UNCONSTRAINED HANDWRITTEN TEXT RECOGNITION SYSTEM

E. Dolgova, D. Kurushin

Perm State Technical University
29, Komsomolsky Avenue, Perm, Russia

Abstract. The paper proposes an approach to adaptation of neural network models for creating OCR-systems, designed to work with uncjnstrained handwriting. The approach is based on the rejection of recognition of continuous characters and the transition to the recognition of individual strokes, which are then going to the characters and / or words of text. This approach can significantly reduce the dimension of neural networks used in the OCR-systems that will enhance their productivity and quality of recognition.
Key words: neural network, mathematical model, python, recognition, vector, count, language model.

004.932.1

ИССЛЕДОВАНИЕ МАТЕМАТИЧЕСКОЙ МОДЕЛИ ШТРИХОВ РУКОПИСНОГО ТЕКСТА

Е.В. Долгова, Д.С. Курушин

Пермский государственный технический университет
614990, Пермь, Комсомольский пр., 29

Аннотация: В статье предлагается и исследуется модель штрихов рукописного текста. Рассматриваемая модель представляет рукописный текст как совокупность штрихов нескольких известных классов. Модель позволяет абстрагироваться от начальных условий письма, таких как угол наклона, скорость, плотность и т.п. Благодаря процедуре векторизации для модели не имеет значения зашумленность исходного изображения и способ его получения. В работе проведен эксперимент, подтверждающий применимость модели для использования в задачах распознавания образов.
Ключевые слова: нейронная сеть, математическая модель, python, распознавание, вектор, граф, языковая модель, распознавание, штрих.

Широкое распространение и увеличение доступности технологий сканирования и цифрового фотографирования привело к быстрому росту цифровых коллекций документов. В таких коллекциях документы хранятся в виде растровых графических файлов. Оцифровка решает множество проблем, связанных с сохранением и организацией доступа к документам. Однако для реализации полнотекстового поиска, изучения содержания, подготовки публикаций требуется перевод из графического формата в текстовый, то есть распознавание текста.

Алгоритмы и программы автоматического распознавания текста разрабатываются уже несколько десятилетий. Можно сказать, что задача распознавания текстов на европейских языках, напечатанных на принтерах, решена.

Сложности многократно увеличиваются при попытке решения задачи распознавания рукописного текста. Введение в электронное использование рукописных исторических документов, хранящихся в архивах и библиотеках России, имеет огромное научное и культурное значение, так как каждая рукопись уникальна.

Поэтому весьма актуальной выглядит задача создания достаточно универсальной программной системы для автоматизированного распознавания рукописного текста.

В настоящее время многие исследователи, (напр. см. [4]) полагают, что распознавание рукописного текста может выполняться по следующему алгоритму:

1. Выделение слов или словосочетаний (основываясь на промежутках между словами);
2. Сегментация текста на элементы (символы, штрихи…);
3. Распознавание элементов;
4. Генерация выходного текста.

В данной работе мы рассмотрим один из возможных подходов к распознаванию штрихов, составляющих рукописный текст.

Распознавание штрихов, представленных наборами точек (в растровом формате) может выполняться любым традиционным способом, например могут использоваться нейронные сети различных конфигураций (см. напр. [2]). Однако, использование нейронных сетей в таком контексте приводит или к необходимости нормализовывать размеры изображения. При нормализации изображения к размерности входного вектора сети неизбежно возникают нелинейные искажения, что ухудшает качество распознавания, или требует увеличения размерности сети.

Слитный рукописный текст характеризуется относительно большим различием в начертаниях одинаковых символов. В зависимости от ближайшего окружения один и тотже символ может писаться по-разному. Отличия в начертании возникают и при изменении скорости письма, геометрической формы бумажного носителя.

Другой сложностью, возникающей при распознавании слитного текста является необходимость сегментации слова на символы перед распознаванием. Задача сегментации слитного рукописного текста может не иметь единственного формального решения. Таким образом, ошибочная сегментация слова на символы может приводить к падению качества распознавания отдельных символов, а при ошибочной сегментации — к фактически случайным результатам распознавания.

