

ИССЛЕДОВАНИЕ ОДНОГО МЕТОДА РАСПОЗНАВАНИЯ РУКОПЕЧАТНЫХ СИМВОЛОВ

Дробков А.В., Семенов А.Б.
Кафедра информационных технологий

Поступила в редакцию 25.10.2009, после переработки 12.11.2009.

В данной статье предлагается и исследуется метод распознавания рукопечатных символов. Выбираются признаки для растров символов, конфигурация классификатора-распознавателя и изучается точность нескольких полученных распознавателей на реальных данных.

In this paper we investigate a method for handprinted characters recognition. We select characteristics for symbol rasters, select classifier's configuration and research the precision of the proposed method on a real input data.

Ключевые слова: распознавание символов, классификация, рукопечатные символы, искусственный интеллект.

Keywords: characters recognition, classification, handprinted characters, artificial intelligence.

1. Введение

Распознавание символов является одной из главных задач среди проблем искусственного интеллекта. Хорошо известно, что распознавание символов широко применяется на практике. Программные комплексы, способные распознавать символы из электронных и бумажных документов с высокой точностью, могли бы ускорить процесс ввода данных в компьютер, сделав его почти полностью автоматическим. На сегодняшний день такие комплексы уже существуют, однако говорить об их массовости и универсальности преждевременно, поскольку известных программных решений такого типа не много, и каждое из них решает задачу распознавания со своими специфическими ограничениями. Большой успех был достигнут в последнее время в области распознавания машинопечатных символов (т.е. символов некоторого заданного шрифта). Символы же, написанные человеком и получаемые с бумажных носителей, значительно труднее распознавать, по понятным причинам широкой варьированности почерка. До сих пор до конца не решена задача распознавания рукописного письма. В данной же статье исследуется конкретный метод распознавания категории символов, получаемых с бумажных носителей, заполненных человеком - рукопечатных символов. Фактически, данный термин обозначает печатные символы, написанные шариковой ручкой или на графическом планшете от руки.

2. Распознавание анкет

Предположим, что у нас имеется некоторый образец бумажной анкеты, на которой присутствуют ячейки. Когда анкета заполняется человеком, в каждую ячейку вписывается печатная буква (или же ячейка оставляется пустой, см. рис. 1).



Рис. 1: Часть изображения заполненной отсканированной анкеты

Процесс получения информации из анкеты в текстовом виде можно разбить на две подзадачи:

- 1) Обработка и анализ отсканированного изображения анкеты с целью получения (вырезания) растров вписанных в неё букв.
- 2) Распознавание растров вырезанных символов.

Автор данной статьи в работе [2] показывает, как можно решить задачу первого этапа. Предложенный подход состоит в следующем:

- 1) Цветное изображение отсканированной анкеты подвергается *бинаризации*.
- 2) Выполняется выравнивание полученного бинарного изображения (на случай, если пользователь положил анкету под сканер неровно, и.т.д.).

В [2] алгоритм выравнивания основан на преобразовании Хафа. С помощью этого преобразования на анкете производится поиск известных графических элементов (в частности, окружностей), на основе найденных координат которых легко определяется требуемый угол поворота и величина смещения анкеты. Можно предполагать, что сканер обладает высоким качеством сканирования, а вышеупомянутые операции преобразования крайне незначительно искажают изображение, поэтому из выравненной анкеты впоследствии вырезаются прямоугольные области, координаты которых известны из образца анкеты. В [12] изучалось несколько методов бинаризации (в том числе, адаптивных). После проведенных вычислительных экспериментов оказалось, что для выбранных анкет является пригодным простой метод пороговой бинаризации, который к тому же является и самым быстрым среди различных методов бинаризации. Было экспериментально установлено,

что бинаризация с порогом 215 даёт неплохие результаты качества бинаризации на многих анкетах. Следующий этап — распознавание растров вырезанных из анкеты букв.

3. Формирование признаков описаний

Задачу распознавания символов можно рассматривать в терминах *классификации*. Пусть у растра имеется n вычисляемых признаков, по которым классификатор будет судить, какой букве из алфавита соответствует данный растр. Следует логичный вывод, что признаки должны по-возможности выбираться так, чтобы по ним можно было максимально правильно отличать один класс символов от другого (иначе говоря, для символов отличных классов их численные значения также должны отличаться на достаточную величину). Выбор признаков является нетривиальной задачей.



