Навчання з неповною інформацією від вчителя при розпізнаванні текстових зображень *

Савчинський Б.Д., Олефіренко С.А.

Вступ

Задача розпізнавання тексту, незважаючи на свою популярність та уявну вивченість, все ще містить значний простір для наукових досліджень. Одночасно розпізнавання текстового рядка є яскравим і, напевно, одним з найпростіших прикладів в структурному розпізнаванні зображень. Саме з цієї причини ця задача є зручним полігоном для впровадження та експериментальної перевірки нових методів в структурному розпізнаванні.

Однією з галузей структурного розпізнавання, яка, на наш погляд, потребує значних досліджень, є навчання та оцінка параметрів розпізнаючих алгоритмів на основі навчальної вибірки. Як відомо, навчальна вибірка містить певну кількість зображень та відповідних їм результатів розпізнавання. У випадку розпізнавання зображення текстового рядка результатом є не лише послідовність літер, що йому відповідає, але й сегментація цього рядка на зображення окремих літер. При цьому до сегментації висуваються зазвичай доволі жорсткі вимоги: однакові літери в межах різних сегментів, що їх містять, повинні бути однаковим чином відцентровані. Це означає, що координати відповідних пікселів однакових літер на різних сегментах мають бути однаковими. Побудова такої сегментації потребує значних зусиль та часу вчителя.

В даній роботі ми пропонуємо постановку та алгоритм розв'язку задачі оцінки параметрів алгоритму розпізнавання текстового рядка на основі навчальної вибірки, що містить лише приклади зображень текстових рядків та відповідних їм послідовностей літер і не містить сегментації зображень на зображення окремих літер. Таке формулювання задачі навчання дозволяє значно спростити та прискорити побудову навчальної вибірки, звівши її до простого набору тексту, що відповідає навчальним зображенням.

Робота складається з чотирьох розділів, перший з яких присвячено основним означенням та постановці задачі, другий – її розв'язанню, третій та четвертий відповідно – експериментальній перевірці алгоритмів та висновкам.

*Робота виконувалась в рамках проекту EU INTAS PRINCESS 04-77-7347

1 Означення та постановка задачі навчання

Введемо основні позначення, що будуть використовуватись в роботі.

Полем зору T назвемо прямокутну підмножину двовимірної цілочисельної решітки — множину координат пікселів зображення:

$$T = \{\, (i,j) \mid i = \overline{0, W-1}, j = \overline{0, H-1} \,\}.$$

Величину W називатимемо шириною, а H — висотою поля зору. Елементи $t \in T$ поля зору вважатимемо звичайними двовимірними векторами, зокрема визначена операція додавання двох елементів поля зору як покомпонентне додавання відповідних координат пікселів зображення.

Нехай V — множина значень яскравості піксела. *Зображенням* назвемо функцію $x\colon T \to V$, його висота та ширина збігаються відповідно з висотою та шириною поля зору. Ми розглядатимемо два типи зображень: зображення, подані на розпізнавання, та еталонні зображення літер, про які йтиметься далі в цьому розділі. Вважатимемо, що всі ці зображення мають однакову висоту H, але, взагалі кажучи, різну ширину.

Скінчену множину A_0 називатимемо алфавітом. Елементами алфавіту є літери тексту. Пропуск між словами в тексті вважатимемо за окрему літеру, що також належить до алфавіту. Послідовність елементів алфавіту $\bar{k} = (k_1, k_2, \ldots, k_L)$, $k_l \in A_0, l = \overline{1, L}$ називатимемо текстовим рядком. За допомогою $L_{\bar{k}}$ позначатимемо надалі довжину рядка $\bar{k}$.

З кожною літерою $k \in A_0$ пов'яжемо її еталонне зображення e_k, визначене на полі зору висоти H та ширини $d(k)$, що залежить від літери. Ширини $d(k), k \in A_0$ еталонних зображень всіх літер вважатимемо фіксованими і відомими. Множину еталонних зображень позначатимемо E.

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.

