Пошук розмірів еталонів при розпізнаванні текстових зображень

Олефіренко С.А.

Міжнародний науково-навчальний центр
інформаційних технологій та систем, Київ, Україна

Анотація

Запропоновано підхід до оцінки параметрів алгоритму розпізнавання тексту (еталонів букв) на основі навчальної вибірки, яка складається з прикладів зображень текстових рядків та відповідних їм послідовностей літер і не містить сегментації навчальних зображень на зображення окремих літер. Також запропоновано метод пошуку розмірів еталонів літер одночасно із оцінюванням їх значень. Сформульована таким чином загальна задача навчання дозволяє значно спростити пошук еталонів літер, необхідних для розпізнавання заданого джерела зображень тексту, прибравши необхідність в ручному знаходженні розмірів еталонів та виділення на зображенні сегментів кожної букви і залишивши лише необхідність набрати текст, що відповідає навчальним зображенням.

1. Вступ

Робота присвячена навчанню еталонів символів в задачі розпізнавання зображень друкованого тексту. Не дивлячись на солідний вік проблеми і відповідний широкий спектр як методів, так і відповідного програмного забезпечення, задачу розпізнавання тексту навряд чи можна вважати повністю розв'язаною. Так, розпізнавання сильнозашумлених текстів або текстів, що містять літери в нестандартному написанні, досі містить багато труднощів. Більшість з цих труднощів можна було б розв'язати, якби поряд з ефективним методами розпізнавання існували ефективні методи навчання розпізнаванню.

Задача навчання розпізнаванню друкованого тексту полягає переважно в побудові еталонів символів цього тексту. Існує ряд підходів до її розв'язку(див. [1, 2, 3]), але більшість з них потребує вказання для кожного навчального зображення його точної сегментації на зображення окремих символів. Отримання такої сегментації є найбільш складним етапом побудови навчальної вибірки і створює суттєві труднощі для використання таких методів.

В нашій роботі ми пропонуємо інший підхід до навчання, який не потребує сегментацій навчальних зображень. Замість цього навчальна вибірка складається з зображень текстових рядків та відповідних їм послідовностей літер, тобто власне самих текстових рядків. Отримання таких даних потребує набагато менших зусиль (достатньо лише прочитати і записати текст), що означає суттєве зменшення праці по настройці алгоритму розпізнавання на нове джерело тексту.

Крім того, ми пропонуємо підхід до навчання розмірів еталонів символів тексту. Це позбавляє оператора від ручного визначення цих розмірів, що також складає важливу і доволі значну частину ручної праці по підготовці навчальних даних.

Робота складається з 6 розділів. Наступний присвячено основним означенням, третій та четвертий — постановці задач навчання, з фіксованими розмірами еталонів та в загальному вигляді. В п'ятому наведені результати експериментів, а в шостому — висновки.

2. Означення та постановка задачі навчання

Введемо основні позначення, що будуть використовуватись в роботі.

Позначатимемо вхідні зображення через x. Всі вони мають однакову висоту H, але, взагалі кажучи, різну ширину. Вважатимемо, що незашумлене зображення, яке відповідає заданому текстовому рядку, є горизонтальною послідовністю еталонних зображень літер рядка, причому ці зображення не перекриваються, а можливі проміжки між ними заповнюються кольором фону. Будемо вважати, що вхідне зображення відрізняється від незашумленого зображення лише адитивним гаусівським шумом, що накладається в кожному пікселі незалежно від інших.

Скінчену множину A_0 літер тексту називатимемо алфавітом. Послідовність елементів алфавіту $\bar{c} = (c_1, c_2, \ldots, c_L)$, $c_l \in A_0, l = \overline{1, L}$ називатимемо текстовим рядком. За допомогою $L_{\bar{c}}$ позначатимемо надалі довжину рядка $\bar{c}$.

Для формального опису проміжків між зображеннями літер введемо додатковий елемент алфавіту. Назвемо його *вставкою* і позначатимемо κ.

Множину $A_0 \bigcup \{\kappa\}$, що складається з алфавіту A_0 та вставки κ, будемо позначати A, а її елементи $a \in A$ називатимемо *символами*.

З кожним символом $a \in A$ пов'яжемо його еталонне зображення e_a висоти H та ширини $d(a)$, що залежить від літери. Еталон вставки має ширину $d(\kappa) = 1$. Множину еталонних зображень позначатимемо E.

Називатимемо сегментом поіменований прямокутний фрагмент, який містить зображення певного символу. При цьому висота фрагмента збігається з висотою H вхідного зображення, а ширина — з шириною відповідного символу. Таким чином сегмент s визначається координатою свого лівого краю i та символом $a \in A$, зображення якого він містить.

