УДК 004.93'14

РАСПРЕДЕЛЕННАЯ ПРОГРАММНАЯ ТЕХНОЛОГИЯ РАСПОЗНАВАНИЯ ТЕКСТОВОЙ ИНФОРМАЦИИ

Алейкин В.В., Ладыженский Ю.В.
Донецкий национальный технический университет
E-mail: aleykin.vladislav@gmail.com

В статье рассмотрена методика создания распределенной системы распознавания текстовой информации. Описывается алгоритм и реализация комплексного подхода распознавания текста с искажениями с использованием нейроподобного, шаблонного и структурного методов. Представлены временные результаты распознавания для MISD архитектуры.

4

Общая постановка проблемы

В настоящий момент существует ряд программных продуктов, которые способны распознавать с вероятностью более 90% сканированные текстовые документы хорошего качества. Данные показатели достаточно велики, чтобы использовать такие программы, как в офисах при распознавании документов, так и в промышленности для контроля продукции и маркированных деталей.

При распознавании бумажных документов на практике ошибки в 10 буквах на одном листе не так значимы, как ошибки, получаемые в промышленной сфере.

Существующие программные продукты распознавания текстовой информации способны убрать слабые помехи в виде зернистого шума, связанного с низким качеством съемки. Более крупные помехи существующие программы не способны определить и убрать со снимков.

Постановка задач исследования

Важными для развития теории и практики распознавания являются Captcha-изображения. Captcha - это полностью автоматизированный публичный тест Тьюринга, призванный отличать компьютеры от людей (рис. 1). По отношению к автоматическому распознаванию существуют понятия «слабая» Captcha и «прочная» Captcha. В числе слабостей, облегчающих распознавание: фиксированный шрифт, фиксированное положение символов, отсутствие искажений, отделение символов от фона с использованием цветового ключа или размытия по Гауссу, лёгкое отделение символов друг от друга и другие [1].

Рисунок 1 – Публичный тест Тьюринга - Captcha

Распознавание Captcha-изображений является сложной задачей. В настоящее время нет системы, которая может распознать абсолютно любую Captcha, существующие системы способны распознать Captcha-изображения только определенного типа.

Для распознавания используются методы восстановления изображения по содержимому на странице, подходы с использование нейронных сетей, шаблонное сопоставление, ручное распознавание и другие [2]. При разработке технологии распознавания текстовой информации ставится задача распознавания Captcha-изображений [3] с повышенным уровнем шума в виде многочисленных отрезков кривых линий, и исследование эффективности применяемых методов при распознавании текстовой информации.

Структура программной системы включает: модуль распознавания, который реализует комплексное исследование символа используя нейроподобный алгоритм, шаблонное и структурное сопоставление; модуль предобработки, реализующий

серию алгоритмов для коррекции изображения и удаления посторонних шумов, негативно влияющих на процесс распознавания. Модуль предобработки выполняет сегментацию входного образа на области для распознавания.

Решение задачи и результаты исследований

Образ символа можно представить матрицей из N элементов, описывающих изображение. В разработанной программе N = 50*50 элементов. Каждый элемент системы может принимать вещественное значение от -1.0 до +1.0. Алгоритм получение такой матрицы показан в части «Обучение сети».

Взаимодействие двух различных образов (W, X) в сети описывается выражением:

$$Res = \sum_{i,j=0}^{N} (w_{ij} * x_{ij}),$$

где Res – результат сравнения двух образов, w_{ij}, x_{ij} – элементы матриц взаимодействий W и X.

Образ W является эталонным образом, по нему производится сравнение с входным образом. В базе данных системы хранится множество M = (W1, W2, …, Wn). Множество M можно определить как память системы, где находятся образы, на которых система была обучена.

Обучение сети

Алгоритм обучения сети имеет существенные отличия в сравнении с такими классическими алгоритмами обучения нейронных сетей как метод коррекции ошибки или метод обратного распространения ошибки. Отличие заключается в том, что вместо последовательного приближения к нужному состоянию с вычислением ошибки, все коэффициенты матрицы рассчитываются по одной формуле, за один цикл, после чего сеть сразу готова к работе.

Цикл обучения сети происходит в два этапа:

1. Вычисление коэффициентов матрицы W эталонного образа;
2. Назначение «горячих областей» – наиболее важных участков в символе. Только с помощью таких участков можно отличить два похожих по структуре символа. Например ‘O’ и ‘Q’ - нижний участок, ‘8’ и ’B’ – левый участок.