Для формального опису проміжків між зображеннями літер введемо додатковий елемент алфавіту. Назвемо його *вставкою* і позначатимемо κ. Вважатимемо, що еталон вставки має ширину $d(\kappa) = 1$, висоту H і належить до множини еталонних зображень E. 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*. Таким чином символ $a \in A$ є або ж літерою алфавіту $k \in A_0$, або ж вставкою κ.

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 $i = \overline{0, W - d(a)}$ та символом $a \in A$, зображення якого він містить. Координату лівого краю сегмента $s = (i, a)$ позначатимемо $i(s)$, а пов'язаний з ним символ — $a(s)$. Множину всіх сегментів позначатимемо S. З кожним сегментом пов'яжемо елемент t_s поля зору зображення, координати якого співпадають з координатами лівого краю сегмента: $t_s = (i, 0)$.

Сегментацією зображення назвемо послідовність сегментів $\bar{s} = (s_1, \ldots, s_N)$ довільної довжини

N, які покривають все поле зору 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 $\bar{S}$.

Вважатимемо, що вхідне зображення відрізняється від незашумленого зображення, процес побудови якого з послідовності символів описано вище, лише гаусівським шумом визначеної дисперсії σ, що накладається в кожному пікселі незалежно від інших. Таким чином, ймовірнісний розподіл $p(x|\,\bar{s}; E)$ зображення x за умови відомих сегментацій $\bar{s}$ та множини еталонів E приймає вигляд

$$p(x|\,\bar{s}; E) = \prod_{n=1}^{N(\bar{s})} p(x|\,s_n; E) = \prod_{n=1}^{N(\bar{s})} \prod_{t \in T(s_n)} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{(x(t_{s_n} + t) - e_{a(s_n)}(t))^2}{2\sigma^2} \right\}, \qquad (1)$$

де за допомогою $N(\bar{s})$ позначено кількість сегментів в сегментації $\bar{s}$, а за допомогою $T(s)$ — прямокутний фрагмент поля зору, який відповідає сегменту s.

Задача розпізнавання зображення x полягає в пошуку найімовірнішої сегментації $\bar{s}^*$:

$$\bar{s}^* = \arg\max_{\bar{s}} p(x, \bar{s}; E) = \arg\max_{\bar{s}} p(\bar{s}) \cdot p(x|\,\bar{s}; E).$$

Як відомо [1], вона розв'язується за допомогою алгоритму динамічного програмування.

Навчання алгоритму розпізнавання, якому власне присвячена дана робота, полягає в оцінці значень параметрів цього алгоритму, якими є множина еталонів E та апріорний розподіл сегментацій $\{p(\bar{s}) \mid \bar{s} \in \bar{S}\}$, на основі навчальної вибірки.

Перш ніж перейти до формулювання задачі, відзначимо зв'язок між сегментаціями та послідовностями літер алфавіту. Довільній сегментації $\bar{s} = (s_1, \ldots, s_N)$ відповідає послідовність символів $\bar{a}(\bar{s}) = (a_1, \ldots, a_N | a_n = a(s_n), n = \overline{1, N})$. В свою чергу, цій послідовності символів відповідає послідовність літер, що отримується з неї видаленням всіх вставок. Таким чином, будь-яка послідовність літер $\bar{k} = (k_1, \ldots, k_N), k_n \in A_0$ пов'язана з множиною $\bar{S}(\bar{k})$ всіх сегментацій таких, що відповідні їм текстові рядки після видалення всіх вставок збігаються з $\bar{k}$.

Перейдемо до формулювання задачі навчання. Нехай

$$D = \begin{pmatrix} x^1 & x^2 & \ldots & x^M \\ \bar{k}^1 & \bar{k}^2 & \ldots & \bar{k}^M \end{pmatrix} -$$

навчальна вибірка, що складається з M вхідних зображень і M відповідних їм текстових рядків.

