполнялась в рамках проекта EU INTAS PRINCESS 04-77-7347

1 Определение и постановка задачи обучения

Введем основные обозначения, которые будут использоваться в работе.

Полем зрения T назовем прямоугольное подмножество двумерной целочисленной решетки — множество координат пикселей изображения:

$$T = \{\, (i, j) \mid i = \overline{0, W-1}, j = \overline{0, H-1} \,\}.$$

Величину W будем называть шириной, а H — высотой поля зрения. Элементы $t \in T$ поля зрения будем считать обычными двухмерными векторами, в частности определена операция сложения двух элементов как покомпонентное сложение соответствующих координат пикселей изображения.

Пусть V — множество значений яркости пиксела. *Изображением* назовем функцию $x\colon T \to V$, его высота и ширина совпадают с высотой и шириной поля зрения, соответственно. Мы будем различать два типа изображений: изображения, поданные на распознавание, и эталонные изображения букв, о которых речь пойдет ниже в этом разделе. Будем считать, что все эти изображения имеют одинаковую высоту H, но, вообще говоря, разную ширину.

Конечное множество A_0 будем называть *алфавитом*. Элементами алфавита являются буквы текста. Последовательность элементов алфавита $\bar{k} = (k_1, k_2, \ldots, k_L)$, $k_l \in A_0, l = \overline{1, L}$ будем называть *текстовой строкой*. При помощи $L_{\bar{k}}$ будем обозначать в дальнейшем длину строки $\bar{k}$.

С каждой буквой $k \in A_0$ свяжем ее эталонное изображение e_k, определенное на поле зрения высоты H и ширины $d(k)$, которая зависит от буквы. Ширины $d(k), k \in A_0$ эталонных изображений всех букв буем считать фиксированными и известными. Множество эталонных изображений обозначим через E.

Будем считать, что не зашумленное изображение, которое соответствует заданной текстовой строке, является горизонтальной последовательностью эталонных изображений букв строки, причем эти изображения не перекрываются, а возможные промежутки между ними заполняются цветом фона.

Для формального описания промежутков между изображениями букв введем дополнительный элемент алфавита. Назовем его *вставкой* и обозначим через κ. Будем считать, что эталон вставки имеет ширину $d(\kappa) = 1$, высоту H и принадлежит множеству эталонных изображений E. Множество $A_0 \bigcup \{\kappa\}$, которое состоит из алфавита A_0 и вставки κ, будем обозначать как A, а ее элементы $a \in A$ будем называть *символами*. Таким образом, символ $a \in A$ является или буквой алфавита $k \in A_0$, или вставкой κ.

Будем называть сегментом поименованный прямоугольный фрагмент, который содержит изображение определенного символа. При этом высота фрагмента совпадает с высотой H входного изображения, а ширина — с шириной соответствующего символа. Таким образом сегмент s определяется координатой своего левого края $i = \overline{0, W - d(a)}$ и символом $a \in A$, изображение которого он содержит. Координату левого края сегмента $s = (i, a)$ будем обозначать $i(s)$, а связанный с ним символ — $a(s)$. Множество всех сегментов обозначим S. Свяжем с каждым сегментом элемент t_s поля зрения изображения, координаты которого совпадают с координатами левого края сегмента:

$t_s = (i, 0)$.

Сегментацией изображения назовем последовательность сегментов $\bar{s} = (s_1, \ldots, s_N)$ произвольной длины N, которые покрывают все поле зрения и расположены вплотную один к другому:

$$\begin{cases} i(s_1) = 0; \\ i(s_{n+1}) = i(s_n) + d(a(s_n)), n = \overline{1, N-1}; \\ i(s_N) + d(a(s_N)) = W. \end{cases}$$

Множество всех сегментаций обозначим $\bar{S}$.

Будем считать, что входное изображение отличается от не зашумленного изображения, процесс построения которого из последовательности символов описан выше, лишь гауссовским шумом определенной дисперсии σ, который накладывается в каждом пикселе независимо от остальных. Таким образом, вероятность $p(x|\,\bar{s}; E)$ изображения x при условии известных сегментаций $\bar{s}$ и множества эталонов E принимает вид

$$p(x|\,\bar{s}; E) = \prod_{n=1}^{N(\bar{s})} p(x|\,s_n; E) = \prod_{n=1}^{N(\bar{s})} \prod_{t \in T(s_n)} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{(x(t_{s_n} + t) - e_{a(s_n)}(t))^2}{2\sigma^2} \right\}, \qquad (1)$$

где при помощи $N(\bar{s})$ обозначено количество сегментов в сегментации $\bar{s}$, а при помощи $T(s)$ — прямоугольный фрагмент поля зрения, который отвечает сегменту s.

Задача распознавания изображения x состоит в поиске наиболее вероятной сегментации $\bar{s}^*$:

$$\bar{s}^* = \arg\max_{\bar{s}} p(x, \bar{s}; E) = \arg\max_{\bar{s}} p(\bar{s}) \cdot p(x|\,\bar{s}; E).$$

Как известно [1], она решается при помощи алгоритма динамического распознавания.

Обучение алгоритма распознавания, которому собственно и посвящена данная работа, состоит в оценке значений параметров этого алгоритма, которыми являются множество эталонов E и априорное распределение сегментаций $\{p(\bar{s}) \mid \bar{s} \in \bar{S}\}$, на основании обучающей выборки.

Прежде чем перейти к формулировке задачи, отметим связь между сегментациями и последовательностями букв алфавита. Произвольной сегментации $\bar{s} = (s_1, \ldots, s_N)$ соответствует последовательность символов $\bar{a}(\bar{s}) = (a_1, \ldots, a_N | a_n = a(s_n), n = \overline{1, N})$. В свою очередь, этой последовательности символов соответствует последовательность букв, которая получается из нее удалением всех вставок. Таким образом, любая последовательность букв $\bar{k} = (k_1, \ldots, k_N), k_n \in A_0$ связана с множеством $\bar{S}(\bar{k})$ всех сегментаций таких, что соответствующие им текстовые строки после удаления всех вставок совпадают с $\bar{k}$.

Перейдем к формулировке задачи обучения. Пусть

$$D = \begin{pmatrix} x^1 & x^2 & \ldots & x^M \\ \bar{k}^1 & \bar{k}^2 & \ldots & \bar{k}^M \end{pmatrix} -$$

обучающая выборка, которая состоит из M входных изображений и M соответствующих им текстовых строк.