Вычисление коэффициентов базируется на следующем правиле:

$$W_{ij} = \begin{cases} 1.0 & (1) \\ p \cdot X_{ij} & (2) \\ 0.0 & (3) \\ n \cdot X_{ij} & (4) \end{cases}$$

где p и n коэффициенты обучения сети. Выражение (1) соответствует случаю, при котором пиксель черный и угол поворота символа = 0; выражение (2) соответствует черному пикселю и наличию угла поворота. Выражения (3) и (4) используются соответственно для белого пикселя. В разработанной программе для каждого обучаемого образа вначале коэффициенты p = 0.9, n = 0.1. Коэффициенты изменяются после каждого поворота символа следующим образом: p = p – 0.1; n = n + 0.1.

При обучении шаг поворота символа был выбран равным 2 градусам, максимальный угол поворота не превышает ±15 градусов.

После обучения в матрице W хранится образ символа, повернутый на разные углы. При такой организации процесса обучения при распознавании можно не учитывать угол поворота входного символа.

Отличительной особенностью обучения является то, что матрица весовых коэффициентов настраивается детерминированным алгоритмом раз и навсегда, и затем весовые коэффициенты больше не изменяются.

Процесс распознавания

Процесс распознавания можно разделить на три этапа:

1. Сопоставление входного образа с образом из базы данных.
2. Проверка «горячих пикселей».
3. Принятие решения.

Сопоставление двух образов происходит по следующему алгоритму:

$$Y = 0.0;\ C = 0;$$
$$Y =$$
$$Y = Y / C;$$

Важно отметить, что при сопоставлении входного образа с образом из памяти результат Y может принимать вещественное значения от -1.0 до 1.0. При этом при тестировании определено, что максимальное реальное значение на выходе < 1 (даже при сравнении с тем же образом на котором проходило обучение). Такой результат не является недоработкой, такие результаты получаются из-за наложения повернутых образов под разным углом.

Выходное значение Y определяет степень схожести входного образа с образом из памяти. Существуют следующие ситуации:

- $(Y < 0.1)$ – образы различны, переход к другому образу из базы данных;
- $(Y > 0.1)$ И $(Y > 0.3)$ – образы имеют схожие области, возможно образ повернут. Требуется повторить сравнение с поворотом входного образа.
- $(Y > 0.3)$ – входной символ сильно похож с образом из базы данных. Принятие символа, как результата.

После успешного завершения первого этапа, необходимо сравнить «горячие области». Алгоритм сопоставления такой же, как в первом этапе, только для меньшей области. Данный этап необходим, так как различие между результатами двух символов может быть 5-10% для всего образа. Такой показатель очень мал для выбора между двумя символами и часто получается из-за помех. Сравнение меньших областей, где обязательно должны быть

различия между символами приводят к правильному принятию решения.

Предобработка изображений

Для решения проблемы, связанной с большим количеством помех на изображении разработан специальный модуль предобработки. Данный модуль выполняет преобразование цветного изображения в монохромное, удаление помех в виде кривых, сегментирование на отдельные символы.

На вход модулю подается цветное изображение, содержащее символы с наложенным шумом. Входное изображение подвергается бинаризации по яркости для отделения нужной информации от фонового цвета; подсчитывается количество значимых черных пикселей на изображении и строится вертикальная гистограмма яркости (рис. 3); если количество значимых пикселей меньше установленного порогового значения (на гистограмме отсутствуют четко выделенные пики), то применяется алгоритм сегментирования изображения, использующий гистограмму яркости и высоту символа[5], иначе, если гистограмма содержит четко выделенные пики, то применяется алгоритм сегментирования по пикам вертикальной гистограммы яркости[6].

В данном алгоритме правильное пороговое значение очень важно, так как символы различаются по толщине написания, все символы разделяются на два класса. В первый класс входят все символы с жирным написанием, во второй – с обычным написанием. После описанных этапов на выходе получается монохромное изображение с удаленными помехами и шумом, на котором все символы отделены друг от друга.

На рис. 3 представлены результаты работы реализованного модуля. Слева вверху – входной образ. Справа – бинаризация образа. Снизу – гистограмма яркости. Прямоугольники на изображении – сегментированные символы.