Ймовірність $p(x, \bar{k}; E)$ пари $(x, \bar{k})$ зображення x та текстового рядка $\bar{k}$, рівна сумарній ймовірності $\sum_{\bar{s} \in \bar{S}(\bar{k})} p(x, \bar{s}; E)$ усіх сегментацій, що відповідають рядку $\bar{k}$. Ймовірність вибірки $p(D, E)$ таким чином приймає вигляд

$$p(D; E) = \prod_{m=1}^{M} p(x^m, \bar{k}^m; E) = \prod_{m=1}^{M} \sum_{\bar{s} \in \bar{S}(\bar{k}^m)} p(x^m, \bar{s}; E) = \prod_{m=1}^{M} \sum_{\bar{s} \in \bar{S}(\bar{k}^m)} p(\bar{s}) \cdot p(x^m|\,\bar{s}; E).$$

Задача 1 *Задача навчання алгоритму розпізнавання полягає у знаходженні таких еталонів E^* та апріорних ймовірностей сегментацій $p^*(\bar{s})$, які максимізують ймовірність вибірки D:*

$$(E^*, p^*(\bar{s})) = \arg\max_{(E,p(\bar{s}))} \prod_{m=1}^{M} \sum_{\bar{s}\in\bar{S}(\bar{k}^m)} p(\bar{s}) \cdot p(x^m|\,\bar{s}; E) =$$

$$= \arg\max_{(E,p(\bar{s}))} \prod_{m=1}^{M} \sum_{\bar{s}\in\bar{S}(\bar{k}^m)} \frac{p(\bar{s})}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\sum_{n=1}^{N(\bar{s})} \sum_{t\in T(s_n)} \frac{(x^m(t_{s_n}+t) - e_{a(s_n)}(t))^2}{2\sigma^2} \right\}. \quad (2)$$

Нам невідомий алгоритм точного розв'язку задачі 1. В даній роботі для пошуку розв'язку було використано алгоритм самонавчання, описаний в [1]. Як відомо, цей алгоритм, взагалі кажучи, забезпечує пошук лише локального екстремуму. Водночас, з практичної точки зору це не становить значної проблеми, оскільки якість оцінки параметрів може бути легко проконтрольована візуально, а на основі результатів розпізнавання зображень з вибірки може бути оцінено якість розпізнавання зображень, що не увійшли до неї.

Водночас, алгоритм самонавчання, описаний в [1], не може бути використаний для вирішення задачі 1 безпосередньо, оскільки потребує експоненційних за розмірами вхідних зображень часу та пам'яті. В наступному розділі описано, яким чином цей алгоритм повинен бути реалізований для свого ефективного використання при розв'язанні задачі 1.

2 Розв'язок задачі навчання

Спочатку сформулюємо для задачі 1 алгоритм самонавчання у тому вигляді, як він описаний в [1]. Як вже було сказано, в такому вигляді він не може бути використаний безпосередньо через свою значну ресурсоємкість. Після цього перетворимо його еквівалентним чином, суттєво знизивши його ресурсоємкість, залишивши при цьому без змін суть виконаних операцій.

2.1 Базовий алгоритм самонавчання

Перш за все, введемо додаткові позначення. Для множини $\bar{S}(\bar{k}^m)$ сегментацій, символьні рядки яких після видалення всіх вставок збігаються з m-тим рядком з навчальної вибірки $\bar{k}^m,\ m = \overline{1, M}$, введемо еквівалентне позначення $\bar{S}_m$. Через $\bar{S}_m(s)$ позначатимемо підмножину множини $\bar{S}_m$, що складається лише із тих сегментацій, які містять сегмент s.

Алгоритм самонавчання є ітеративним. За допомогою верхнього індекса r позначатимемо значення величин, які вони приймають на r-ій ітерації алгоритму. Нехай $E^0, p^0(\bar{s}),\ \bar{s} \in \bar{S}$ — відповідно початкові значення еталонів символів та апріорний розподіл сегментацій зображення. Кожна ітерація алгоритму складається з двох кроків. На першому кроці (розпізнавання) обчислюються оцінки $\hat{\alpha}^r(x^m, \bar{s}),\ m = \overline{1, M}, \bar{s} \in \bar{S}_m$ апостеріорних розподілів сегментацій $\bar{s}$ для кожного навчального зображення x^m:

$$\hat{\alpha}^r(x^m, \bar{s}) = \frac{p^r(\bar{s}) \cdot p(x^m|\bar{s}; E^r)}{\sum\limits_{\bar{s}\in\bar{S}_m} p^r(\bar{s}) \cdot p(x^m|\bar{s}; E^r)}, \quad m = \overline{1, M},\ \bar{s} \in \bar{S}_m. \quad (3)$$