Рис. 2: Признаки R1-R6 символа «А»

В данной статье предложено использовать следующие числовые признаки для классификации:

R1, R2, R3, R4, R5, R6	Отношения площадей буквы в соответствующих регионах ограничивающего прямоугольника (см. рис. 2) к общей площади буквы
GR	Отношение площади буквы к площади её ограничивающего прямоугольника (кол-во черных пикселей, деленное на общее число пикселей)
WHR	Отношение ширины ограничивающего прямоугольника к его высоте
GravityCenterX, GravityCenterY	Относительные координаты центра тяжести прямоугольника

Все вышеописанные признаки выбирались исходя из эвристических соображений о возможности эффективно отличать различные классы символов. Например, буква Т имеет верхнее преобладание площади, т.е. признак R1 численно больше признака R3 чего нельзя, например, сказать о букве О. Итого получается, что для каждого распознаваемого растра буквы будет вычислен такой вектор:

$$w = (R1, R2, R3, R4, R5, R6, GR, WHR, GravityCenterX, GravityCenterY)$$

4. Классификатор распознаватель

В качестве классификатора-распознавателя выбрана многослойная нейронная сеть прямого распространения. Нейронные сети могут выступать как превосходные классификаторы. Считается, что нейронные сети обладают свойством *обобщения* данных при обучении, а так же весьма выгоден тот факт, что при небольшом числе нейронов они способны обеспечивать высокую скорость вычисления *отклика*. Однако, существуют и другие модели классификаторов. В работе [9] автором также упоминаются в частности такие классификаторы, как

- байесовский классификатор,
- классификатор по ближайшему соседу,
- полиномиальный классификатор.

Также отмечают сложности, связанные с использованием этих классификаторов, которые делают их мало применимыми на практике. Рассмотрим более детально математическую модель нейронной сети, используемую в данной статье. Нейрон имеет $n+1$ входов, на эти входы подаются вещественные числа s_1, s_2, \dots, s_n и на последний $n+1$ -ый вход всех нейронов всегда подается единица. У нейрона имеются весовые коэффициенты w_1, w_2, \dots, w_{n+1} , которые соответствуют в том же порядке входным значениям s_i . Первое, что делает нейрон — вычисляет следующую сумму

$$Sum = \sum_{i=1}^n w_i s_i + w_{n+1}. \quad (1)$$

Данный результат затем подставляется в функцию активации ϕ (см. (2)):

$$\phi(Sum) = \phi\left(\sum_{i=1}^n w_i s_i + w_{n+1}\right) \quad (2)$$

— данное число и является выходным значением нейрона на всех его имеющихся выходах. На рисунке 3 изображен один нейрон. В качестве функции активации в работе использовалась *логистическая функция*:

$$\phi(v) = \frac{1}{1 + e^{-v}}. \quad (3)$$

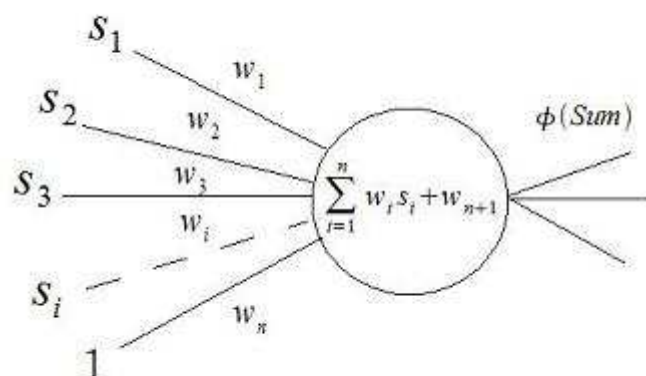


Рис. 3: Единичный нейрон

Всего в нейронной сети, которая использовалась в работе, три слоя, на первом — 10 нейронов, 20 и 31 на втором и третьем, соответственно.

Итак, процесс распознавания одного символа с помощью нейронных сетей выглядит следующим образом:

- 1) По графическому представлению символа вычисляется его вектор признаков w (по правилам, описанным в разделе 2). В (5), (6) и (7) показаны векторы для реальных символов
- 2) Вектор w подается на вход обученной нейронной сети, которая производит вычисления и возвращает отклик — вектор o . В (4) показан эталонный вектор для символа «Б», второй элемент которого равен единице, а остальные нулю.
- 3) По вектору o определяется ответ нейросети о том, какой именно символ был входным. Символ соответствует индексу максимального по величине числа в этом векторе (т.е., например, если максимальное число в векторе — предпоследнее, то это символ «Ю»).