Вероятность $p(x, \bar{k}; E)$ пары $(x, \bar{k})$ изображения x и текстовой строки $\bar{k}$, равна суммарной вероятности $\sum_{\bar{s} \in \bar{S}(\bar{k})} p(x, \bar{s}; E)$ всех сегментаций, которые соответствуют строке $\bar{k}$. Вероятность выборки

$p(D, E)$ таким образом принимает вид

$$p(D; E) = \prod_{m=1}^{M} p(x^m, \bar{k}^m; E) = \prod_{m=1}^{M} \sum_{\bar{s} \in \bar{S}(\bar{k}^m)} p(x^m, \bar{s}; E) = \prod_{m=1}^{M} \sum_{\bar{s} \in \bar{S}(\bar{k}^m)} p(\bar{s}) \cdot p(x^m |\, \bar{s}; E).$$

Задача 1 *Задача обучения алгоритма распознавания состоит в нахождении таких эталонов E^* и априорных вероятностей сегментаций $p^*(\bar{s})$, которые максимизируют вероятность выборки D:*

$$\begin{aligned}(E^*, p^*(\bar{s})) &= \arg \max_{(E, p(\bar{s}))} \prod_{m=1}^{M} \sum_{\bar{s} \in \bar{S}(\bar{k}^m)} p(\bar{s}) \cdot p(x^m |\, \bar{s}; E) = \\ &= \arg \max_{(E, p(\bar{s}))} \prod_{m=1}^{M} \sum_{\bar{s} \in \bar{S}(\bar{k}^m)} \frac{p(\bar{s})}{\sqrt{2\pi\sigma^2}} \exp \left\{ - \sum_{n=1}^{N(\bar{s})} \sum_{t \in T(s_n)} \frac{(x^m(t_{s_n} + t) - e_{a(s_n)}(t))^2}{2\sigma^2} \right\}. \quad (2)\end{aligned}$$

Нам неизвестен алгоритм точного решения задачи 1. В данной работе для поиска решения был использован алгоритм самообучения, описанный в [1]. Как известно, этот алгоритм, вообще говоря, обеспечивает поиск лишь локального экстремума. Вместе с тем, с практической точки зрения это не составляет значительной проблемы, поскольку качество оценки параметров может быть легко проконтролировано визуально, а на основе результатов распознавания изображений из выборки может быть оценено качество распознавания изображений, которые не вошли в нее.

Вместе с тем, алгоритм самообучения, описанный в [1], не может быть использован для решения задачи 1 непосредственно, поскольку требует экспоненциальных по размерам входных изображений времени и памяти. В следующем разделе описано, каким образом этот алгоритм должен быть реализован для своего эффективного использования при решении задачи 1.

2 Решение задачи обучения

Сначала сформулируем для задачи 1 алгоритм самообучения в том виде, в котором он описан в [1]. Как уже было сказано, в таком виде он не может быть использован непосредственно из-за своей значительной временной сложности. После этого преобразуем его эквивалентным образом, существенно снизив его сложность, оставив при этом без изменений суть выполняемых операций.

2.1 Базовый алгоритм самообучения

Прежде всего, введем дополнительные обозначения. Для множества $\bar{S}(\bar{k}^m)$ сегментаций, символьные строки которых после удаления всех вставок совпадают с m-той строкой из обучающей выборки $\bar{k}^m,\ m = \overline{1, M}$, введем эквивалентное обозначение $\bar{S}_m$. Через $\bar{S}_m(s)$ будем обозначать подмножество множества $\bar{S}_m$, состоящее лишь из тех сегментаций, которые содержат сегмент s.

Алгоритм самообучения является итеративным. При помощи верхнего индекса r будем обозначать значения величин, которые они принимают на r-ой итерации алгоритма. Пусть $E^0, p^0(\bar{s}),\ \bar{s} \in \bar{S}$ — соответственно начальные значения эталонов символов и априорное распределение сегментаций изображения. Каждая итерация алгоритма состоит из двух шагов. На первом шаге (распознавание) вычисляются оценки $\hat{\alpha}^r(x^m, \bar{s}),\ m = \overline{1, M}, \bar{s} \in \bar{S}_m$ апостериорных распределений сегментаций $\bar{s}$ для

каждого обучающего изображения x^m:

$$\hat{\alpha}^r(x^m, \bar{s}) = \frac{p^r(\bar{s}) \cdot p(x^m|\bar{s}; E^r)}{\sum\limits_{\bar{s} \in \bar{S}_m} p^r(\bar{s}) \cdot p(x^m|\bar{s}; E^r)}, \quad m = \overline{1, M}, \ \bar{s} \in \bar{S}_m. \tag{3}$$

На втором шаге (обучение) оцениваются априорные вероятности сегментаций $p^{r+1}(\bar{s}),\ \bar{s} \in \bar{S}$ и эталонные изображения символов E^{r+1} в соответствии с формулами:

$$p^{r+1}(\bar{s}) = \frac{\sum\limits_{m=1}^{M} \hat{\alpha}^r(x^m, \bar{s})}{M}, \quad \bar{s} \in \bar{S}; \tag{4}$$

$$E^{r+1} = \arg\max_{E} \sum_{m=1}^{M} \sum_{\bar{s} \in \bar{S}_m} \hat{\alpha}^r(x^m, \bar{s}) \cdot \log p(x^m|\bar{s}; E^r). \tag{5}$$

Реализация алгоритма самообучения в виде (3)–(5) невозможна, поскольку величины $\hat{\alpha}^r(x^m, \bar{s})$ должны быть вычислены для всех возможных сегментаций изображений выборки, количество которых растет экспоненциально с размерами изображений. Модификация алгоритма, которая лежит в основе эффективной реализации, состоит в итеративном вычислении величин, связанных с отдельными сегментами, а не с сегментациями в целом, как это происходит в базовом алгоритме.