На другому кроці (навчання) оцінюються апріорні ймовірності сегментацій $p^{r+1}(\bar{s}),\ \bar{s} \in \bar{S}$ та еталонні зображення символів E^{r+1} згідно формул:

$$p^{r+1}(\bar{s}) = \frac{\sum\limits_{m=1}^{M} \hat{\alpha}^r(x^m, \bar{s})}{M}, \quad \bar{s} \in \bar{S}; \tag{4}$$

$$E^{r+1} = \arg\max_{E} \sum_{m=1}^{M} \sum_{\bar{s} \in \bar{S}_m} \hat{\alpha}^r(x^m, \bar{s}) \cdot \log p(x^m|\bar{s}; E^r). \tag{5}$$

2.2 Ефективна реалізація алгоритму самонавчання

Реалізація алгоритму самонавчання у вигляді (3)–(5) неможлива, оскільки величини $\hat{\alpha}^r(x^m, \bar{s})$ повинні бути обчислені для всіх можливих сегментацій зображень вибірки, кількість яких зростає експоненційно з розмірами зображень. Модифікація алгоритму, що лежить в основі ефективної реалізації, полягає в ітеративному обчисленні величин, пов'язаних з окремими сегментами, а не з сегментаціями в цілому, як це має місце в базовому алгоритмі.

Вважатимемо, що окремі сегменти будь-якої сегментації є незалежними один від одного, тобто що ймовірність сегментації $\bar{s}$ визначається формулою:

$$p(\bar{s}) = \prod_{n=1}^{N(\bar{s})} p(s_n), \tag{6}$$

де $p(s),\ s \in S$ — апріорні ймовірності сегментів. Очевидна також наступна рівність:

$$p(s) = \sum_{\bar{s} \in \bar{S}_m(s)} p(\bar{s}), \quad s \in S. \tag{7}$$

Замість величин $\hat{\alpha}(x, \bar{s})$, що відповідають апостеріорним ймовірностям сегментацій зображення x, введемо величини $\alpha(x, s),\ s \in S$, які є оцінками апостеріорних ймовірностей окремих сегментів:

$$\alpha^r(x^m, s) = \sum_{\bar{s} \in \bar{S}_m(s)} \hat{\alpha}^r(x^m, \bar{s}) = \frac{\sum\limits_{\bar{s} \in \bar{S}_m(s)} p^r(\bar{s}) \cdot p(x^m|\bar{s}; E^r)}{\sum\limits_{\bar{s} \in \bar{S}_m} p^r(\bar{s}) \cdot p(x^m|\bar{s}; E^r)}, \quad m = \overline{1, M},\ s \in S. \tag{8}$$

Підставивши (1) та (6) у (8), отримаємо:

$$\alpha^r(x^m, s) = \frac{\sum\limits_{\bar{s} \in \bar{S}_m(s)} \prod\limits_{n=1}^{N(\bar{s})} p^r(s_n) \cdot p(x^m|\, s_n; e^r_{a(s_n)})}{\sum\limits_{\bar{s} \in \bar{S}_m} \prod\limits_{n=1}^{N(\bar{s})} p^r(s_n) \cdot p(x^m|\, s_n; e^r_{a(s_n)})}, \quad m = \overline{1, M},\ s \in S. \tag{9}$$

Обчислення за формулою (9) не можуть бути зроблені безпосередньо, але далі в підрозділі 2.3 буде наведено ефективний алгоритм таких обчислень, що базується на методі динамічного програмування.