$$\{0, 1, 0, 0, 0, 0, \dots, 0\} \tag{4}$$

На рисунке 4 показаны примеры символов, которые распознает описываемый в данной работе метод. Формулы (5) и (6) показывают вычисленные вектора признаков для первых двух букв «А». Вектор (7) — это вектор признаков для буквы «У».

$$\{0.282 \ 0.425 \ 0.246 \ 0.297 \ 0.294 \ 0.329 \ 0.519 \ 0.574 \ 0.629 \ 0.419 \} \tag{5}$$



Рис. 4: Реальные символы для распознавания

$$\{0.226 \ 0.457 \ 0.267 \ 0.295 \ 0.358 \ 0.277 \ 0.477 \ 0.575 \ 0.655 \ 0.344 \} \quad (6)$$

$$\{0.331 \ 0.314 \ 0.279 \ 0.197 \ 0.335 \ 0.272 \ 0.528 \ 0.471 \ 0.595 \ 0.396 \} \quad (7)$$

Даже на глаз видна некоторая схожесть первых двух векторов и их заметное отличие от третьего вектора. Дальнейшая задача состоит в том, что нейронную сеть необходимо предварительно *обучить* на собранной заранее обучающей базе символов. Мы будем добиваться, чтобы на основе этих данных нейронная сеть смогла бы корректно классифицировать поступающие ей на входы образцы, которые никогда не предоставлялись ей на стадии обучения.

5. Сбор обучающей базы

В работе [9] автор использовал обучающую базу, состоящую из более чем полу-миллиона символов. Создание базы уже из нескольких тысяч символов представляет собой сложную технически задачу, поскольку требуется эти символы сперва нарисовать. В работе предложен метод полуавтоматической генерации базы символов, а так же метод полностью автоматической генерации базы символов на основе уже сгенерированных символов. Последний, как будет показано далее, используется для проверки *устойчивости* полученного распознавателя. Под устойчивостью, во-первых, понимается некоторая субъективная оценка, характеризующая насколько хорошо распознаватель смог правильно определить символы, написанные почерком, отличным от обучающего, во-вторых, численные характеристики точности, по которым и дается эта оценка. Для генерации обучающей базы символов автором была использована сенсорная панель (графический планшет). Была разработана программа с графическим интерфейсом, получающая данные с этой панели. Пользователь может рисовать символ на сенсорной панели, штрих в режиме реального времени отображается на экране. После того как пользователь нарисовал символ, у него есть возможность сохранить его в виде bmp-файла. База символов представляет собой набор из тридцати одной папки, которые соответствуют распознаваемым символам. Внутри этих папок лежат графические изображения размерами 62x62 пикселя (это умышленно принятое ограничение, размер обусловлен стандартным размером клетки на бумажной анкете, с учетом сканирования с разрешением 300 dpi). В данной работе база символов была собрана с помощью графического планшета двумя людьми в равных долях. В общей сложности в базе содержится 7130 символов.

6. Автоматическая генерация базы символов

Была сделана успешная попытка реализовать алгоритм, позволяющий автоматически генерировать новые символы, применяя к уже имеющимся различные геометрические деформации. В качестве таких деформаций были взяты преобразования раstra: поворот, скос и масштабирование. Эти деформации задаются стандартными аффинными преобразованиями растров:

$$\begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix} \quad (8)$$

$$\begin{pmatrix} scale & 0 \\ 0 & scale \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix} \quad (9)$$

$$\begin{pmatrix} 1 & -scale \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix} \quad (10)$$

Опишем, как устроен процесс генерации.

- 1) Алгоритму на вход подается растровое изображение исходной буквы.
- 2) В алгоритме устанавливается число преобразований $nTransforms$, которые будут проведены последовательно с буквой. Автор использовал $nTransforms$, равное 3.
- 3) На очередном шаге алгоритма генерируется случайное число из равномерного целочисленного распределения $[1; nTransforms]$ которое определяет номер преобразования, которое будет применяться на данном шаге.
- 4) Как только преобразование установлено, алгоритм генерирует случайным образом параметры этого преобразования, применяет его к символу и переходит на шаг 2, если данное преобразование не является последним. Для преобразования масштабирования случайным параметром является масштаб. Для поворота – угол поворота, а для скоса – коэффициент скоса. Эти параметры генерировались с помощью равномерного распределения из следующих отрезков:
 - $[-0.1; 0.1]$ для угла поворота (в радианах),
 - $[0.9849; 1.005]$ для масштабирования,
 - $[-0.18; 0.18]$ для скоса.

