Настройка алгоритма распознавания текста

Савчинский Б.Д., Камоцкий А.В.

31 марта 2006 г.

Введение

Задача распознавания текста является, наверное, самой давней и популярной среди задач распознавания документов. Традиционно, начиная с работ Ковалевского [1, 2], задача распознавания строки текста формулируется как задача поиска наилучшей последовательности эталонов букв и решается с помощью методов динамического программирования.

Однако, сама по себе такая постановка задачи не даёт ответа на вопрос, каким образом строить эталоны букв. Традиционно [3, 4, 5] считается, что изображение текста искажается случайным шумом определённого типа. Этим самым задаётся статистическая модель генерирования изображения буквы по её эталону. На основе полученной модели и примеров изображений букв их эталоны строятся как оценки максимального правдоподобия.

Поскольку уже сама задача построения статистической модели изображений букв является довольно сложной, то обычно эта модель фиксируется волевым путём, исходя из некоторых априорных соображений.

В нашей работе мы предлагаем другой подход к задаче построения эталонов, который позволяет избежать оценивания их статистических моделей. Сформулированные и решённые нами задачи являются развитием задач настройки параметров автономного стохастического автомата, впервые рассмотреных в [6]. Для обозначния этого подхода, так же, как это сделано в [6], мы будем употреблять слово «настройка» в отличие от задач статистического оценивания, для которых принят термин «обучение».

Статья состоит из шести частей. В первой части мы формулируем основые определения и постановку задачи разпознавания строки текста. Во второй части представлена постановка задачи построения эталонов букв как задачи настройки. Третья и четвёртая часть посвящены её решению. Пятая часть содержит результаты экспериментов. Выводы к статье сформулированы в заключительной части.

1 Основные определения. Постановка задачи распознавания

Рассмотрим задачу распознавания одной строки текста по её изображению.

Назовём *полем зрения* T прямокутное подмножество двумерной целочисленной решётки. Её высоту и ширину будем обозначать соответственно H и W:

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

Элементами поля зрения T являются координаты пикселов изображения.

Пусть V – некоторое множество сигналов. *Изображением* x будем называть функцию вида $x : T \to V$. Яркость пиксела с координатами t на изображении x будем обозначать $x(t)$ или x_t. Множество всех возможных изображений обозначим V^T.

Пусть A – конечное множество, которое состоит из имён букв, а $\alpha \in A$ – это имя некоторой буквы. *Строку текста* будем понимать как последовательность имён α.

Будем считать, что изображения всех букв на поле зрения имеют одинаковую высоту H и ширину $d(\alpha)$, которая зависит лишь от имени буквы α. Таким образом, будем считать определённой *функцию ширины буквы* вида $d : A \to \mathbb{N}$.

Сегментом s будем называть пару (α, j), где α – имя буквы из A, а $j \in (1, 2, \ldots, W)$. Каждый сегмент $s = (\alpha, j)$ определяет фрагмент $T(s)$ поля зрения с именем $\alpha(s)$, который имеет высоту H, ширину $d(\alpha(s))$ и начинается со столбца номер $j(s)$:

$$T(s) = \left\{(i, j) \mid i \in \overline{1, H}, j \in \overline{j(s), j(s) + d(\alpha(s)) - 1}\right\}.$$

Таким образом, сегменты — это поименованные фрагменты поля зрения. Множество всех сегментов будем обозначать S.

Всякой строке текста $(\alpha_1, \ldots, \alpha_L)$ длины L, изображённой на x, однозначно соответствует последовательность сегментов $(s_1, \ldots, s_L)$, которые вплотную примыкают друг к другу и все вместе полностью покрывают поле зрения T:

$$\left\{\begin{array}{rcll} \alpha(s_l) & = & \alpha_l & \forall l \in \overline{1, L} \\ j(s_1) & = & 1 & \\ j(s_{l+1}) & = & j(s_l) + d(\alpha(s_l)) & \forall l \in \overline{1, L-1} \\ j(s_L) + d(\alpha(s_L)) & = & W + 1 & \end{array}\right. \tag{1}$$

Последовательность $\overline{s} = (s_1, \ldots, s_L)$, удовлетворяющую условиям (1), назовём *сегментацией*. Её длину будем обозначать $L(\overline{s})$. Множество всех возможных сегментаций на поле зрения будем обозначать $\overline{S}$.