Перейдемо до формули (4). Для довільного фіксованого сегмента $s \in S$ просумуємо (4) за всіма сегментаціями, які його містять. Тоді згідно (7) та (8) отримаємо:

$$p^{r+1}(s) = \frac{\sum\limits_{m=1}^{M} \alpha^r(x^m, s)}{M}, \quad s \in S. \tag{10}$$

Величини $p^{r+1}(s)$ можуть бути обчислені безпосередньо за даною формулою.

Множина еталонів E складається із значень усіх пікселів еталонів усіх символів: $E = \{e_a(t) \mid t \in T_a, a \in A\}$. Максимум суми (5) визначається системою рівностей:

$$\begin{cases} \frac{\partial}{\partial e_a(t)} \sum_{m=1}^{M} \sum_{\bar{s} \in \bar{S}_m} \hat{\alpha}^r(x^m, \bar{s}) \cdot \log p(x^m | \bar{s}; E^r) = 0, \\ t \in T_a, a \in A. \end{cases}$$

Підставивши величини $p(x^m|\bar{s}; E^r)$ згідно (1) за допомогою нескладних алгебраїчних перетворень послідовно отримаємо:

$$\frac{\partial}{\partial e_a(t)} \sum_{m=1}^{M} \sum_{\bar{s} \in \bar{S}_m} \hat{\alpha}^r(x^m, \bar{s}) \sum_{n=1}^{N(\bar{s})} \sum_{t \in T(s_n)} \frac{(x^m(t(s_n)+t) - e_{a(s_n)}(t))^2}{2\sigma^2} = 0,$$

$$\sum_{m=1}^{M} \sum_{\bar{s} \in \bar{S}_m} \hat{\alpha}^r(x^m, \bar{s}) \sum_{n=1}^{N(\bar{s})} \mathbf{1}_{\{a(s_n)=a\}} \cdot (e_a(t) - x^m(t_{s_n}+t)) = 0,$$

$$\sum_{m=1}^{M} \sum_{\substack{s \in S: \\ a(s)=a}} \sum_{\bar{s} \in \bar{S}_m(s)} \hat{\alpha}^r(x^m, \bar{s}) \cdot (e_a(t) - x^m(t_s+t)) = 0,$$

$$\sum_{m=1}^{M} \sum_{\substack{s \in S: \\ a(s)=a}} \alpha^r(x^m, s) \cdot (e_a(t) - x^m(t_s+t)) = 0.$$

З останньої формули видно, що точка екстремуму — єдина і в цій точці досягається глобальний максимум суми (5). Тому оптимальний еталон для довільного символу $a \in A$ будується зваженим усередненням за всіма сегментами з тим же іменем a відповідних фрагментів зображень з вагами $\alpha(x^m, s)$:

$$e_a^{r+1}(t) = \frac{\sum_{\substack{s \in S: \\ a(s)=a}} \sum_{m=1}^{M} \alpha^r(x^m, s) \cdot x^m(t_s+t)}{\sum_{\substack{s \in S: \\ a(s)=a}} \sum_{m=1}^{M} \alpha^r(x^m, s)}, \quad t \in T_a. \tag{11}$$

Таким чином, від базового алгоритму (3)–(5) ми перейшли до алгоритму, який розв'язує ту саму задачу, але оцінює іншу множину параметрів, що складається з еталонів всіх символів та апріорних ймовірностей сегментів: $\{e_a, p(s) \mid a \in A, s \in S\}$. На відміну від базового отриманий алгоритм (9)–(11) може бути ефективно реалізований: величини $p(s),\ s \in S$ та еталони символів $e_a,\ a \in A$ можуть бути обчислені безпосередньо згідно формул (10) та (11), а алгоритм обчислення величин $\alpha^r(x, s),\ s \in S$ згідно (9) наведено в наступному підрозділі.