В настоящем исследовании мы поставили задачу изучить структуру штриха рукописного текста и создать математическую модель, описывающую произвольный штрих текста таким образом, чтобы визуально сходные штрихи имели близкое математическое описание, а непохожие — различное.

Сформулируем требования к модели штриха:

1) независимость от геометрических размеров. Действительно, в зависимости от условий письма штрихи, составляющие символы могут иметь произвольный размер, однако это не влияет (в разумных пределах) на восприятие текста человеком;

2) независимость от наклона и поворота. Во многих случаях (в частности в случае неограниченного рукописного текста) наклон письма нелинейно меняется в рамках одного документа. Подстройка модели под разные участки рукописи вручную сложна и, потому, нежелательна;

3) нечувствительность к носителю. Распознавать приходится как «бумажные» документы, так и слова, написанные от руки, или на досках, стенах и т.п. Способ оцифровки различных носителей также различается кардинально;

4) постоянное число характерных признаков штриха. Дальнейшее использование модели предполагает нормализацию данных под конкретный распознающий алгоритм (обычно нейронную сеть). Нормализация штриха не должна вносить дополнительных искажений после оцифровки.

Векторное представление штриха в той или иной степени отвечает требованиям, предъявляемым к модели, однако требуется некоторая доработка методики представления штрихов. Рассмотрим ее.

Для получения данных, пригодных для анализа исходное отсканированное изображение необходимо нормализовать по яркости и цвету (см. [7]) и сегментировать (см. [6]). Алгоритмов нормализации яркости и сегментации существует достаточно много, в настоящей работе рассматривается более удобный для исследовательских нужд алгоритм, основанный на работах Wang и Suen (см. [9]). Результат работы алгоритма, показан на рис.1.

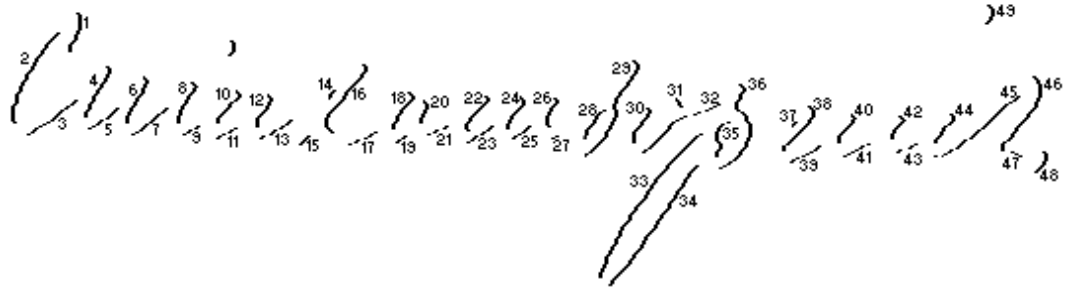

Рис.1. Результат сегментации растрового изображения

Для векторизации растра авторами используется алгоритм potrace, представляющий растр набором замкнутых многоугольников или т.н. *безьеугольников* — замкнутых фигур стороны которых заданы кривыми Безье.

Используя ряд методик, описанных в документации на potrace [5], potrace преобразует растровое изображение в набор замкнутых путей P_i. Замкнутый путь P_i соответствует некоторому элементу, обозначим его W_j. Индексы отличаются, т.к. некоторые пути могут соответствовать незначимым элементам, или оставшемуся после фильтрации шуму.

Таким образом, задача классификации путей P_i в элементы W_j может быть сформулирована следующим образом: найти такую классифицирующую функцию, что:

$$F(P_i) = \begin{cases} W_j, P_i \in W \\ \varnothing, P_i \notin W \end{cases}, \qquad (4)$$

где W — множество известных рукописных элементов.