Введём множество $E \subset \mathbb{R}^K$ векторов параметров. Каждый вектор $\overline{e} \in E$ определяет эталонный вид изображений всех букв $\alpha \in A$. Для определения *схожести* изображения x на участке, определяемом сегментом s, на букву $\alpha(s)$, введём *локальную функцию отличия* $f : E \times S \times V^T \to \mathbb{R}$. Чем большая схожесть, тем меньшим должно быть её значение, которое будем обозначать $f_{\overline{e}}(s, x)$.

Меру *схожести изображения* x и строки $(\alpha_1, \ldots, \alpha_L)$, которой соответствует сегментация $\overline{s} = (s_1, \ldots, s_L)$, мы будем определять как суммарную схожесть каждого сегмента $\overline{s}$, то есть как $\sum_{l=1}^{L(\overline{s})} f_{\overline{e}}(s_l, x)$.

Задача 1 *распознавания изображения x строки текста состоит в нахождении сегментации $\overline{s}^* = (s_1^*, s_2^*, \ldots, s_L^*)$, на которую оно похоже больше всего:*

$$\overline{s}^* = \arg\min_{\overline{s} \in \overline{S}} \sum_{l=1}^{L(\overline{s})} f_{\overline{e}}(s_l, x) \tag{2}$$

Строго говоря, функция $\sum_{l=1}^{L(\overline{s})} f_{\overline{e}}(s_l, x)$ может принимать минимальное значение на некотором множестве сегментаций $\overline{S}^*$. В таком случае будем считать, что $\overline{s}^*$ обозначает любой элемент этого множества. Ситуацию, когда решение (2) обязательно должно быть единственным, мы будем обозначать знаком равенства $\overset{!}{=}$.

Задача 1 решается методом динамического программирования, что впервые было сделано в работе Ковалевского [1]. Формальный подход к распознаванию изображений, которые рассматриваются как последовательности более простых, меньших своих частей, получил название «метод еталонных последовательностей» [2].

В дальнейшем будем полагать, что у нас имеется готовая процедура — *алгоритм распознавания* $\overline{s}^*_P(x, \overline{e})$ для нахождения сегментации $\overline{s}^*$ путём решения задачи 1 для заданной тройки $P = \langle A, d, f_{\overline{e}} \rangle$ — множества имён букв A, функции ширины буквы d и локальной функции отличия $f_{\overline{e}}$, определённой с точностью до вектора параметров $\overline{e}$. Реализация $\overline{s}^*_P(x, \overline{e})$ нас не будет интересовать, однако далее мы будем рассчитывать, что она позволяет отслеживать случаи, когда решение $\overline{s}^*$ не единственно.

На практике P фиксируется, исходя из предположений о природе входного изображения. Например, в машинописном тексте буквы имеют фиксированную ширину, множество их имён A определяется доступными печатными символами, а вектор параметров $\overline{e}$ может содержать эталонные, чётко напечатанные изображения букв.

Обычно модель входного изображения предусматривает, что на идеальное изображение $e(\alpha)$ буквы α действует случайный аддитивный шум r со сферически-симметричной функцией распределения $p_{\text{ш}}(r)$:

$$x = e(\alpha) + r. \tag{3}$$

Здесь x – зашумленное изображение буквы. В таком случае можно применить корреляционный метод распознавания [3, 2], а в качестве локальной функции отличия $f_{\overline{e}}$ взять функцию правдоподобия. При предположении, что плотность вероятности $p_{\text{ш}}(r)$ шума является некоторой монотонно убывающей функцией $f(\cdot)$ от суммы квадратов его компонент

$$p_{\text{ш}}(r) = f\left(r^2\right), \tag{4}$$

функция $f_{\overline{e}}$ принимает вид эвклидового расстояния между векторами $x = \{x_{i,j} \mid (i,j) \in T(s)\}$ и $e(\alpha) = \left\{e^{\alpha}_{i,j} \mid (i,j) \in T(s)\right\}$:

$$f_{\overline{e}}(s, x) = \sum_{\substack{i \in \overline{1,H} \\ j = \overline{1, d(\alpha(s))}}} \left(e^{\alpha(s)}_{i,j} - x_{i, j(s)+j}\right)^2. \tag{5}$$

2 Выбор вектора параметров. Задача настройки

Настройка алгоритма распознавания $\overline{s}^*_P(x, \overline{e})$, решающего задачу 1, состоит в выборе значения вектора параметров $\overline{e}$ для заданной тройки $P = \langle A, d, f_{\overline{e}} \rangle$.