2.2 Эффективная реализация алгоритма самообучения

Будем считать, что отдельные сегменты любой сегментации являются независимыми один от другого, то есть что вероятность сегментации $\bar{s}$ определяется формулой:

$$p(\bar{s}) = \prod_{n=1}^{N(\bar{s})} p(s_n), \tag{6}$$

где $p(s),\ s \in S$ — априорные вероятности сегментов. Очевидно также следующее равенство:

$$p(s) = \sum_{\bar{s} \in \bar{S}_m(s)} p(\bar{s}), \quad s \in S. \tag{7}$$

Вместо величин $\hat{\alpha}(x, \bar{s})$, которые соответствуют апостериорным вероятностям сегментаций изображения x, введем величины $\alpha(x, s),\ s \in S$, которые являются оценками апостериорных вероятностей отдельных сегментов:

$$\alpha^r(x^m, s) = \sum_{\bar{s} \in \bar{S}_m(s)} \hat{\alpha}^r(x^m, \bar{s}) = \frac{\sum\limits_{\bar{s} \in \bar{S}_m(s)} p^r(\bar{s}) \cdot p(x^m|\bar{s}; E^r)}{\sum\limits_{\bar{s} \in \bar{S}_m} p^r(\bar{s}) \cdot p(x^m|\bar{s}; E^r)}, \quad m = \overline{1, M}, \ s \in S. \tag{8}$$

Подставив (1) и (6) в (8), получим:

$$\alpha^r(x^m, s) = \frac{\sum\limits_{\bar{s} \in \bar{S}_m(s)} \prod\limits_{n=1}^{N(\bar{s})} p^r(s_n) \cdot p(x^m|\, s_n; e^r_{a(s_n)})}{\sum\limits_{\bar{s} \in \bar{S}_m} \prod\limits_{n=1}^{N(\bar{s})} p^r(s_n) \cdot p(x^m|\, s_n; e^r_{a(s_n)})}, \quad m = \overline{1, M}, \ s \in S. \tag{9}$$

Вычисления по формуле (9) не могут быть проведены непосредственно, но дальше в подразделе 2.3 будет указан эффективный алгоритм таких вычислений, основанный на методе динамического программирования.

Перейдем к формуле (4). Для произвольного фиксированного сегмента $s \in S$ просуммируем (4) по всем сегментациям, которые его содержат. Тогда, в соответствии с (7) и (8), получим:

$$p^{r+1}(s) = \frac{\sum\limits_{m=1}^{M} \alpha^r(x^m, s)}{M}, \quad s \in S. \tag{10}$$

Величины $p^{r+1}(s)$ могут быть вычислены непосредственно по данной формуле.

Множество эталонов E состоит из значений всех пикселей эталонов всех символов: $E = \{e_a(t) \mid t \in T_a, a \in A\}$. Максимум суммы (5) определяется системой уравнений:

$$\begin{cases} \frac{\partial}{\partial e_a(t)} \sum_{m=1}^{M} \sum_{\bar{s} \in \bar{S}_m} \hat{\alpha}^r(x^m, \bar{s}) \cdot \log p(x^m | \bar{s}; E^r) = 0, \\ t \in T_a, a \in A. \end{cases}$$

Из этой системы следует, что оптимальный эталон для произвольного символа $a \in A$ строится взвешенным усреднением соответствующих фрагментов изображений с весами $\alpha(x^m, s)$ по всем сегментам с именем a:

$$e_a^{r+1}(t) = \frac{\sum\limits_{\substack{s \in S: \\ a(s)=a}} \sum\limits_{m=1}^{M} \alpha^r(x^m, s) \cdot x^m(t_s + t)}{\sum\limits_{\substack{s \in S: \\ a(s)=a}} \sum\limits_{m=1}^{M} \alpha^r(x^m, s)}, \quad t \in T_a. \tag{11}$$

Таким образом, от базового алгоритма (3)–(5) мы перешли к алгоритму, который решает ту же задачу, но оценивает другое множество параметров, которое состоит из эталонов всех символов и априорных вероятностей сегментов: $\{e_a, p(s) \mid a \in A, s \in S\}$. В отличие от базового полученный алгоритм (9)–(11) может быть эффективно реализован: величины $p(s),\ s \in S$ и эталоны символов $e_a,\ a \in A$ могут быть вычислены непосредственно по формулам (10) и (11), а алгоритм вычисления величин $\alpha^r(x, s),\ s \in S$ в соответствии с (9) приведен в следующем подразделе.

2.3 Алгоритм вычисления апостериорных вероятностей сегментов $\alpha(x, s)$

В этом подразделе ми рассмотрим алгоритм вычисления апостериорных вероятностей сегментов $\alpha(x, s)$ для произвольного изображения x и соответствующей ему текстовой строки $\bar{k}$. Обозначим длину строки $\bar{k}$ через L: $\bar{k} = (k_1, \ldots, k_l, \ldots, k_L)$. Вероятность $p(s) \cdot p(x|\, s; e_{a(s)})$ сегмента $s = (i, a)$ будем называть штрафом за этот сегмент и будем обозначать через $f(i, a)$. Штрафом за сегментацию $\bar{s}$ будем называть произведение штрафов за сегменты, из которых она состоит. Будем различать сегменты, именем которых является буква из множества A_0 — такие сегменты мы будем называть *значащими* — и сегменты, именем которых является вставка.

Нас будут интересовать только сегментации, которые отвечают строке $\bar{k}$, то есть сегментации из множества $\bar{S}(\bar{k})$, которое мы для простоты дальнейшей записи обозначим через $\bar{S}^\circ$. Такие сегментации содержат L значащих сегментов, которые отвечают буквам строки, и произвольное количество вставок. Введем функцию $\mathrm{sgf}\colon \{1, \ldots, L\} \times \bar{S}(\bar{k}) \to S$, значение которой $\mathrm{sgf}(l, \bar{s})$ указывает на l-ый значащий сегмент сегментации $\bar{s}$, то есть на сегмент, который отвечает букве с порядковым номером l строки $\bar{k}$. Кроме этого, введем две функции $b_L(l, \bar{s})$ та $b_R(l, \bar{s})$, $l = \overline{1, L}$, которые принимают значения, соответственно, координат левого та правого краев l-го значащего сегмента.

Для произвольной сегментации $\bar{s}$ сегмент с координатой левого края 0 будем называть началом сегментации, а сегмент с координатой правого края W — ее концом.