Результаты работы алгоритма можно видеть на рисунке 5.

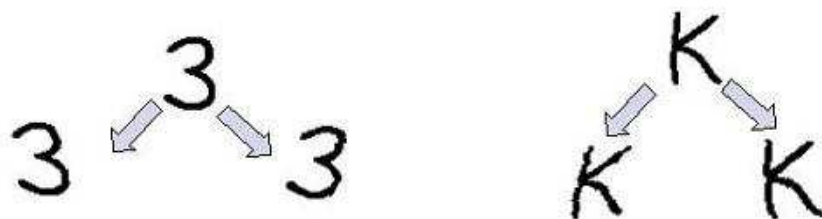


Рис. 5: Полученные с помощью деформаций символы

7. Алгоритм обучения нейросети

Нейросеть обучалась в работе алгоритмом обратного распространения ошибки. Данный алгоритм подробно описан в [3]. Поскольку этот алгоритм считается одним из самых популярных в литературе, и в работе не было внесено каких-либо принципиальных доработок при его реализации, мы опишем лишь его идею. По символам обучающей базы строится база признаков, состоящая из векторов. Данные вектора начинают подаваться на вход обучаемой нейронной сети в произвольном порядке. Условие случайности порядка является рекомендуемым многими авторами и было подтверждено экспериментально авторами данной работы: по сравнению с фиксированным порядком, случайный порядок позволяет за меньшее время достичь более высокой точности обучения, кроме того, избегав мгновенного попадания в локальный минимум (подтверждено экспериментально для конкретного примера). Опишем шаги алгоритма:

1) *Инициализация.*

Веса нейронной сети устанавливаются случайным образом — каждый коэффициент получает значение с помощью равномерного распределения из отрезка $[-2.5; 2.5]$.

2) *Проверка условия выхода.*

На данном этапе проверяется условие, что алгоритм уже отработал указанное время, либо достигнута требуемая точность обучения, либо достигнуто указанное число *эпох*, т.е. циклов обучения (шаги 2-5 как раз и задают этот цикл). Если условие истинно — алгоритм выходит, иначе следует переход к пункту 3.

3) *Случайное упорядочивание обучающего множества.*

Обучающие вектора признаков переупорядочиваются случайным образом.

4) *Цикл по всему обучающему множеству.*

Для каждого элемента обучающего множества в порядке, заданном в пункте 3, считается отклик нейросети и вычисляется Евклидово расстояние между этим откликом и ожидаемым откликом. Если это расстояние больше заданного фиксированного числа, то происходит коррекция коэффициентов нейросети, иначе алгоритм сразу же переходит к следующему элементу обучающего множества.

5) *Переход к шагу 2.*

Когда цикл заканчивается, алгоритм переходит к проверке условия завершения.

Формально, цель применения данного алгоритма — это решение следующей оптимизационной задачи

$$\sum_i (\Phi(p_i) - e_i)^2 \rightarrow \min, \quad (11)$$

где p_i — это вектор признаков i -го символа из обучающей базы, Φ — оператор, вычисляющий вектор-отклик нейросети на этом векторе, а e_i — требуемый отклик.

8. Полученные результаты точности

В таблице 1 приведены основные результаты распознавания на различном числе символов в обучающей базе. По результатам видно, что с увеличением ее размера точность возрастает. Элементы выходного вектора нейросети задают *гипотезы*, а величина этих элементов — *степень уверенности* в этих гипотезах.

Таблица 1: Основные результаты распознавания

№	Число символов в обучающей базе	Число символов в тестовой базе	Точность распознавания	Точность среди первых 3х гипотез по степ. уверенности
1	3576	3554	3032/3554 (85%)	3344/3554 (94%)
2	1787	5316	4432/5316 (83%)	4993/5316 (93%)
3	5334	1778	1563/1778(87%)	1711/1778 (96%)

В следующей таблице приведены результаты точности на символах, сгенерированных по методу, описанном в разделе 5.

Таблица 2: Результаты устойчивости к изменению почерка

№	Число символов в обучающей базе	Число символов в тестовой базе(сгенерирована автоматически)	Точность	Точность среди первых 3х гипотез
1	3576	31	19 / 31 (61%)	23/31 (74%)
2	3576	651	362 / 651 (55%)	444/651 (68%)
3	3576	1581	872/1581 (55%)	1106 /1581 (69%)
4	5334	31	21/31 (67%)	25/31 (80%)
5	5334	651	386/651 (59%)	509/651 (78%)
6	5334	1581	964/1581 (60%)	1260/1581 (79%)

Заключение

В работе получен результат точности распознавания, составивший 87% корректно распознанных символов. Данный результат дает основание на продолжение исследований, которые будут направлены на улучшение построенных механизмов распознавания и подбор других признаков растра, позволявших бы повысить эту точность.