Рассмотрим векторное представление некоторого штриха:

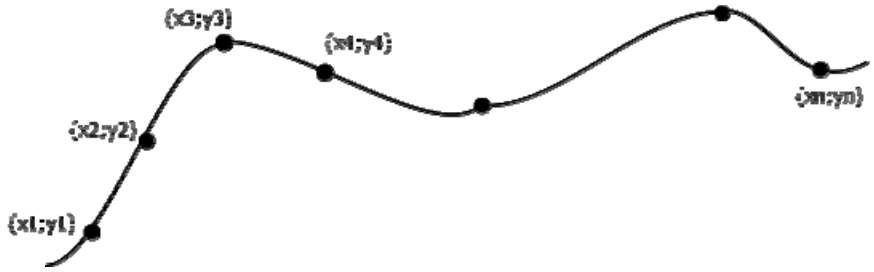

Рис.2. Векторное представление штриха

Кривая (штрих) представлена множеством координат точек $\{\{x_1, y_1\}, \{x_1, y_1\}, \ldots\{x_n, y_n\}\}$. Нейронные сети в силу особенностей их архитектуры используют входные вектора (традиционно обозначаемые $\vec{x}$) некоторой размерности, определяемой разработчиком при проектировании сети.

Результат векторизации растра, представленный на рис. 2. имеет произвольное количество узловых точек, характеризуемых координатами и (опционально) радиусом кривизны в данной точке. Данные такого рода плохо подходят для представления входных векторов нейронных сетей потому что зависят от начала координат системы штрихов, поворота ее относительно (обычно) верхнего левого угла при оцифровке и масштаба изображения.

Таким образом, нам необходимо построить такую модель векторного представления штриха, которая была бы инварианта относительно указанных выше переменных.

Каждую кривую (штрих) необходимо разбить на постоянное число сегментов. Для этого находим длину каждого вектора, из которых состоит кривая по формуле:

$$|\overrightarrow{p_n}| = \sqrt{(x_n - x_{n-1})^2 + (y_n - y_{n-1})^2}, \quad (1)$$

находим суммарную длину пути L как:

$$L = \sum_{i=1}^{n} |\overrightarrow{a_n}|. \quad (2)$$

Находим длину сегмента как:

$$l = \frac{L}{N}, \quad (3)$$

где N необходимое число сегментов или размерность входного вектора нейронной сети.

Для разбиения пути L на сегменты, строятся ряд окружностей радиуса l, так что центром первой является вершина v_1 (w_1). Точка w_2 определяется методом поиска точки пересечения окружности и отрезка, с центром в этой точке строится новая окружность того же радиуса, определяющая вершину w_3, и так далее, пока не будет достигнуто такое состояние, что все узлы лежат внутри очередной окружности (рис 3).

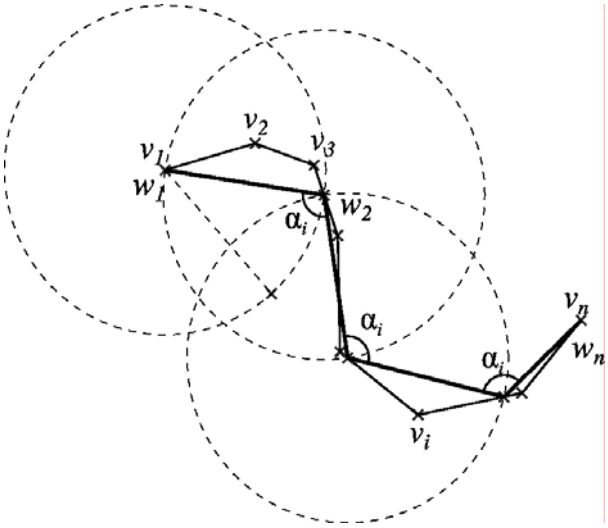

Рис.3. Выделение сегментов и определение углов поворота

Находим координаты точек пересечения прямой и окружности:

$$\begin{cases} \dfrac{x-x_1}{x_2-x_1} = \dfrac{y-y_1}{y_2-y_2} \\ (x-x_0)^2+(y-y_0)^2 = l^2 \end{cases}, \qquad (4)$$

где, $(x_0;y_0)$ – центр окружности, $(x_1;y_1)$ и $(x_2;y_2)$ – координаты начала и конца прямой, l – радиус окружности.