Як видно из формулы (9), для произвольного сегмента $s = (i, a)$ величина $\alpha(x, s)$ равна отношению суммарного штрафа за все сегментации, которые содержат сегмент s, к суммарному штрафу за все сегментации. Первая из этих сумм может быть разбита на произведение двух сумм: суммы штрафов за части сегментаций с начала до сегмента s и суммы штрафов за части сегментаций от сегмента s до конца. Будем различать два случая вычисления величины $\alpha(x, s)$: случай, когда s является значащим сегментом, и случай, когда s соответствует сегменту-вставке. В первом случае числитель формулы (9) принимает вид:

$$\sum_{\bar{s}\in\bar{S}^\circ(s)}\prod_{n=1}^{N(\bar{s})} p(s_n)\cdot p(x|\, s_n; e_{a(s_n)}) = \sum_{\substack{1\le l\le L:\\ k_l=a}}\sum_{\substack{\bar{s}\in\bar{S}^\circ(s)\\ \mathrm{sgf}(l,\bar{s})=s}} f(i_1,a_1)\cdot f(i_2,a_2)\cdot\ldots\cdot f(i_{N(\bar{s})},a_{N(\bar{s})}) = \tag{12}$$

$$= \sum_{\substack{1\le l\le L:\\ k_l=a}}\left[\sum_{\substack{\bar{s}\in\bar{S}^\circ(s)\\ \mathrm{sgf}(l,\bar{s})=s}} f(i_1,a_1)\cdot\ldots\cdot f(i,a)\right]\times\left[\sum_{\substack{\bar{s}\in\bar{S}^\circ(s)\\ \mathrm{sgf}(l,\bar{s})=s}} f(i,a)\cdot\ldots\cdot f(i_{N(\bar{s})},a_{N(\bar{s})})\right]\times\frac{1}{f(i,a)},$$

где $\bar{s} = (s_1, s_2, \ldots, s_{N(\bar{s})})$, $s_n = (i_n, a_n)$. Штраф за сегмент $s = (i, a)$ содержится в обоих выражениях в квадратных скобках, поэтому мы делим выражение на этот штраф для компенсации. Отметим также, что условия $\mathrm{sgf}(l, \bar{s}) = s$, $b_R(l, \bar{s}) = i + d(k_l)$ и $b_L(l, \bar{s}) = i$ идентичны, поэтому в дальнейшем вместо первого условия будут использоваться два остальных.

Выражения в квадратных скобках зависят от l — номера l-ой буквы в текстовой строке — и сегмента $s = (i, k_l)$. Обозначим первый из них через $F_1(i + d(k_l), l)$, а второй — через $B_1(i, l)$, где $i + d(k_l)$ — координата правого края сегмента s:

$$F_1(i,l) = \sum_{\substack{\bar{s}\in\bar{S}^\circ(s):\\ b_R(l,\bar{s})=i}} f(i_1,a_1)\cdot f(i_2,a_2)\cdot\ldots\cdot f(i-d(k_l),k_l), \tag{13}$$

$$B_1(i,l) = \sum_{\substack{\bar{s}\in\bar{S}^\circ(s):\\ b_L(l,\bar{s})=i}} f(i,k_l)\cdot\ldots\cdot f(i_{N(\bar{s})},a_{N(\bar{s})}) \tag{14}$$

Для произвольной пары (i, l) рассмотрим все сегментации, в которых l-ым значащим сегментом является сегмент $s(i, k_l)$. Тогда $F_1(i + d(k_l), l)$ равняется суммарному штрафу за част этих сегментаций с начала до сегмента s включительно; $B_1(i, l)$ равняются суммарному штрафу за части этих сегментаций от сегмента s включительно до конца.

Аналогично преобразуем формулу (9) для случая, когда $s = (i, a)$ соответствует сегменту-вставке ($a(s) = \kappa$):

$$\sum_{\bar{s}\in\bar{S}^\circ(s)}\prod_{n=1}^{N(\bar{s})} p(s_n)\cdot p(x|\, s_n; e_{a(s_n)}) = \sum_{l=0}^{L}\sum_{\substack{\bar{s}\in\bar{S}^\circ(s)\\ b_R(l,\bar{s})\le i,\\ b_L(l+1,\bar{s})\ge i+1}} f(i_1,a_1)\cdot f(i_2,a_2)\cdot\ldots\cdot f(i_{N(\bar{s})},a_{N(\bar{s})}) = \tag{15}$$

$$= \sum_{l=0}^{L}\left[\sum_{\substack{\bar{s}\in\bar{S}^\circ(s)\\ b_R(l,\bar{s})\le i,\\ b_L(l+1,\bar{s})\ge i+1}} f(i_1,a_1)\cdot\ldots\cdot f(i,a)\right]\times\left[\sum_{\substack{\bar{s}\in\bar{S}^\circ(s)\\ b_R(l,\bar{s})\le i,\\ b_L(l+1,\bar{s})\ge i+1}} f(i,a)\cdot\ldots\cdot f(i_{N(\bar{s})},a_{N(\bar{s})})\right]\times\frac{1}{f(i,a)}.$$

Обозначим первое выражение в квадратных скобках через $F_0(i+1,l)$, а второе — через $B_0(i,l)$:

$$F_0(i,l) = \sum_{\substack{\bar{s}\in\bar{S}^\circ(s):\\ b_R(l,\bar{s})\le i-1,\\ b_L(l+1,\bar{s})\ge i}} f(i_1,a_1)\cdot f(i_2,a_2)\cdot\ldots\cdot f(i-1,\kappa), \tag{16}$$

$$B_0(i,l) = \sum_{\substack{\bar{s}\in\bar{S}^\circ(s):\\ b_R(l,\bar{s})\le i,\\ b_L(l+1,\bar{s})\ge i+1}} f(i,k_l)\cdot\ldots\cdot f(i_{N(\bar{s})},a_{N(\bar{s})}), \tag{17}$$

где для единообразия записи положено $b_R(0,\bar{s})=0,\ b_L(L+1,\bar{s})=W$.

Рассмотрим все сегментации, в которых присутствует сегмент $s=(i,\kappa)$, перед которым встретилось ровно l значащих сегментов. $F_0(i+1,l)$ равняется суммарному штрафу за части сегментаций с начала до сегмента s включительно; $B_0(i,l)$ равняется суммарному штрафу за части сегментаций от сегмента s включительно до конца.