2.3 Алгоритм обчислення апостеріорних ймовірностей сегментів $\alpha(x, s)$

В даному розділі ми розглянемо алгоритм обчислення апостеріорних ймовірностей сегментів $\alpha(x, s)$ для довільного зображення x та відповідного йому текстового рядка $\bar{k}$. Позначимо довжину рядка $\bar{k}$ як L: $\bar{k} = (k_1, \ldots, k_l, \ldots, k_L)$. Ймовірність $p(s) \cdot p(x|\, s; e_{a(s)})$ сегмента $s = (i, a)$ називатимемо штрафом за цей сегмент і позначатимемо $f(i, a)$. Штрафом за сегментацію $\bar{s}$ називатимемо добуток штрафів за сегменти, з яких вона складається. Розрізнятимемо сегменти, іменем яких є літера з множини A_0 — такі сегменти ми називатимемо *значущими* — та сегменти, іменем яких є вставка.

Нас цікавитимуть лише сегментації, що відповідають рядку $\bar{k}$, тобто сегментації з множини $\bar{S}(\bar{k})$, яку ми для простоти подальшого запису позначатимемо $\bar{S}^\circ$. Такі сегментації містять L значущих сегментів, які відповідають літерам рядка, та довільну кількість вставок. Введемо функцію $\mathrm{sgf}\colon \{1,\dots,L\} \times \bar{S}(\bar{k}) \to S$, значення якої $\mathrm{sgf}(l,\bar{s})$ вказує на l-ий значущий сегмент сегментації $\bar{s}$, тобто на сегмент, який відповідає літері з порядковим номером l рядка $\bar{k}$. Крім цього, введемо дві функції $b_L(l,\bar{s})$ та $b_R(l,\bar{s})$, $l = \overline{1,L}$, які приймають значення, відповідно, координат лівого та правого країв l-го значущого сегмента.

Для довільної сегментації $\bar{s}$ сегмент з координатою лівого краю 0 називатимемо початком сегментації, а сегмент з координатою правого краю W — її кінцем.

Як видно з формули (9), для довільного сегмента $s = (i,a)$ величина $\alpha(x,s)$ є відношенням сумарного штрафу за всі сегментації, що містять сегмент s, до сумарного штрафу за всі сегментації. Перша з цих сум може бути розбита на добуток двох сум: суми штрафів за частини сегментацій від початку до сегмента s і суми штрафів за частини сегментацій від сегмента s до кінця. Розрізнятимемо два випадки обчислення величини $\alpha(x,s)$: випадок, коли s є значущим сегментом, і випадок, коли s відповідає сегменту-вставці. У першому випадку чисельник формули (9) набуває вигляду:

$$\sum_{\bar{s}\in\bar{S}^\circ(s)} \prod_{n=1}^{N(\bar{s})} p(s_n)\cdot p(x|\, s_n; e_{a(s_n)}) = \sum_{\substack{1\le l\le L:\\ k_l=a}} \sum_{\substack{\bar{s}\in\bar{S}^\circ(s)\\ \mathrm{sgf}(l,\bar{s})=s}} f(i_1,a_1)\cdot f(i_2,a_2)\cdot\ldots\cdot f(i_{N(\bar{s})},a_{N(\bar{s})}) = \tag{12}$$

$$= \sum_{\substack{1\le l\le L:\\ k_l=a}} \left[\sum_{\substack{\bar{s}\in\bar{S}^\circ(s)\\ \mathrm{sgf}(l,\bar{s})=s}} f(i_1,a_1)\cdot\ldots\cdot f(i,a)\right] \times \left[\sum_{\substack{\bar{s}\in\bar{S}^\circ(s)\\ \mathrm{sgf}(l,\bar{s})=s}} f(i,a)\cdot\ldots\cdot f(i_{N(\bar{s})},a_{N(\bar{s})})\right] \times \frac{1}{f(i,a)},$$

де $\bar{s} = (s_1, s_2, \dots, s_{N(\bar{s})}), s_n = (i_n, a_n)$.