Из системы уравнений (4) выражаем значения x и y.

Из первого уравнения выражаем значение x:

$$(x_2-x_1)(y-y_1) = (x-x_1)(y_2-y_1),$$
$$x_2-x_1 = a,$$
$$y_2-y_1 = b,$$
$$a(y-y_1) = b(x-x_1),$$
$$ay-ay_1 = bx-bx_1,$$
$$ay-ay_1-bx+bx_1,$$
$$bx_1-ay_1 = c,$$
$$ay-bx+c = 0,$$

$$x = \frac{ay+c}{b},$$
$$(x-x_0)^2+(y-y_0)^2 = l^2,$$
$$x^2-2xx_0+x_0^2+y^2-2yy_0+y_0^2 = l^2,$$
$$2x_0 = d,$$

$$2y_0 = e,$$
$$x_0^2 + y_0^2 = g,$$
$$x^2 + y^2 - sx - ey + g = l^2,$$

Подставляем значение x, выраженное из первого уравнения:

$$(\frac{ay + c}{b})^2 + y^2 - d\left(\frac{ay + c}{b}\right) - ey + g = l^2$$

Приводим уравнение к общему знаменателю:

$$\frac{(ay + c)^2 + y^2b^2 - db(ay + c) - eyb^2 + gb^2 - l^2b^2}{b^2} = 0,$$
$$(ay + c)^2 + y^2b^2 - db(ay + c) - eyb^2 + gb^2 - l^2b^2 = 0,$$
$$a^2y^2 + 2acy + c^2 + y^2b^2 - abdy - bdc - eyb^2 + gb^2 - l^2b^2 = 0,$$
$$y^2(a^2 + b^2) + y(2ac - abd - eb^2) + c^2 + gb^2 - abc - l^2b^2 = 0,$$
$$(a^2 + b^2) = m,$$
$$2ac - abd - eb^2 = n,$$
$$c^2 + gb^2 - abc - l^2b^2 = s,$$
$$my^2 + ny + s = 0.$$

Находим дискриминант уравнения:

$$D = n^2 - 4ms \qquad (5)$$

если $D>0$, то получаем две точки пересечения прямой и окружности;
если $D=0$, то получаем одну точку пересечения прямой и окружности;
если $D<0$, то точек пересечения прямой и окружностью нет, следовательно мы достигли конца штриха.

Получаем координаты точек пересечения прямой и окружности:

$$y = \frac{-n \pm \sqrt{D}}{2m} \qquad x = \frac{ay + c}{b}.$$

Находим координаты векторов по найденным точкам пересечения как:

$$\overrightarrow{p^j} = \{x_n - x_{n-1}; y_n - y_{n-1}\}. \qquad (6)$$

Получаем восемь векторов равной длины и определяем угол между ними по формуле:

$$\varphi = arccos\left(\frac{x_1x_2 + y_1y_2}{\sqrt{x_1^2 + y_1^2} * \sqrt{x_2^2 + y_2^2}}\right). \qquad (7)$$

Угол φ рассчитывается до тех пор, пока D (см. 5) не окажется меньше 0.

Для исследования описанной выше модели была создана программа на АЯП Python - (объектно-ориентированном высокоуровневом языке программирования с динамической семантикой).

Для примера используем выделенные штрихи рукописного слова на рис. 1. Программа измеряет длину каждого штриха, вычисляет радиус окружности, строит вектора и вычисляет угол между ними по формуле (7). Пример расчета для штриха № 36 показан на рис.4. Результаты вычисления значений для каждого штриха представлены в табл.1, отрицательные значения углов соответствуют повороту против часовой стрелки (см. рис. 4).