Поскольку последним (с координатой правого края равной W) сегментом всякой сегментации может быть или L-й значащий сегмент, или же вставка, перед которой встретились все L значащих сегментов, то суммарный штраф за все сегментации (знаменатель формулы (9)) равняется

$$Z = \sum_{\bar{s}\in\bar{S}^\circ}\prod_{n=1}^{N(\bar{s})} p(s_n)\cdot p(x|\,s_n;e_{a(s_n)}) = F_0(W,L)+F_1(W,L). \tag{18}$$

Подобными соображениями можно выразить Z через величины B_0 та B_1:

$$Z = B_0(0,0)+B_1(0,0).$$

Величины F_0 и F_1 могут рассматриваться как суммы штрафов за пути на некотором графе. Рассмотрим граф, вершинами которого являются координаты столбиков $i=\overline{0,W}$ поля зрения изображения, а ребра соответствует сегментам на этом изображении: сегменту $s=(i,a)$ соответствует ребро $(i\to i+d(a))$ с именем a, которое мы обозначим $\varepsilon(i,i+d(a),a)$. При этом на графе присутствуют ребра для тех и только тех сегментов, которые встречаются в сегментациях из $\bar{S}(\bar{k})$. С каждым ребром $\varepsilon(i,i+d(a),a)$ на графе связан штраф $f(i,a)$. Штрафом за путь является произведение штрафов его ребер. Ребро будем называть *значащим*, если соответствующий ему сегмент — значащий. Тогда величина $F_1(i,l)$ равняется суммарному штрафу за все пути на графе от вершины 0 до вершины i, которые содержат l значащих ребер и последним ребром которых является значащее ребро; $F_0(i,l)$ равняется суммарному штрафу за все пути на графе от вершины 0 до вершины i, которые содержат l значащих ребер и последним ребром которых является не значащее ребро. Соответственно, штраф за все сегментации Z равняется суммарному штрафу за все пути на графе от вершины 0 до вершины W, которые содержат L значащих ребер.

Аналогично величины B_0 и B_1 на этом графе являются штрафами за все пути от вершин до конечной вершины W: $B_1(i,l)$ — суммарный штраф за пути от i до W, которые содержат $L-l+1$ значащих ребер и последним ребром которых является значащее ребро $\varepsilon(i,i+d(k_l),k_l)$; $B_0(i,l)$ — суммарный штраф за пути от i до W, которые содержат $L-l$ значащих ребер и последним ребром которых является значащее ребро $\varepsilon(i,i+1,\kappa)$.

В конце приведем эквивалентное рекурсивное определение величин F_0 та F_1, которое определяет эффективный алгоритм их вычисления.

Пусть i принимает все возможные значения координат столбиков изображения включительно с фиктивным столбиком с координатой W: $i = \overline{0, W}$, а l — номер буквы строки $\bar{k}$ включительно с фиктивной буквой „пустая строка“: $l = \overline{0, L}$. Тогда величины $F_0(i, l)$ и $F_1(i, l)$ определяются следующими рекурсивными соотношениями:

$$\begin{cases} F_0(0,0) = 1, \\ F_1(0,0) = 0, \\ F_0(i,l) = f(i-1,\kappa) \cdot (F_0(i-1,l) + F_1(i-1,l)), \quad l = \overline{0,L}, i = \overline{1,W}, \\ F_1(i,l) = f(i-d(k_l),k_l) \cdot (F_0(i-d(k_l),l-1) + F_1(i-d(k_l),l-1)), \quad l = \overline{1,L}, i = \overline{d(k_l),W}. \end{cases} \tag{19}$$

Очевидно, что введенные таким образом величины F_1 и F_0 совпадают с приведенными в (13) и (16).

Аналогично дадим рекурсивное определение величин $B_0(i, l)$ и $B_1(i, l)$:

$$\begin{cases} B_0(W,L) = 1, \\ B_1(W,L) = 0, \\ B_0(i,l) = f(i,\kappa) \cdot (B_0(i+1,l) + B_1(i+1,l)), \quad l = \overline{0,L},\ i = \overline{0,W-1}, \\ B_1(i,l) = f(i,k_l) \cdot (B_0(i+d(k_l),l+1) + B_1(i+d(k_l),l+1)), \quad l = \overline{0,L-1},\ i = \overline{0,W-d(k_l)}. \end{cases} \tag{20}$$

Вычисленные таким образом B_1 и B_0 совпадают с приведенными в (14) и (17).

Из соотношений (9), (12) и (15) следует окончательное выражение для значения апостериорной вероятности $\alpha(x, s)$ произвольного сегмента $s = (i, a)$:

$$\alpha(x,s) = \begin{cases} \frac{1}{Z} \sum\limits_{\substack{1 \le l \le L: \\ k_l = a}} \frac{F_1(i+d(a),l) \cdot B_1(i,l)}{f(i,a)}, & \text{если } a \in A_0; \\ \frac{1}{Z} \sum\limits_{l=0}^{L} \frac{F_0(i+1,l) \cdot B_0(i,l)}{f(i,a)}, & \text{если } a = \kappa, \end{cases} \tag{21}$$

где F_1 и F_0 вычислены в соответствии с (13) и (16), B_1 и B_0 — в соответствии с (14) и (17), а Z — в соответствии с (18).

2.4 Алгоритм обучения задачи распознавания

Теперь, когда указаны способы вычисления всех величин, которые необходимы для эффективной реализации алгоритма самообучения, ми сформулируем его еще раз, в целостном виде. Но сначала обратим внимание читателя на начальные значения эталонов символов E^0, априорных вероятностей сегментов $p^0(\bar{s}),\ \bar{s} \in \bar{S}$, которые устанавливаются на первом шаге работы алгоритма, а также критерий остановки алгоритма. Их выбор и влияние на работу алгоритма описаны в разделе, посвященном экспериментам.

Алгоритм 1 Обучение задачи распознавания

1: Выбираем начальные значения параметров $(E^0, p^0(\bar{s}))$.

2: Пользуясь полученными параметрами $(E^r, p^r(\bar{s}))$ как истинными, вычисляем величины F_0, F_1 и B_0, B_1 в соответствии с (19) и (20).

3: По F_0, F_1 и B_0, B_1 вычисляем $\alpha^r(x^m, s)$ в соответствии с (21).

4: В соответствии с (10) и (11), оцениваем новые значения параметров $(E^{r+1}, p^{r+1}(s))$.