Пусть задано множество тестовых изображений $X^o = \{x^o_r \mid r \in \overline{1, R}\}$ и множество соответствующих им сегментаций $\overline{S}^o = \{\overline{s}^o_r \mid r \in \overline{1, R}\}$.

Задача 2 *настройки алгоритма распознавания состоит в нахождении такого вектора параметров $\overline{e}^*$, чтобы результатом распознавания каждого изображения-образца x^o_r была соответствующая тестовая сегментация $\overline{s}^o_r$ и только она:*

$$\overline{s}^o_r \stackrel{!}{=} \arg\min_{\overline{s} \in \overline{S}} \sum_{l=1}^{L(\overline{s})} f_{\overline{e}^*}(s_l, x^o_r) \qquad \forall r \in \overline{1, R}.$$

Сложность задачи состоит в том, что минимум ищется на множестве сегментаций $\overline{S}$, мощность которого пропорциональна количеству *всех возможных текстов*, которые возможно изобразить на поле зрения.

3 Решение задачи настройки

Мы предлагаем решение задачи настройки 2 для случая, когда функция отличия является линейной по параметрам $\overline{e} = \{e_1, \ldots, e_K\}$, то есть может быть представлена в виде:

$$f_{\overline{e}}(s, x) = \sum_{k=1}^{K} e_k \cdot \varphi_k(s, x) = \langle \overline{e}, \overline{\varphi}(s, x) \rangle . \tag{6}$$

Здесь $\overline{\varphi}(s, x) = \{\varphi_1(s, x), \ldots, \varphi_K(s, x)\}$ – некоторый вектор, который непосредственно зависит от изображения x и сегмента s.

Отметим, что вид функции отличия (6) включает широко используемый случай (5), когда компоненты $\overline{e}$ принимают значения из множества сигналов пикселов изображения V, а сам вектор параметров непосредственно содержит идеальные изображения всех букв $\alpha \in A$. Однако, в общем случае, вектор $\overline{e}$ может иметь более сложный и не визуальный смысл.

Обозначим множество пар тестовых изображений и соответствующих им сегментаций как Π:

$$\Pi = \left\{ (x_r^o, \overline{s}_r^o) \mid r \in \overline{1, R} \right\} .$$

Теперь задача настройки 2 принимает следующий вид:

найти $\overline{e}^*$ такой, чтобы:

$$\overline{s}^o \overset{!}{=} \arg\min_{\overline{s} \in \overline{S}} \sum_{l=1}^{L(\overline{s})} \sum_{k=1}^{K} e_k^* \varphi_k(s_l, x^o) \qquad \forall (x^o, \overline{s}^o) \in \Pi. \tag{7}$$

Равенство (7) означает, что для каждой пары $(x^o, \overline{s}^o) \in \Pi$ должна выполнятся система неравенств:

$$\left\{ \sum_{l=1}^{L(\overline{s})} \sum_{k=1}^{K} e_k^* \varphi_k(s_l^o, x^o) < \sum_{l=1}^{L(\overline{s})} \sum_{k=1}^{K} e_k^* \varphi_k(s_l, x^o) \qquad \forall \overline{s} \neq \overline{s}^o, \right.$$

что эквивалентно

$$\left\{ \sum_{l=1}^{L(\overline{s})} \sum_{k=1}^{K} e_k^* \varphi_k(s_l, x^o) - \sum_{l=1}^{L(\overline{s})} \sum_{k=1}^{K} e_k^* \varphi_k(s_l^o, x^o) > 0 \qquad \forall \overline{s} \neq \overline{s}^o. \right. \tag{8}$$

Перепишем (8), сгруппировав компоненты вектора параметров $\overline{e}^*$ и обозначив все члены при e_k^* как $\varphi'_k(\overline{s}, \overline{s}^o, x^o)$:

$$\left\{ \sum_{k=1}^{K} e_k^* \varphi'_k(\overline{s}, \overline{s}^o, x^o) > 0 \qquad \forall \overline{s} \neq \overline{s}^o. \right. \tag{9}$$

Таким образом, задача настройки 2 состоит в решении (9) относительно вектора $\overline{e}^*$.

Поскольку линейная система (9) содержит $\left|\overline{S}\right| - 1$ неравенств, для её решения невозможно использовать классические методы линейной оптимизации, так как их сложность непосредственно зависит от количества неравенств. Однако, решение (9) можно найти с помощью методов разделения множеств точек гиперплоскостью, сложность которых не зависит от мощности системы, а зависит от других её свойств.