Сегментацією зображення ширини W назвемо послідовність сегментів $\bar{s} = (s_1, \ldots, s_N)$ довільної довжини N, які покривають все зображення і розташовані впритул один до одного:

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

Множину всіх сегментацій позначимо $\bar{S}$.

Сегментація $\bar{s}$ визначає зображення, складене із зображень еталонів, що відповідають символам сегментації, розташованих горизонтально впритул один до одного. Введемо штраф $f(x, \bar{s}, E)$ за сегментацію $\bar{s}$ як суму квадратів відхилень кольорів пікселів такого складаного зображення від пікселів зображення x тих же розмірів. Тоді задача розпізнавання зображення x при заданій множині еталонів E полягає в пошуку сегментації, що має мінімальний штраф:

$$\bar{s}^* = \arg\min_{\bar{s}} f(x, \bar{s}, E).$$

Відомо, що ця задача розв'язується за допомогою динамічного програмування ([1]).

Параметром задачі є множина E, тобто еталонні зображення символів. Метою даної роботи є побудова цих зображень на основі навчальної множини. Кожне зображення визначається своїми розмірами та кольорами пікселів. В роботі [4] вказано спосіб визначення кольорів пікселів еталонів за умови відомих значень їх розмірів. Короткий опис метода наведено нижче, а основна частина роботи присвячена визначенню розмірів еталонів та загальному алгоритму навчання, що складається з цих двох методів.

3. Побудова еталонів символів відомих розмірів

Сформулюємо задачу побудови еталонів символів відомих розмірів на підставі лише вказаних текстових рядків для навчальних зображень.

Нехай задана навчальна множина D, що складається із вхідних зображень $x^1, x^2, \ldots, x^M$ та відповідних їм текстових рядків $\bar{c}^1, \bar{c}^2, \ldots, \bar{c}^M$. Нехай $p(D; E)$ — ймовірність навчальної множини, параметризована відомими еталонними зображеннями E (в нашій роботі ця ймовірність визначалась як добуток гаусівських відхилень значень пікселів всіх зображень від значень відповідних пікселів еталонів). Задача побудови еталонів полягає в пошуку таких значень E^*, які максимізують ймовірність $p(D; E)$:

$$E^* = \arg\max_E p(D; E). \quad (1)$$

В роботі [4] для її розв'язку використаний ітеративний алгоритм самонавчання [5, 6]. Там же наведені практичні приклади застосування алгоритму.

4. Пошук розмірів еталонів

Використання задачі (1) на практиці обмежується необхідністю задавати розміри еталонів. В нашій постановці висота еталонів відома і дорівнює висоті вхідних зображень, але ширини еталонів невідомі. Ручне визначення цих ширин є неприроднім і трудомістким. В цьому розділі ми модифікуємо задачу (1) для автоматичної настройки ширин еталонів разом з їх розкраскою.

Нехай D — навчальна множина, $\bar{w} = \{d(\alpha) | \, \alpha \in A\}$ — вектор, що містить ширини всіх символів; $E(\bar{w})$ — еталони символів, ширини яких визначаються вектором $\bar{w}$. Необхідно знайти такий набір еталонів, що визначається ширинами $\bar{w}^*$ і розкраскою $E^*(\bar{w}^*)$, які максимізують $p(D; \bar{w}, E(\bar{w}))$ — ймовірність навчальної множини, параметризовану значеннями еталонів:

$$(\bar{w}^*, E^*(\bar{w}^*)) = \arg\max_{(\bar{w}, E(\bar{w}))} p(D; \bar{w}, E(\bar{w})). \quad (2)$$

Розв'язок задачі в такому вигляді нам невідомий. Ми пропонуємо наближений алгоритм її розв'язку. Він складається в ітеративному розбитті (2) на дві підзадачи: пошуку оптимальної розкраски еталонів при фіксованих їх розмірах (задача (1)) і пошуку оптимальних розмірів еталонів з використанням заданої розкраски.

Отриманий ітеративний алгоритм на кожній ітерації виконує наступні три дії:

Алгоритм 1 Ітеративний алгоритм пошуку ширин еталонів разом з їх розкраскою

1: Знайти еталони $E^*(\bar{w}^0)$ з фіксованими поточними ширинами $\bar{w}^0$ за допомогою алгоритму, що розв'язує задачу (1):

$$E^*(\bar{w}^0) = \arg\max_{E(\bar{w}^0)} p(D; E(\bar{w}^0)). \qquad (3)$$

2: Розпізнати навчальні зображення x^m, $m = \overline{1, M}$ з використанням побудованих еталонів $E^*(\bar{w}^0)$, і отримати сегментації навчальних зображень:

$$\bar{s}^m = \arg\min_{\bar{s}\in\bar{S}} f(x^m, \bar{s}, E^0(\bar{w}^0)),\ m = \overline{1, M}. \quad (4)$$