5: Если выполняется критерий остановки, то конец. Иначе переходим на шаг 2.

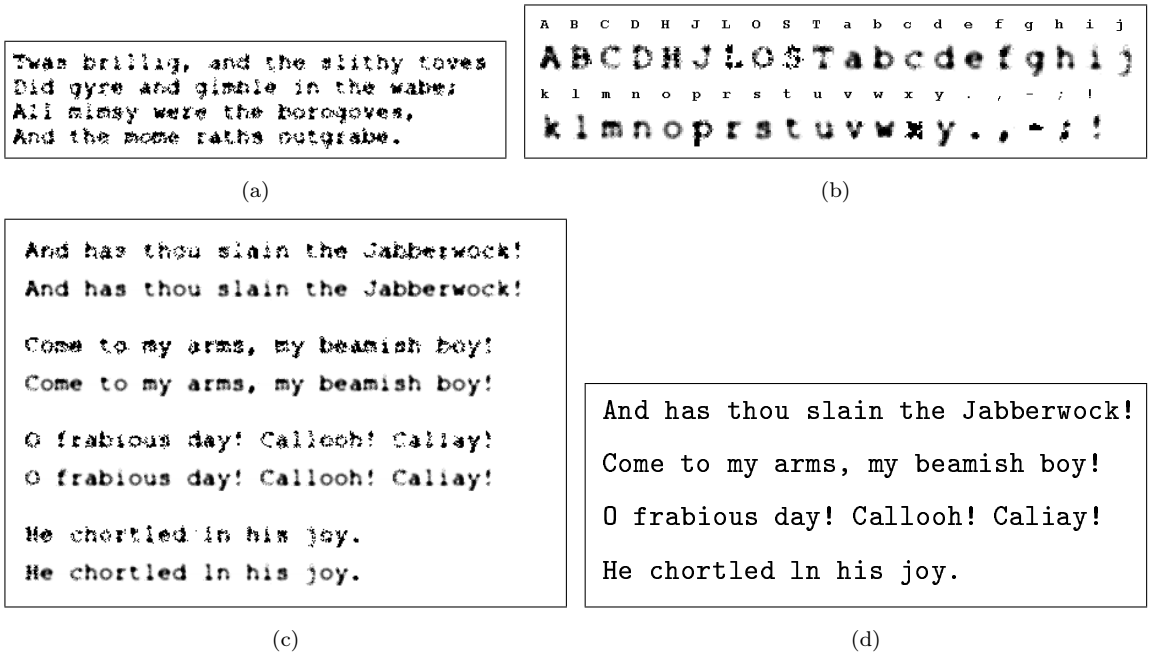
3 Экспериментальная проверка

3.1 Тестовые примеры

Эксперименты проводились следующим образом: на вход алгоритма подавалась выборка, которая состояла из изображений текстовых строк и соответствующих им текстовых строк, а также размеры эталонов всех букв алфавита и величина дисперсии σ^2 гауссовского распределения (1). Экспериментальные выборки были разбиты на две части: обучающую и тестовую. Первая часть использовалась для построения эталонов символов при помощи описанного в статье подхода, вторая же часть выборки использовалась для оценки качества распознавания с помощью построенных эталонов.

В этом подразделе мы рассмотрим четыре разных примера. Первый был создан искусственным путем, остальные являются натуральными с разными видами искажения изображений.

3.1.1 Пример „Jabberwocky“



(a) (b)

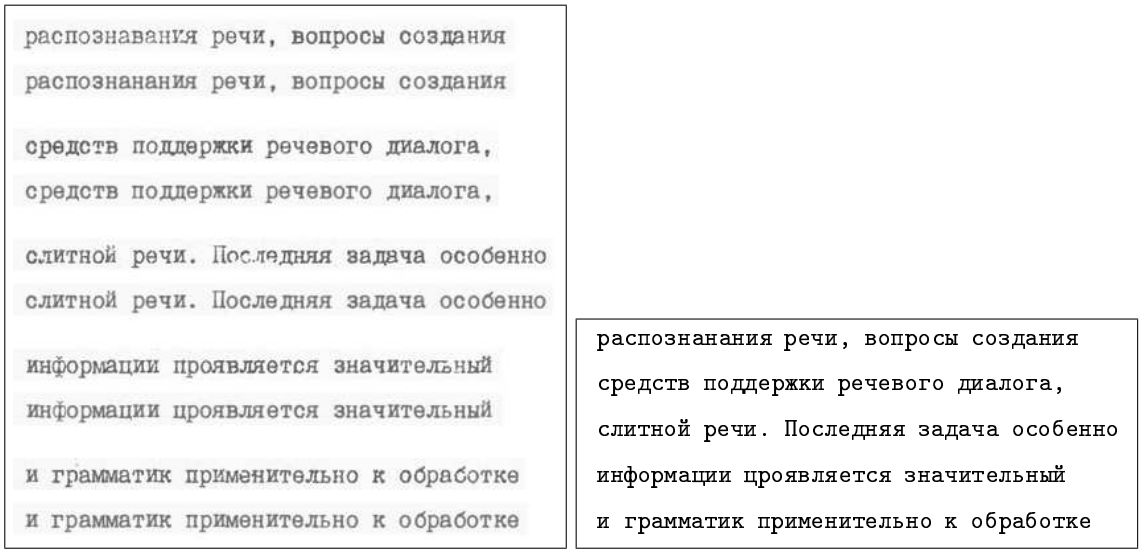
(c) (d)

Рис. 1: Пример „Jabberwocky“. (a) — пример входного изображения, (b) — построенные эталоны, (c), (d) — результаты распознавания

В этом примере изображения были созданы искусственным путем с применением модели деградации изображений, описанной в [2]. Эта модель отображает зашумление изображений черно-белого печатного текста после многочисленных операций печати и копирования. Такой вид зашумления относительно хорошо описывается статистической моделью, принятой в данной работе. На рис. 1 приведены входные изображения и результаты работы алгоритма. На образцах тестовых входных изображений (рис. 1(a)) можно оценить характер зашумления. Рис. 1(b) содержит изображения эталонов символов, полученных при обучении. На рис. 1(c) приведены пары изображений текстовых строк, которые демонстрируют результат распознавания текстовых изображений. В каждом паре первое изображение — это исходная строка тестовой выборки, а второе — изображение, склеенное из эталонов букв текстовой строки, которая является результатом распознавания этого изображения. В конце концов, на рис. 1(d) приведен распознанный алгоритмом текст.