В нашей задаче распознавания строки текста мы использовали алгоритмы Козинца и персептрона. Их особенность в том, что поиск решения системы (9) происходит путём итеративного корректирования значения вектора параметров на основе любого одного невыполняющегося неравенства. Поэтому для своей работы они нуждаются лишь в эффективном методе поиска такого неравенства. Сходимость этих алгоритмов за конечное число шагов определяют соответствующие теоремы, описанные в [6].

В нашем случае, невыполнение любого неравенства в системе (9), для текущего значения $\overline{e}$, означает, что для некоторой тестовой пары $(x^o, \overline{s}^o)$ не выполняется равенство (7), то есть, определённое тестовое изображение x^o распознаётся неправильно. Следовательно, для проверки выполнения (9) необходимо убедиться в выполнении (7), то есть решить задачу разпознавания 1 для всех $x^o \in X^o$.

Таким образом, алгоритмы Козинца и персептрона итеративно корректируют вектор параметров $\overline{e}$, оперируя при этом процедурой $\overline{s}^*_P(x, \overline{e})$. При заданном $P = \langle A, d, f_{\overline{e}} \rangle$, она находит решение задачи распознавания 1, которое определяет то неравенство системы (9), которое не выполняется для текущего значения $\overline{e}$.

Алгоритм 1 Алгоритм Козинца для нахождения решения (9)

1: Выбираем произвольный вектор $\overline{e} \neq 0$ из выпуклой оболочки множества векторов $\{\overline{\varphi}'\}$. Например, берём его равным любому $\overline{\varphi}'(\overline{s}, \overline{s}^o_r, x^o_r) \neq 0$, $r \in \overline{1, R}$, $\overline{s} = \overline{s}^*_P(x^o_r, 0)$.

2: Ищем неравенство системы, которое не выполняется для текущего значения $\overline{e}$:

$$\text{найти } r \in \overline{1, R}, \text{ для которого } \overline{s}^o_r \overset{!}{\neq} \overline{s}^*_P(x^o_r, \overline{e}) .$$

3: Если такого r нет, то **конец**. Вектор $\overline{e}^*$ найден.

4: Вычислить новое значение вектора параметров как перпендикуляр, опущенный на отрезок, соединяющий вектора $\overline{e}$ и $\overline{\varphi}' = \overline{\varphi}'(\overline{s}^*_P(x^o_r, \overline{e}), \overline{s}^o_r, x^o_r)$:

$$\overline{e} := k \cdot \overline{e} + (1 - k) \cdot \overline{\varphi}', \tag{10}$$

$$\text{где } k = \frac{(\overline{\varphi}', \overline{\varphi}')}{(\overline{e}, \overline{e}) + 2(\overline{e}, \overline{\varphi}') + (\overline{\varphi}', \overline{\varphi}')}. \tag{11}$$

Перейти на шаг 2.

Алгоритм персептрона очень похож на алгоритм Козинца, но выглядит более простым. Применительно к решению системы (9), он имеет следующий вид:

Алгоритм 2 Алгоритм персептрона для нахождения решения (9)

1: Положить $\overline{e} = 0$.

2: Ищем неравенство системы, которое не выполняется для текущего значения $\overline{e}$:

$$\text{найти } r \in \overline{1, R}, \text{ для которого } \overline{s}^o_r \overset{!}{\neq} \overline{s}^*_P(x^o_r, \overline{e}) .$$

3: Если такого r нет, то **конец**. Вектор $\overline{e}^*$ найден.

4: Вычислить новое значение вектора параметров $\overline{e}$ по формуле:

$$\overline{e} := \overline{e} + \overline{\varphi}'(\overline{s}^*_P(x^o_r, \overline{e}), \overline{s}^o_r, x^o_r)$$

и перейти на шаг 2.

Для того, чтобы считать задачу настройки алгоритма распознавания текста полностью сформулированной и решённой, нам осталось конкретизировать лишь локальную функцию отличия $f_{\overline{e}}$. Её вид будет определять способ вычисления вектора $\overline{\varphi}'\left(\overline{s}, \overline{s}^o, x\right)$, которым оперируют алгоритмы 1 и 2.