3: Покращити оцінку ширин $\bar{w}^*$ на підставі попередніх значень ширин і отриманих сегментацій:

$$\left\{\bar{w}^0, x^m, \bar{c}^m, \bar{s}^m \,|\, m = \overline{1, M}\right\} \to \bar{w}^*. \qquad (5)$$

Основний інтерес містить третя задача, оскільки перші дві мають відомий розв'язок. За відсутністю простору лише коротко неформально опишемо постановку задачі (5). Будемо вважати, що сегментація навчального зображення x, отримана на кроці 2, визначає положення літер на ньому. Положення літери будемо ототожнювати з положенням її центра. Для довільного набору ширин еталонів літер їх положення на навчальному зображенні визначають оцінку максимальної правдоподібності розкраски еталонів. (В нашому випадку така оцінка — результат усереднення усіх фрагментів навчального зображення, що відповідають літері.) Оцінені еталони, в свою чергу, визначають штраф за подібність побудованої таким чином нової сегментації (що визначається центрами літер та їх новими ширинами) до навчального зображення. Задача полягає в пошуку таких ширин еталонів, що мінімізують цей штраф.

Ми пропонуємо метод розв'язку даної задачі, що полягає в зведенні її до одної з задач пошуку оптимальної розмітки [7, 8], що має розв'язок. Кожній літері алфавіту в цій новій задачі відповідає пара вершин графа, що визначає задачу розмітки (для лівої та правої половини літери). Мітки вершин задають ширини еталонів відповідних половин літер, а найкраща розмітка — оптимальні в певному сенсі ширини еталонів.

5. Результати експериментів

На вхід програми подавались дві множини: навчальна, що містила зображення текстових рядків і відповідні їм "правильні" текстові рядки, і тестова, що містила зображення із того ж джерела. На

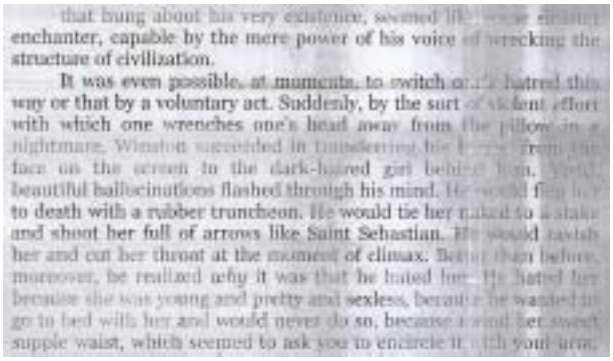

Рис. 1: *Приклад 1. Зразок навчальних зображень.*

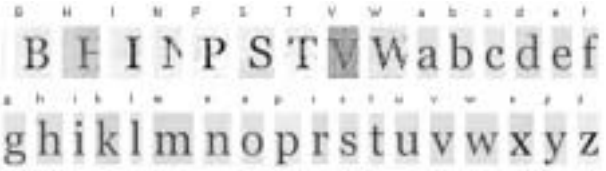

Рис. 2: *Приклад 1. Побудовані еталони.*

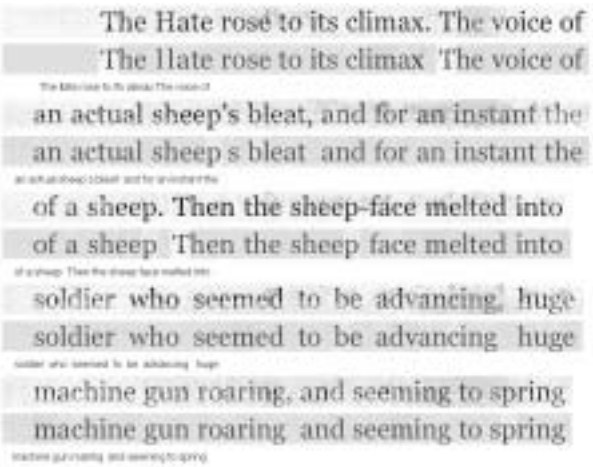

Рис. 3: *Приклад 1. Результати розпізнавання.*

кожному кроці роботи програми спочатку виконувався пошук оптимальних ширин при поточних еталонах згідно задачи (5), а потім — розкраска еталонів при поточних ширинах згідно задачи (3).

В якості початкової ширини еталона обиралось значення, що кратне висоті зображення (а саме $H/4$). Таке занадто мале значення пов'язано з тим, що алгоритм самонавчання набагато гірше реагує на ширини, що більші за істинні, ніж на ширини, що є меншими за істинні. Причина цього знаходиться у постановці задачі, де не дозволяється накладання еталонів один на одного.