Обучающая выборка имела объем порядка 600 букв. Ошибка при распознавании составила около 1%. Собственно, в результатах распознавания присутствуют две ошибки и в ооих случаях перепутаны буквы „l“ та „i“. Эталоны этих букв достаточно похожи, и шум легко делает более вероятной неправильную букву.

3.1.2 Пример „Book Intro“



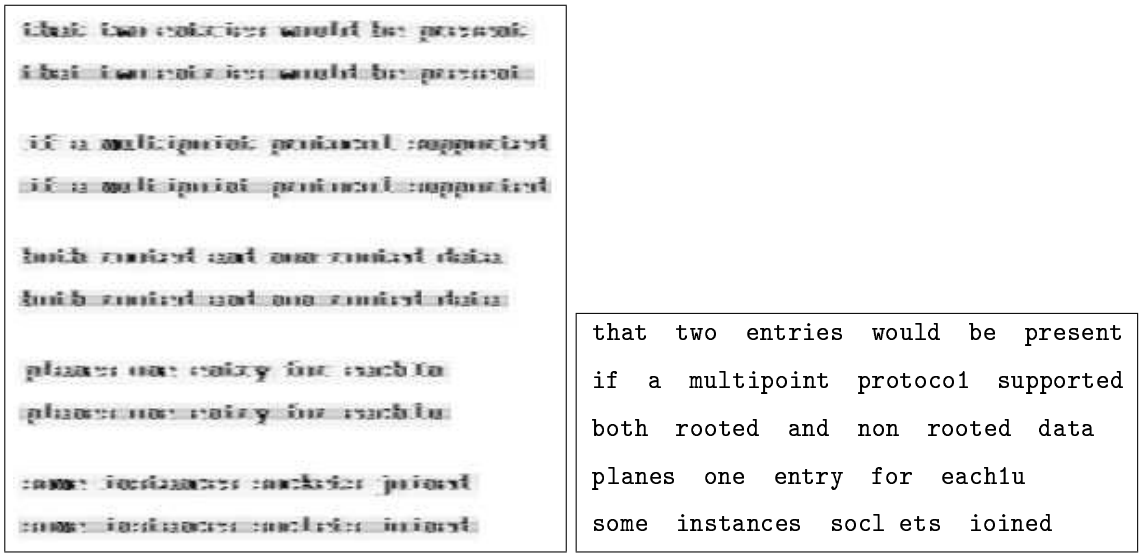
(a) (b)

Рис. 2: Пример „Book Intro“. Результаты распознавания

Второй пример содержит изображения строк отсканированной страницы книги. Результаты работы алгоритмов приведены на рис. 2 и в Табл. 1. Типичная ошибка в этом примере — перепутывание букв „п“, „н“, „и“, эталоны которых схожи друг с другом.

3.1.3 Пример „Gothic“

Следующий пример является реальным изображением с сильным искажением. Результаты распознавания приведены на рис. 3 и в Табл. 1. Ошибка при распознавании тестовой выборки соста-



(a)

that two entries would be present
if a multipoint protoco1 supported
both rooted and non rooted data
planes one entry for each1u
some instances socl ets ioined

(b)

Рис. 3: Пример „Gothic“. Результаты распознавания

вила 3%, причем около половины ошибок возникли из-за того, что необходимые буквы не были представлены в обучающей выборке.

3.1.4 Пример „BigBrother“

Последний пример также является реальным. Он был получен сканированием текста, распечатанного на лазерном принтере с дефектным барабаном. Как видно из рис. 4(a), характер зашумления изображения значительно отличается от принятой модели, в первую очередь неравномерностью насыщенности фона. Тем не менее, подбором дисперсии σ^2 удалось достичь распознавания тестовой выборки с уровнем ошибки 5%.

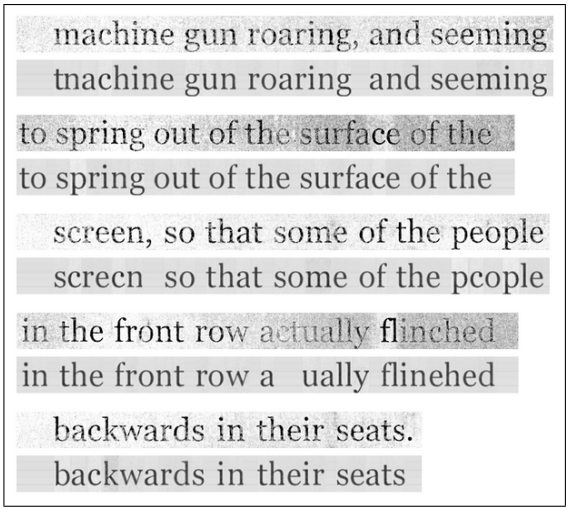
3.2 Начальные значения параметров алгоритма

На первом шаге работы алгоритма устанавливаются начальные значения эталонов символов и априорных вероятностей сегментов. Кроме того, перед началом работы необходимо задать значение дисперсии σ^2 гауссовского распределения (1).

В рассмотренных экспериментах начальные параметры $(E^0, p^0(\bar{s}))$ были выбраны следующими: одинаковые вероятности для всех сегментов, эталонные изображения черные для всех букв и белые для вставки. Такие эталоны являются естественными для темного текста на светлом фоне. Если сравнивать с случайно сгенерированными начальными значениями эталонов, черно-белые обеспечивают более быстрое схождение алгоритма и построение эталонов немного лучшего визуального качества.

Дисперсия σ^2 является характеристикой зашумленности изображений, которые подаются на вход алгоритма. Ее оптимальное значение зависит от входных изображений, но влияние на результаты работы несущественны.

(a)



(b)

```
tnachine gun roaring and seeming
to spring out of the surface of the
screcn so that some of the pcople
in the front row a  ually flinehed
backwards in their seats
```

(c)

Рис. 4: Пример „Big Brother“. (a) — входное изображение, (b), (c) — результаты распознавания

Эксперименты показали, что алгоритм очень чувствителен к изменению фиксированных ширин эталонов букв. Задание ширин, больших за истинные, может значительно ухудшить результаты обучения и распознавания. Причины этого кроются в постановке задачи распознавания, где запрещается наложение эталонов соседних букв.