4 Выбор локальной функции отличия

Как было показано Ковалевским [2], предположения (3) и (4) о модели изображения не слишком сужают область практического применения — алгоритм распознавания, который хорошо работает в условиях сферически-симметричного шума, будет приемлимо работать при любом шуме с независимыми компонентами и не слишком большой дисперсией.

Так как локальная функция отличия (5) не является линейной за параметром $\overline{e}$, то есть не может быть представлена в виде (6), мы расширили её до следующего вида:

$$f_{\overline{e}}\left(s, x\right) = \sum_{\substack{i \in \overline{1,H} \\ j = \overline{1, d(\alpha(s))}}} \ddot{e}^{\alpha}_{i,j} \cdot x^2_{i,j(s)+j} + \dot{e}^{\alpha}_{i,j} \cdot x_{i,j(s)+j} + e^{\alpha}_{i,j} \tag{12}$$

При этом вектор параметров формируется из всех коефициентов e:

$$\overline{e} = \left\{ \ddot{e}^{\alpha}_{i,j}, \dot{e}^{\alpha}_{i,j}, e^{\alpha}_{i,j} \mid \alpha \in A, i \in \overline{1, H}, j = \overline{1, d\left(\alpha\right)} \right\},$$

а вектор $\overline{\varphi}'\left(\overline{s}, \overline{s}^o, x\right)$ состоит из соответствующих компонент:

$$\overline{\varphi}'\left(\overline{s}, \overline{s}^o, x\right) = \left\{ \ddot{\varphi}'^{\alpha}_{i,j}, \dot{\varphi}'^{\alpha}_{i,j}, \varphi'^{\alpha}_{i,j} \mid \alpha \in A, i \in \overline{1, H}, j = \overline{1, d\left(\alpha\right)} \right\}, \tag{13}$$

которые вычисляются следующим образом:

$$\begin{aligned}
\ddot{\varphi}'^{\alpha}_{i,j} &= \sum_{\substack{s \in \overline{s} \\ \alpha(s) = \alpha}} x^2_{i,j(s)+j} - \sum_{\substack{s \in \overline{s}^o \\ \alpha(s) = \alpha}} x^{o2}_{i,j(s)+j}, \\
\dot{\varphi}'^{\alpha}_{i,j} &= \sum_{\substack{s \in \overline{s} \\ \alpha(s) = \alpha}} x_{i,j(s)+j} - \sum_{\substack{s \in \overline{s}^o \\ \alpha(s) = \alpha}} x^{o}_{i,j(s)+j}, \\
\varphi'^{\alpha}_{i,j} &= \sum_{\substack{s \in \overline{s} \\ \alpha(s) = \alpha}} 1 - \sum_{\substack{s \in \overline{s}^o \\ \alpha(s) = \alpha}} 1.
\end{aligned}$$

Эксперименты с алгоритмами 1 и 2 настройки вектора параметров $\overline{e}$ локальной функции отличия (12) показали, что на реальных изображениях настройка происходит за время, слишком большое для практического использования этих алгоритмов. Значительное число итераций, необходимых алгоритмам Козинца и персептрона, можна объяснить следующими эмпирическими рассуждениями. Различные компоненты вектора параметров $\overline{e}$ в (12) неравномерно влияют на значения локальной функции отличия $f_{\overline{e}}\left(s, x\right)$. Вследствие этого, каждое изменение значения коэффициента $\ddot{e}^{\alpha}_{i,j}$ требует адекватной подстройки $\dot{e}^{\alpha}_{i,j}$, а изменение $\dot{e}^{\alpha}_{i,j}$ аналогичным образом приводит к необходимости дополнительной настройки $e^{\alpha}_{i,j}$.

Одинаковое влияние различных компонент вектора параметров $\overline{e}$ на значения $f_{\overline{e}}\left(s, x\right)$ обеспечивается ортонормированным базисом, в котором может быть записана функция $f_{\overline{e}}$:

$$f_{\overline{e}}(s,x) = \sum_{\substack{i\in\overline{1,H} \\ j=\overline{1,d(\alpha(s))}}} \ddot{e}^{\alpha}_{i,j}\cdot\psi_2\left(x_{i,j(s)+j}\right) + \dot{e}^{\alpha}_{i,j}\cdot\psi_1\left(x_{i,j(s)+j}\right) + e^{\alpha}_{i,j}\cdot\psi_0\left(x_{i,j(s)+j}\right), \qquad (14)$$