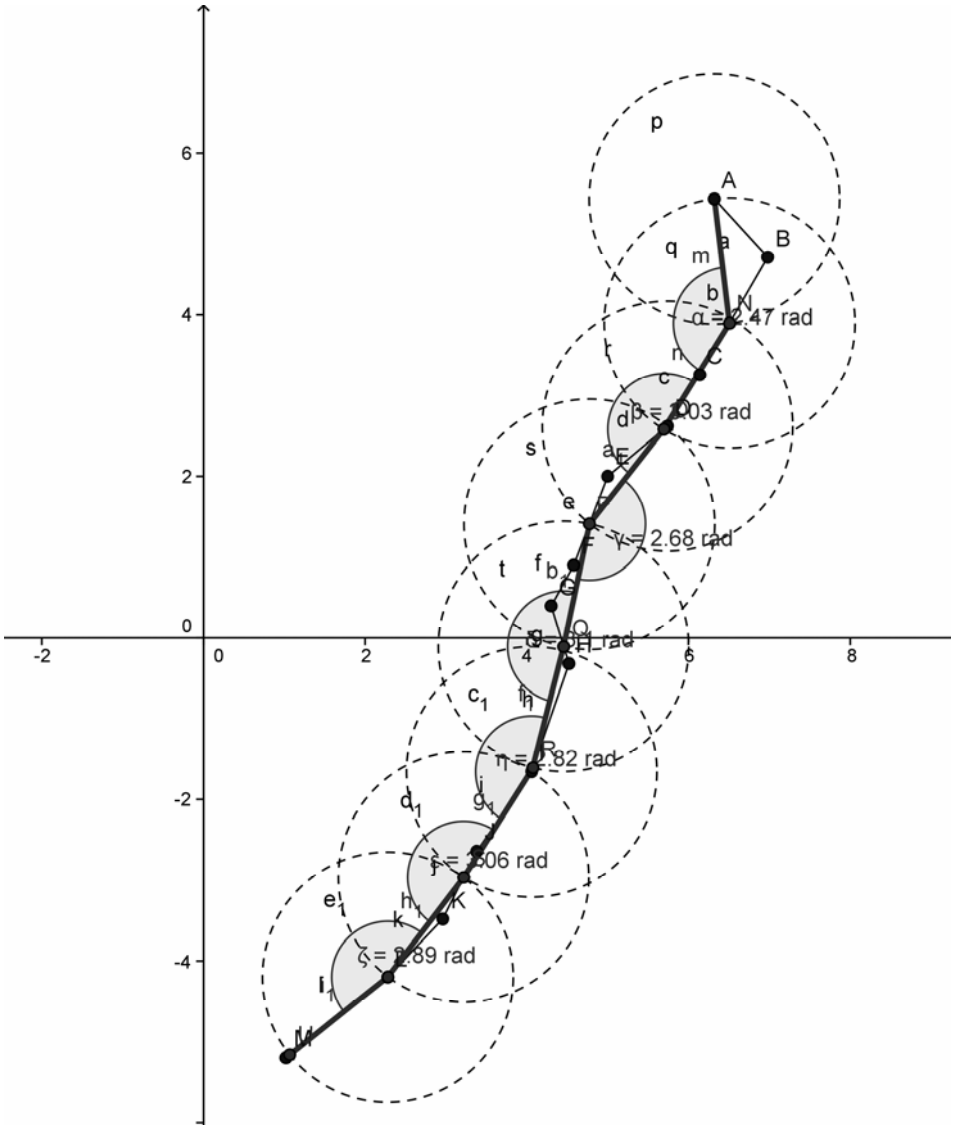

Рис.4. Нормализация штриха

Таблица 1

Результаты определения углов поворота

№	***L***	***R***	φ
1	4,5	0,4	-0,92; -0,96; -0,97; -0,92; -0,91; 0,87; 0,85

№	*L*	*R*	φ
2	8,0	1,0	0,93; 0,99; 0,96; 1; 0,97; 0,97; 0,93
3	5,0	0,6	-0,87; 0,95; -0,99; -0,98; 0,93; -0,95; 0,93
4	4,5	0,56	-0,87; 0,95; -0,99; -0,98; 0,93; -0,95; 0,93
5	2,8	0,35	-1; -0,97; -0,96; -0,96; 0,96; -0,95; 1
6	4,2	0,52	-0,66; -0,95; -0,95; 0,98; 1; 0,91; -0,98
7	3,8	0,47	-0,9; 1; 0,98; -0,94; 0,97; -0,94; -0,98
8	4,0	0,5	-0,6; -0,96; 0,97; 0,99; 0,95; 0,92; 0,9
9	2,0	0,25	-0,93; -0,99; 0,98; 0,99; -0,95; 0,95; -0,96
10	3,5	0,4	-0,62; 0,96; 0,96; -0,94; 0,98; 1; 0,75
11	2,0	0,25	-0,99; -0,94; -0,99; -0,98; 0,97; -0,97; -0,98
12	3,5	0,4	-0,81; -0,82; -0,97; -0,98; 0,98; 0,81; 0,91
13	3,5	0,4	-0,93; -0,94; 0,98; 0,96; -0,93; 0,99; -0,97
14	1,0	0,1	1; 1; 1; 1; 1; 1; 1; 1
15	1,0	0,1	1; 1; 1; 1; 1; 1; 1; 1
16	7,0	0.87	-0,66; -0,98; 0,89; -0,92; 0,92; 0,68; 0,79
17	2,8	0,35	-0,92; 0,99; -0,95; 1; 0,96; -0,95; 0,99
18	4,0	0,5	-0,57; -0,74; 0,8; -0,9; 0,9; -0,98; 0,87
19	1,2	0,15	1; 1; 1; 1; 1; 1; 1; 1
20	2,5	0,3	-0,96; -0,66; -0,94; 0,95; 0,99; 0,79; 0,83
21	2,5	0,3	0,92; -0,95; -0,97; 0,94; 0,97; -0,89; 0,9
22	3,8	0,47	-0,52; 0,97; -0,95; 0,98; 0,93; 0,93; 0,66
23	2,0	0,25	-0,96; 0,97; -0,93; 0,96; 0,99; 0,95; -0,91