3.3 Критерии остановки алгоритма обучения

Очевидным критерием остановки алгоритма является правильное распознавание всех изображений из обучающей выборки. Но формулировка задачи в виде минимизации риска не гарантирует, что такой критерий когда-либо будет выполнен.

Эксперименты показали, что алгоритм сходится довольно быстро — как правило, необходимо от 3 до 5 итераций для того, чтобы получить результат, который уже не улучшается, в том смысле, что функция правдоподобия мало изменяется. Тем не менее, полученный результат может оказаться локальным максимумом функции правдоподобия, а не глобальным, что отображается в неправильном построении нескольких эталонов и, соответственно, неправильном распознавании обучающей выборки. В таком случае алгоритм можно вытолкнуть из локального максимума реинициализацией неправильно построенных эталонов и априорных вероятностей сегментов.

На основании этих наблюдений был разработан эвристический алгоритм реинициализации параметров. На каждом его шаге алгоритм обучения проходит несколько итераций, после чего распознается обучающая выборка. Распознанные строки сравниваются с истинными из выборки, и эталоны букв, на которых алгоритм ошибся, объявляются „виновными“ и сбрасываются на начальные значения. После этого начинается новый шаг. Алгоритм работает до тех пор, пока все изображения из обучающей выборки не будут распознаваться правильно или же до достижения заданного количества итераций.

3.4 Применение алгоритма как вспомогательного для алгоритмов настройки параметров

Описанный подход может использоваться не только как самостоятельный метод статистического оценивания эталонов, но и как вспомогательный метод для других алгоритмов обучения, например алгоритмов настройки (см. [3, 4]).

Алгоритмы настройки не так сильно, как описанный нами, зависят от характера зашумленности изображений, поэтому при значительном отклонении характера зашумления от принятой в работе гауссовской модели обеспечивают лучшие результаты.

Недостатком алгоритмов настройки является требование наличия в обучающей выборке точной сегментации каждого из изображений. Как уже было неоднократно указано, построение таких сегментаций обычно требует значительных усилий и времени оператора.

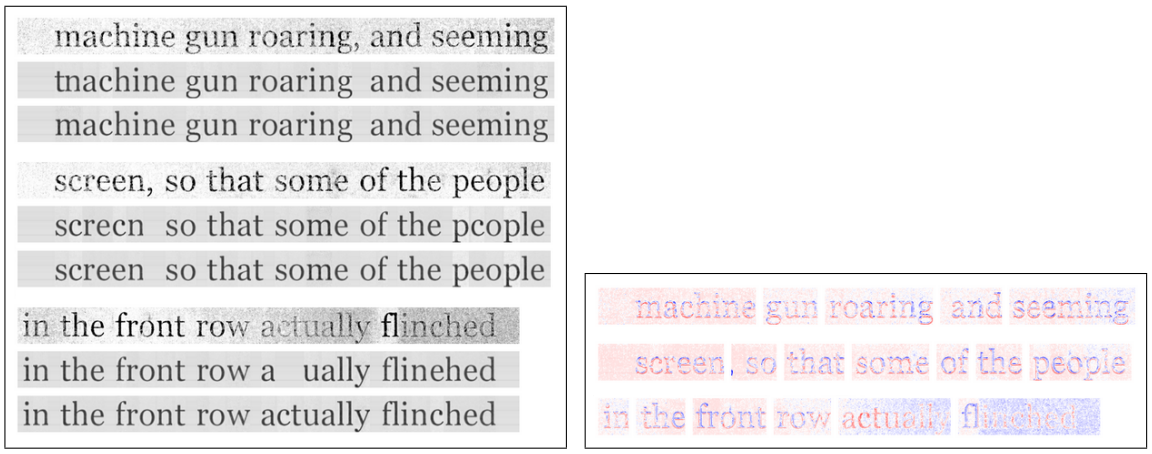
Вместе с тем на основе описанного нами подхода даже для сильнозашумленных изображений могут быть получены сегментации достаточной точности. Построение сегментации состоит из двух этапов: сначала на основе текстовой строки и входного изображения строятся эталоны букв так, как это описано в данной работе. На втором этапе на основе полученных эталонов находится наиболее вероятная сегментация среди тех, которые отвечают введенной на первом этапе текстовой строке. Эта сегментация и принимается в качестве входной для последующего алгоритма настройки.

Ниже приведен пример работы такого комбинированной настройки эталонов на тексте из примера „Big Brother“. На рис. 5(а) изображены результаты распознавания, где первая строка каждой группы — это входное изображение, вторая — результат обычного распознавания, третья — наиболее вероятная сегментация среди тех, которые отвечают текстовой строке из обучающей выборки.

Заключение

Подход к оцениванию эталонов букв для задачи распознавания текста, предложенный в работе, позволяет существенно уменьшить объем ручной работы при подготовке обучающей выборки, поскольку вместо точной сегментации обучающих изображений на отдельные буквы на вход алгоритма подаются лишь текстовые строки, которые отвечают этим изображением.

Перспективным направлением дальнейших исследований в этой области является алгоритмы обучения, в процессе работы которых автоматически настраиваются не только цвета (градации



(a) (b)

Рис. 5: Автоматическое построение точных сегментаций. Результаты обычного и модифицированного распознавания

серого) эталонных изображений букв, но и размеры этих изображений.

Список литературы

[1] М. Шлезингер and В. Главач. *Десять лекций по статистическому и структурному распознаванию*. Наукова думка, Киев, 2004.

[2] Prateek Sarkar, Henry S. Baird, and Xiaohu Zhang. Training on severely degraded text-line images. In *IAPR 7th International Conference on Document Analysis and Recognition (ICDAR03)*, pages 38–43, Edinburgh, Scotland, August 2003.

[3] Б. Д. Савчинський and О. В. Камоцький. Настройка алгоритму розпізнавання тексту. *Управляющие системы и машины*, (2):17–24, 2005.

[4] Bogdan Savchynskyy and Olexander Kamotsky. Character templates learning for textual images recognition as an example of learning in structural recognition. In *Second International Conference on Document Image Analysis for Libraries (DIAL'06)*, pages 88–95, Lyon, France, April 2006.