где функции $\psi_i(x)$ – ортонормированные полиномы Чебышева. Каждая $\psi_i(x)$ является полиномом i-го порядка от сигнала x. При этом они связаны между собой следующим ограничением:

$$\int\limits_{x\in V} \psi_i(x)\cdot\psi_j(x)\,dx = \begin{cases} 1, & i=j; \\ 0, & \text{иначе.} \end{cases}$$

Компоненты (13) вектора $\overline{\varphi}'(\overline{s},\overline{s}^o,x)$ в этом базисе принимают такой вид:

$$\begin{aligned}
\ddot{\varphi}'^{\alpha}_{i,j} &= \sum_{\substack{s\in\overline{s} \\ \alpha(s)=\alpha}} \psi_2\left(x_{i,j(s)+j}\right) - \sum_{\substack{s\in\overline{s}^o \\ \alpha(s)=\alpha}} \psi_2\left(x^o_{i,j(s)+j}\right), \\
\dot{\varphi}'^{\alpha}_{i,j} &= \sum_{\substack{s\in\overline{s} \\ \alpha(s)=\alpha}} \psi_1\left(x_{i,j(s)+j}\right) - \sum_{\substack{s\in\overline{s}^o \\ \alpha(s)=\alpha}} \psi_1\left(x^o_{i,j(s)+j}\right), \\
\varphi'^{\alpha}_{i,j} &= \sum_{\substack{s\in\overline{s} \\ \alpha(s)=\alpha}} \psi_0\left(x_{i,j(s)+j}\right) - \sum_{\substack{s\in\overline{s}^o \\ \alpha(s)=\alpha}} \psi_0\left(x^o_{i,j(s)+j}\right).
\end{aligned}$$

При использовании локальной функции отличия (14), время работы алгоритмов 1 и 2 при настройке на реальных изображениях уменьшилось в три-пять раз.

5 Результаты експериментов

Мы проверили работу алгоритмов настройки 1 и 2 на изображениях двух типов: собственноручно сгенерированных и реальных, которые возникли в определённых прикладных обстоятельствах.

С помощью искусственных изображений мы смоделировали ситуацию, когда эталонные изображения букв недоступны. В этом случае при неизвестном характере шума можно использовать допущения (3) и (4) для постановки задачи оценивания эталонов букв на основе примеров их изображений. Её решение методом максимального правдоподобия сводится к поиску усреднённых шаблонов букв, доступных в обучающей выборке изображений текстовых строк. Использование полученных эталонов в качестве вектора параметров алгоритма распознавания с локальной функцией отличия (5) мы в дальнейшем будем называть *«подход с усреднением шаблонов букв»*.

Для локальной функции отличия вида (12) или (14), компоненты вектора параметров $\overline{e}$ не могут рассматриваться как эталонные изображения букв. Поэтому его поиск предлагается осуществлять с помощью решения задачи настройки 2.

5.1 Примеры, когда усреднение шаблонов букв не работает

В случае, когда допущения подхода с усреднением шаблонов букв оказываются неправильными, полученное значение вектора параметров $\overline{e}$ не даёт правильного распознавания даже тех обучающих изображений, которые использовались для усреднения. В то же время, алгоритм распознавания, полученный в результате решения задачи настройки 2, лишён этого недостатка.

Рисунки 1 и 2 демонстрируют описанную ситуацию. Для построения эталонов букв использовалось обучающее изображение 1, которое содержит один пример буквы «С» и семь образцов буквы «Е», последний из которых был искажён. После усреднения искажённое изображение буквы «Е» оказывается «более похожим» на эталон буквы «С» и поэтому на изображенении 2 распознаётся неправильно. На обучающем изображении эта ошибка также имеет место — оно распознаётся как «СЕЕЕЕЕЕС».

Рисунок 1: Изображение для настройки усреднением шаблонов букв «С» и «Е»

Рисунок 2: Искажённое изображение буквы «Е» распознаётся неправильно

При использовании предложенных алгоритмов настройки и локальных функций отличия, изображения 1 и 2 распознаются правильно.

Количественно, но не качественно более сложный пример показан на рисунках 3 и 4. Усреднение снова приводит к неправильному распознаванию, а предложенный подход даёт безошибочный результат.