24	3,0	0,37	-0,95; -0,68; -0,96; 1; 0,96; 0,93; 0,93
25	2,0	0,25	-0,99; 0,95; -0,95; -0,99; -0,99; 0,97; 0,96
26	3,0	0,37	-0,53; 0,97; -0,9; 0,9; 0,85; 0,89; 0,82
27	1,0	0,1	1; 1; 1; 1; 1; 1; 1; 1
28	2,0	0,25	0,95; 0,84; -0,9; 0,84; -0,84; 0,88; 0,83
29	5,6	0,7	-0,75; -0,96; 0,85; -0,95; 0,99; -0,87; 0,99
30	2,0	0,25	-0,74; -0,84; 0,89; -0,9; 0,94; 0,89; 0,91
31	1,0	0,1	1; 1; 1; 1; 1; 1; 1; 1
32	4,5	0,56	-0,93; 0,96; 0,98; 0,9; -0,98; 0,95; -0,94
33	9,0	1,12	0,95; -0,99; 0,96; 0,98; -0,99; 0,97; 0,99
34	7,5	0,9	0,98; 0,99; 0,99; -0,96; 0,97; -0,98; -0,92
35	2,5	0,26	0,97; 0,94; 0,86; 0,85; 0,84; 0,86; -0,87
36	5,0	0,6	-0,9; 0,68; -0,88; -0,82; -0,9; -0,9; -0,9
37	0,8	0,1	1; 1; 1; 1; 1; 1; 1; 1
38	4,0	0,5	-0,7; -0,79; -0,98; -0,95; 0,98; 0,94; -0,93
39	2,5	0,3	-0,99; -0,97; 1; -0,98; 0,99; 0,96; 0,99
40	2,5	0,3	-0,79; -0,76; -0,93; 0,93; 0,93; -0,94; 0,96
41	2,5	0,3	0,99; 0,91; 0,99; -0,99; -0,93; 0,89; -0,99
42	2,0	0,25	-0,86; -0,59; -0,93; 0,91; 0,99; 1; 0,78
43	1,6	0,2	0,98; 0,96; -0,94; -0,93; 0,93; -0,98; 0,96
44	2,5	0,3	-0,79; -0,72; -0,94; 0,87; 0,9; -0,85; 0,93
45	7,5	0,9	0,97; 0,98; 1; -0,97; -0,99; -0,96; -0,95
46	6,5	0,8	-0,78; -0,94; -0,88; 0,99; -0,98; -0,97; 0,8
47	1,0	0,1	1; 1; 1; 1; 1; 1; 1; 1
48	1,6	0,2	-0,97; -0,86; -0,93; -0,92; -0,98; -0,88; 0,98
49	1,6	0,2	-0,96; -0,8; -0,83; -0,88; -0,94; 0,96; -0,99