Если говорить о потенциальном результате, то в 96% случаев правильный ответ присутствовал среди трех первых по степени уверенности гипотез. Данный результат близок к отличному и прямым образом свидетельствует о возможностях улучшения точности метода. Вернемся к полученной точности в 87%: этот результат, конечно, ставит под сомнение применимость метода в промышленном программном обеспечении, однако нельзя сказать, что он оказался ниже ожидаемого, так-как, во-первых, размер обучающей базы, рекомендуемый согласно неравенству

$$N > \frac{W}{\epsilon}, \quad (12)$$

например, для точности в 95% составляет 19620 (где W — число коэффициентов в нейросети, N — размер обучающей базы, ϵ — доля ошибок), что более чем втрое превышает имеющееся в работе число обучающих символов. Это может говорить о том, что на имеющейся базе сложно будет получить абсолютную точность распознавания 95% (или же, вообще невозможно). Формула утверждает, что, например, для ошибки в 10% количество примеров обучения должно более чем в 10 раз превосходить количество свободных параметров сети. Рекомендуемые размеры обучающих баз для обеспечения точности распознавания 87% и 95% приведены ниже. Число коэффициентов N в нейросети равно 981.

Для $W = 981$, $\epsilon = 0.13$ результат $N > 7546$, точность 87%.

Для $W = 981$, $\epsilon = 0.05$ результат $N > 19620$, точность 95%

Также отметим, что из таблиц 1 и 2 видно, что в нашем случае точность распознавания возрастала вместе с возрастанием числа элементов в обучающей базе, что экспериментально подтверждает приведенные выше рассуждения.

Во-вторых, для сравнения, в работе [9] автор в своем методе вводил вектор, состоящий из сорока трех признаков, что наводит на мысль о том, что размер вектора, равный 10, возможно, не является достаточным, чтобы обеспечить высокую точность распознавания, хотя безусловно, это зависит ещё и от самих признаков, а не только от их числа.

Список литературы

- [1] Р. Гонсалес, Р. Вудс. Цифровая обработка изображений. М: Техносфера, 2005. - 1072 с.
- [2] Дробков А.В. Автоматическая обработка анкетных данных с использованием параллельных вычислений, курсовая работа. ТвГУ, ф-т ПМиК, 2008.
- [3] Хайкин С. Нейронные сети: полный курс, 2е издание. : Пер. с англ. М. Издательский дом «Вильямс», 2006.

- [4] Калан Р. Основные концепции нейронных сетей. : Пер. с англ. — М : Издательский дом «Вильямс», 2001 — с. : ил.
- [5] Немнюгин С., Стесик О. Параллельное программирование для многопроцессорных вычислительных систем. Спб. 400 с. ВHV, 2002.
- [6] Поляков А.Ю., Брусенцев В.А. «Программирование графики: GDI+ и DirectX». – СПб.: БХВ-Петербург, 2005
- [7] Прэтт. Цифровая обработка изображений, I и II тома. Изд. МИР, 1982 г.
- [8] Рутковская Д., Пилиньский М. Нейронные сети, генетические алгоритмы и нечеткие системы: Пер. с польск. И. Д. Рудинского. Рутковский Л. - М.: Горячая линия -Телеком, 2006. - 452 с.
- [9] Ян Д.Е. Исследование, развитие и реализация методов автоматического распознавания рукописных текстов в компьютерных системах, диссертация на соискание степени кандидата физико-математических наук. М. 2003 - 168 с.
- [10] Семенов А.Б. Обработка и анализ изображений с использованием языка JAVA, курс лекций.
- [11] Семенов А.Б. Основы компьютерной графики: Учеб пособие. – Тверь: Твер. гос. ун-т, 2007. – 136 с.
- [12] Федоров А. Бинаризация черно-белых изображений: состояние и перспективы развития. Интернет ресурс: <http://iu5.bmstu.ru/philippovicha/ITS/IST4b/ITS4/Fyodorov.htm>
- [13] Хокинс Дж., Блейкли С. Об интеллекте - 240 с., с ил.; твердый переплет 2007, 1 кв.; Вильямс