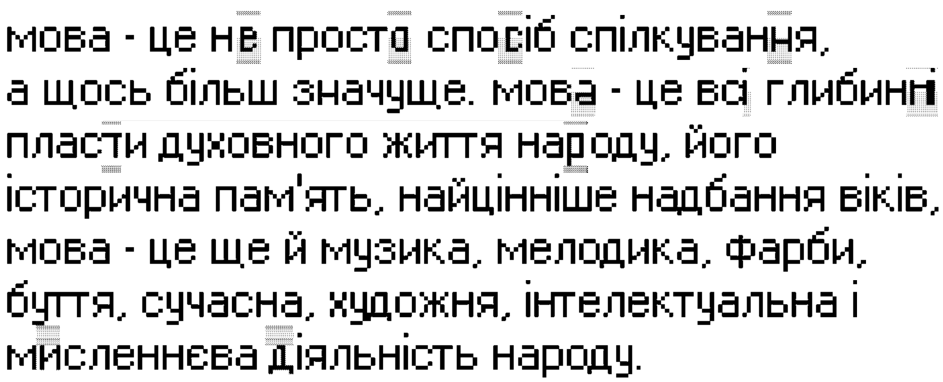


Рисунок 3: Изображение для настройки. Искажённые изображения букв обозначены серым

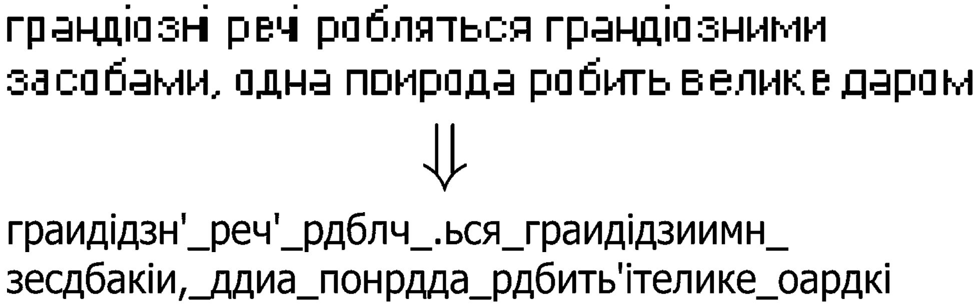


Рисунок 4: Неправильное распознавание на основе шаблонов, усреднённых по изображению 3

5.2 Настройка на реальных изображениях

Изображение 5 использовалось для настройки параметров локальной функции отличия вида (14). Полученный вектор оптимальных параметров использовался для распознавания другого изображения, результат которого представлен на рисунке 6. Настройка происходила с помощью алгоритма Козинца. Для вектора параметров, найденного с помощью алгоритма персептрона, результат распознавания изображения 6 аналогичный. Как видим, он содержит некоторые ошибки. Их можно объяснить тем, что некоторые буквы на обучающем изображении 5 встречались лишь один-два раза или же вообще отсутствовали.

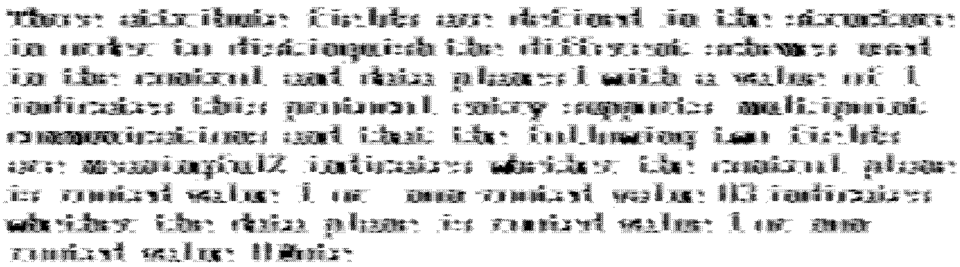
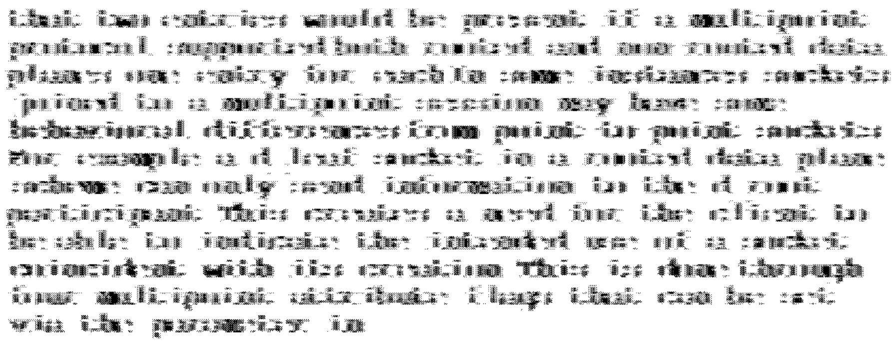