Исследование представленной модели сетью Кохонена показало что:

1. Данное представление применимо для использования совместно с нейросетевыми моделями.

2. Сеть устойчиво выделяет 4 категории штрихов, в настоящее время ведется работа по созданию обратного визуализатора внутреннего представления сети Кохонена в штрихи для удобного анализа результатов классификации.

3. Данная модель может быть использована для уточнения алгоритмов сегментации исходного рукописного текста и для создания обучающих выборок для обучения нейросетевых моделей.

ЛИТЕРАТУРА

1. *Долгова, Е.В., Курушин Д.С.* Компьютерные нейросетевые технологии, Пермь, ПГТУ, 2008.
2. *Мисюрёв, А.В.* Использование искусственных нейронных сетей для распознавания рукопечатных символов, [Электронный документ] (http://ocrai.narod.ru/hp.html). Проверено 2010.12.20.
3. Шаров, С.А. Статистика слов в русском языке. [Электронный документ] (http://www.lingvisto.org/artikoloj/ru_stat.html). Проверено 12.02.2011.
4. *Jaehwa, Park, Venu Govindaraju, and Sargur N. Srihari*. Efficient word segmentation driven by unconstrained handwritten phrase recognition. In Proceedings of International Conference on Document Analysis and Recognition, pages 605-608, 1999.
5. *Selinger, P.* Potrace: a polygon-based tracing algorithm языке. [Электронный документ] (http://potrace.sourceforge.net/potrace.pdf). Проверено 12.02.2011.
6. *Райер, И.А.* Сегментация штрихов и их соединений при распознавании рукописного текста / Материалы Международной конференции по компьютерной графике и машинному зрению. Ч.III: С.151-155, Москва, 1999.
7. *Абраменко, А*. Компьютер читает [Электронный ресурс]. — Электрон. текстовые дан. — 2000. — Режим доступа: http://www.ocrai.narod.ru.
8. *Кузнецов, А.В*. Распознавание рукописного текста [Электронный ресурс]. — Электрон. текстовые дан. — 2001. — Режим доступа: http://www.masters.donntu.edu.ua/2001/fvti/kuznetsov/diss/lib/aiconcep/rec_text.htm
9. *Suen, H.M., Wang, J.F.* «Segmentation of Uniform Colored Text from Color Graphics Background», VISP-144, 1997, P. 317-322.

THE INVESTIGATION OF THE HANDWRITING STROKES MODEL

E. Dolgova, D. Kurushin

Perm State Technical University
29, Komsomolsky Avenue, Perm, Russia

Abstract. In this paper we propose and study a model of handwriting strokes. The model consider handwritten text as a collection of strokes of several known classes. Model allows us to abstract from the initial conditions of the writing, such as angle, velocity, density, etc. Thanks to the procedure of tracing the model is irrelevant to noise of the original image and the way it was received. We conducted an experiment, confirming the applicability of the model for use in pattern recognition problems.

Key words: neural network, mathematical model, python, recognition, vector, count, language model, recognition, stroke.

УДК 004.67 : 519.688