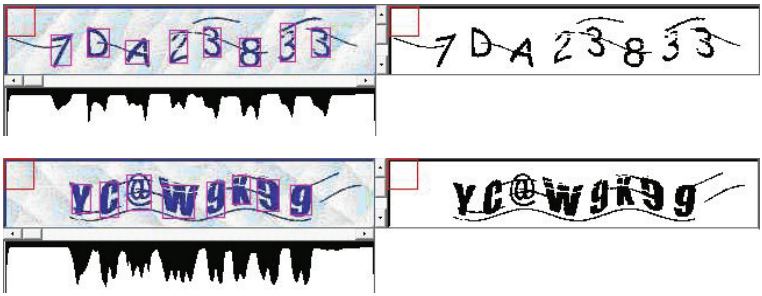

Рисунок 3 – Сегментация входного образа

4

Распределенная обработка

Система была обучена на 428 символах с курсивным и жирным написанием, различного шрифта и размера. При тестировании системы распознавания подавались 1500 различных изображений размером 320*60 пикселей. Входные изображения содержали по 5-10 символов с наложенными помехами. Среднее время распознавания отдельного изображения составило 0.32 секунды, при удачном коэффициенте определения в 68.7%.

Так как планируется расширение базы данных до 5-10 тыс. эталонных образов, то была введена параллельная обработка.

При использовании кластерной обработки пересылка данных увеличивает время распознавания, поэтому параллельность была организована с использованием API OpenMP (MISD архитектура), на многоядерном процессоре. Такая технология параллельной

Таблица 1 – Время параллельной обработки на многоядерном процессоре, с

Кол-во символов в изображении	Количество ядер в процессоре			
	1	2	3	4
5	0.18	0.10	0.08	0.07
8	0.34	0.18	0.13	0.11
10	0.39	0.21	0.14	0.12

обработки была выбрана из-за использования общего хранилища данных, что не требует выполнять операции пересылки данных.

При параллельной обработки на каждое ядро процессора подавался сегментированный участок на входном изображении. Методика распознавания остался без изменений. В таблице 1 показано время распознавания с использованием одноядерной системы и многоядерной. Частота каждого ядра равна 2.8Ghz.

Выводы

Разработан комплексный подход для распознавания с использованием нейроподобного алгоритма, шаблонного и структурного сопоставления. Создан программный модуль предобработки изображений, снижающий уровень помех и выполняющий сегментацию изображения.

Реализована программа распознавания зашумленных изображений (Рис. 4).

При тестировании разработанного алгоритма обработки и распознавания текстовой информации на 1500 различных изображениях вероятность корректного распознавания составила 68.7%.

Реализована параллельная обработка изображений с использованием архитектуры общей памяти и API OpenMP. Произведена оценка временных затрат параллельной обработки на многоядерном процессоре.

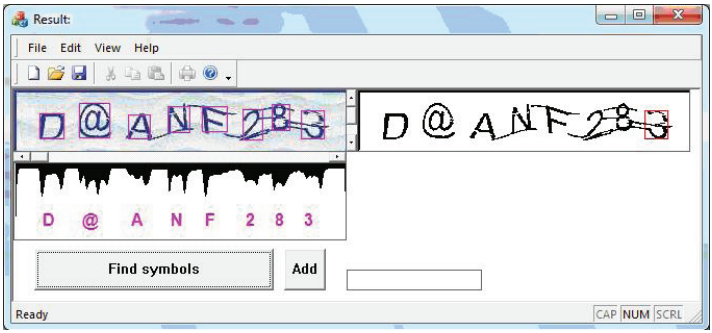

Рисунок 4 – Программа распознавания символов

Литература

[1] “ReCAPTCHA – Надежная защита от спама”. Электронный ресурс. Режим доступа: http://recaptcha.ru (обращение 26.09.2010)

[2] “Breaking da CAPTCHA theShockwaveRider” Электронный ресурс. Режим доступа: http://xain.hackerdom.ru/zine/online/issue0/Breaking%20Da%20CAPTCHA.html (обращение 14.04.2010)

[3] «CAPTCHA Effectiveness» Электронный ресурс. Режим доступа: http://www.codinghorror.com/blog/2006/10/captcha-effectiveness.html (обращение 26.09.2010)

[4] Шапиро Л., Стокман Дж. Компьютерное зрение; Пер с англ. – М.: БИНОМ. Лаборатория знаний, 2009. – 752 с.

[5] Дэвид А. Форсайт, Джин Понс. Компьютерное зрение. Современный подход. : Пер. с англ. – М.: Издательский дом «Вильямс», 2004. – 928 с.

[6] Корнеев В.Д. Параллельное программирование в MPI. – 2-е изд., испр. – Новосибирск: Изд-во ИВМиМГ СО РАН, 2002. – 215 с.

[7] Круглов В.В., Борисов В.В. Искусственные нейронные сети. Теория и практика. – М.: Горячая линия – Телеком, 2001. – 382 с.