Рисунок 5: Пример реального изображения для настройки



$$\Downarrow$$

that_two_entries_wodd_he_present_if_a_multipoint
protocol_supported_hoth_rooted_and_nonerooted_data
planes_one_entry_for_each1n_some_instances_socbets
ioined_to_a_multtipoint_session_mup_h_oe_some
heharioral_differences_from_point_so_point_socaets
ror_enample_a_d_leaf_socaet_in_a_rooted_data_plane
scheme_can_only_send_information_to_the_d_root
participont_1his_creates_a_need_for_the_client_to
he_ahle_to_indicate_the_intended_use_of_a_socaet
coiucident_with_its_creation_This_is_done_through_four
multipoint_attrihute_flags_that_can_he_set
via_the_parameter_in

Рисунок 6: Пример распознавания реального изображения

5.3 Сравнение времени работы алгоритмов настройки

В своём распоряжении мы имели реальные изображения трёх уровней зашумленности, что позволило нам сравнить время работы алгоритмов настройки Козинца и персептрона при использовании локальной функции отличия вида (14) и (12).

Результаты представлены в таблице 1.

Изображение	вид $f_{\overline{e}}$	Козинец		персептрон	
		итерации	время	итерации	время
cGood	(14)	845	00:08:04	688	00:05:54
	(12)	3221	00:31:30	4897	00:42:10
cBad	(14)	3335	00:55:30	4096	01:08:00
	(12)	18778	03:02:10	34417	05:02:00
cVeryBad	(14)	34078	09:30:00	59396	16:30:00

Таблица 1: Время работы алгоритмов настройки Козинца и персептрона на реальных изображениях различного уровня зашумленности

Выяснилось, что время работы алгоритмов настройки увеличивается с возрастанием уровня зашумленности обучающего изображения-образца. Алгоритм Козинца в основном оказался быстрее алгоритма персептрона — только на изображении лучшего качества более простой алгоритм персептрона закончил работу первым. С ухудшением качества изображений выигрыш во времени алгоритма Козинца возрастает. На наихудших изображениях алгоритм Козинца оказался почти вдвое быстрее алгоритма персептрона.

Выводы

Результаты эксперименто свидетельствуют, что предложенный подход настройки алгоритмов распознавания может быть использован для построения систем распознавания текста, особенно когда входные изображения содержат шум с неопределёнными характеристиками.

Изложенный подход позволяет создавать системы распознавания, в которых параметры не являются фиксированными, а корректируются в процессе эксплуатации системы. Результаты коррекции ошибок распознавания, выполненной пользователем или автоматически, можно подавать в качестве входных данных для алгоритма настройки и, таким образом, уменьшать количество подобных ошибок в дальнейшем. Выделение вектора параметров $\overline{e}$ позволяет создавать наборы его оптимальных значений, специализированные для изображений с искажениями разного типа.

Использование предложенных алгоритмов настройки требует тщательного выбора базиса локальной функции отличия. В нашем случае, переход к ортонормированным полиномам Чебышева определяющим образом повлиял на возможность работы с изображениями реальных текстов.

Список литературы

[1] Ковалевский В.А. Оптимальный алгоритм распознавания некоторых последовательностей изображений. *Кибернетика*, (4), 1967.

[2] Ковалевский В.А. *Методы оптимальных решений в распознавании изображений.* Наука, Москва, 1976.

[3] Ковалевский В.А. Корреляционный метод распознавания изображений. *Журнал вычислительной математики и математической физики*, 2(4), 1962.

[4] G.E. Kopec and Lomelin M. Document-specific character template estimation. In *IS&T/SPIE 1996 Intl. Symposium on Electronic Imaging: Science & Technology*, San Jose, CA, Jan. 27–Feb. 2, 1996.

[5] Philip A. Chou, Gary E. Kopec. A stochastic attribute grammar model of document production and its use in document image decoding. In H. Baird L. Vincent, editor, *Document Recognition II*, volume 2422 of *SPIE Proc.*, pages 66–73, 1995.

[6] Michail I. Schlesinger, Vaclav Hlavač. *Ten lectures on statistical and structural pattern recognition.* Kluwer Academic Publishers, Dordrecht/Boston/London, 2002.