Тут ми розглянемо два приклади, обидва є реальними зображеннями з певних джерел. Перший приклад — це зображення відсканованої сторін-

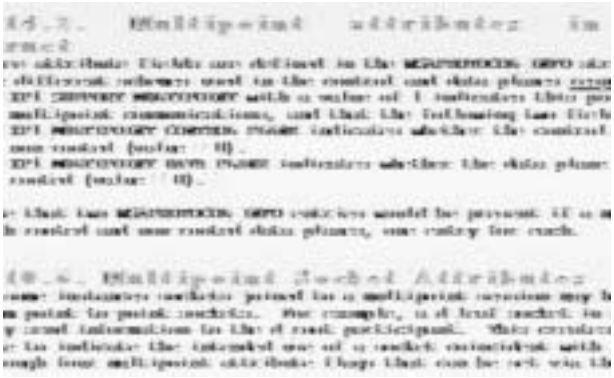

Рис. 4: *Приклад 2. Зразок навчальних зображень.*

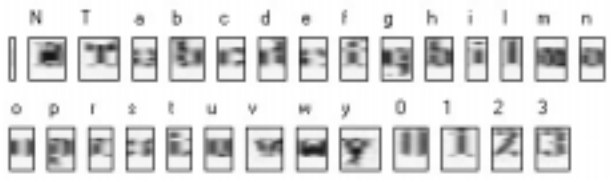

Рис. 5: *Приклад 2. Побудовані еталони.*

ки тексту, роздрукованої на дефектному лазерному принтері. Як ми бачимо (рис. 1), зображення має суттєвий рівень шуму, який до того ж не є рівномірним, що виходить за рамки прийнятої в роботі моделі. Відмітимо, що комерційні пакети, якщо і не відмовляються від розпізнавання цього прикладу, видають результати з неприйнятним рівнем помилки. В нашому випадку помилка розпізнавання тестової вибірки склала $2,5\%$, що є ще доволі високим значенням, але вже дозволяє використовувати, наприклад, словарну корекцію тексту. Еталони всіх літер (рис. 2), крім деяких прописних (що зустрічаються в навчальній вибірці лише один раз) побудовані вірно. На рис. 3 можна побачити результати розпізнавання тестової вибірки (зображення та відповідні їм найкращі сегментації).

Інший приклад, який ми наведемо, також є реальним зображенням з сильною деградацією (рис. 4). Результати навчання наведено на рис. 5. Помилка розпізнавання тестової вибірки склала $2,7\%$, причому більше половини помилок виникло із-за того, що необхідні літери не були представлені в навчальній вибірці.

6. Висновки

Запропонований підхід до навчання алгоритму розпізнавання тексту на нове джерело дозволяє уникнути майже всієї трудомісткої та рутинної праці і тим самим максимально наближає його до практичного використання. Помилка при розпізнаванні складає в середньому $0,8 - 1,5\%$ для зображень, характер шуму яких відносно добре описується прийнятою в роботі моделлю, та до $2 - 3\%$ для сильно зашумлених зображень або зображень, характер шуму яких значно відрізняється від моделі.

Крім самостійного застосування, алгоритм є доволі ефективним в допоміжній ролі — побудові точних сегментацій зображень для використання їх в навчальних вибірках алгоритмів, що потребують ці сегментації, але забезпечують кращу якість розпізнавання за рахунок використання більш складних моделей зображення (напр. настройка [3]).

7. Література

[1] В. А. Ковалевский. Оптимальный алгоритм распознавания некоторых последовательностей изображений. *Кибернетика*, (4):75–80, 1967.

[2] G. E. Kopec. Multilevel Character Templates for Document Image Decoding. In *IS&T/SPIE 1996 Intl. Symposium on Electronic Imaging: Science & Technology*, San Jose, CA, Jan. 27–Feb. 2, 1996.

[3] Савчинський Б.Д., Камоцький О.В. Настройка алгориту розпізнавання тексту // *Управляющие системы и машины.*— 2005.— №2.— С. 17–24.

[4] Б. Д. Савчинский, С. А. Олефиренко. Обучение с неполной информацией от учителя для распознавания текстовых изображений // *Управляющие системы и машины*, 2007(1).— С. 19–29.

[5] М. И. Шлезингер. Взаимосвязь обучения и самообучения в распознавании образов. *Кибернетика*, (2):81–88, 1968.

[6] М. Шлезингер, В. Главач. Десять лекций по статистическому и структурному распознаванию.— Киев: Наукова думка, 2004.— С. 545.

[7] М. И. Шлезингер. Синтаксический анализ двумерных зрительных сигналов в условиях помех. *Кибернетика*, 4:113–130, 1976.

[8] М. И. Шлезингер, В. К. Коваль. Двумерное программирование в задачах анализа изображений. *Автоматика и телемеханика, М.*, 2:149–168, 1976.