Вирази у квадратних дужках залежать від l та сегмента $s = (i, k_l)$. Позначимо перший з них через $F_1(i + d(k_l), l)$, а другий — через $B_1(i,l)$, де $i + d(k_l)$ — координата правого краю сегмента s:

$$F_1(i,l) = \sum_{\substack{\bar{s}\in\bar{S}^\circ(s):\\ b_R(l,\bar{s})=i}} f(i_1,a_1)\cdot f(i_2,a_2)\cdot\ldots\cdot f(i-d(k_l),k_l), \tag{13}$$

$$B_1(i,l) = \sum_{\substack{\bar{s}\in\bar{S}^\circ(s):\\ b_L(l,\bar{s})=i}} f(i,k_l)\cdot\ldots\cdot f(i_{N(\bar{s})},a_{N(\bar{s})}) \tag{14}$$

Для довільної пари (i,l) розглянемо всі сегментації, в яких l-им значущим сегментом є сегмент $s(i,k_l)$. Тоді $F_1(i + d(k_l), l)$ дорівнює сумарному штрафу за частини цих сегментацій від початку до сегмента s включно; $B_1(i,l)$ дорівнюють сумарному штрафу за частини цих сегментацій від сегмента s включно до кінця.

Аналогічно трансформуємо формулу (9) для випадку, коли $s = (i,a)$ відповідає сегменту-вставці

$(a(s) = \kappa)$:

$$\sum_{\bar{s}\in\bar{S}^\circ(s)}\prod_{n=1}^{N(\bar{s})} p(s_n)\cdot p(x|\, s_n; e_{a(s_n)}) = \sum_{l=0}^{L} \sum_{\substack{\bar{s}\in\bar{S}^\circ(s)\\ b_R(l,\bar{s})\le i,\\ b_L(l+1,\bar{s})\ge i+1}} f(i_1,a_1)\cdot f(i_2,a_2)\cdot\ldots\cdot f(i_{N(\bar{s})},a_{N(\bar{s})}) = \quad (15)$$

$$= \sum_{l=0}^{L}\left[\sum_{\substack{\bar{s}\in\bar{S}^\circ(s)\\ b_R(l,\bar{s})\le i,\\ b_L(l+1,\bar{s})\ge i+1}} f(i_1,a_1)\cdot\ldots\cdot f(i,a)\right]\times\left[\sum_{\substack{\bar{s}\in\bar{S}^\circ(s)\\ b_R(l,\bar{s})\le i,\\ b_L(l+1,\bar{s})\ge i+1}} f(i,a)\cdot\ldots\cdot f(i_{N(\bar{s})},a_{N(\bar{s})})\right]\times\frac{1}{f(i,a)}.$$

Позначимо перший вираз у квадратних дужках через $F_0(i+1,l)$, а другий — через $B_0(i,l)$:

$$F_0(i,l) = \sum_{\substack{\bar{s}\in\bar{S}^\circ(s):\\ b_R(l,\bar{s})\le i-1,\\ b_L(l+1,\bar{s})\ge i}} f(i_1,a_1)\cdot f(i_2,a_2)\cdot\ldots\cdot f(i-1,\kappa), \quad (16)$$

$$B_0(i,l) = \sum_{\substack{\bar{s}\in\bar{S}^\circ(s):\\ b_R(l,\bar{s})\le i,\\ b_L(l+1,\bar{s})\ge i+1}} f(i,k_l)\cdot\ldots\cdot f(i_{N(\bar{s})},a_{N(\bar{s})}), \quad (17)$$

де для однотипності позначень покладається $b_R(0,\bar{s}) = 0,\ b_L(L+1,\bar{s}) = W$.

Розглянемо всі сегментації, в яких присутній сегмент $s = (i,\kappa)$, перед яким зустрілось рівно l значущих сегментів. $F_0(i+1,l)$ дорівнює сумарному штрафу за частини сегментацій від початку до сегменту s включно; $B_0(i,l)$ дорівнює сумарному штрафу за частини сегментацій від сегмента s включно до кінця.

Оскільки останнім (з координатою правого краю рівною W) сегментом будь-якої сегментації може бути або L-й значущий сегмент, або ж вставка, перед якою зустрілись всі L значущих сегментів, то сумарний штраф за всі сегментації (знаменник формули (9)) дорівнює

$$Z = \sum_{\bar{s}\in\bar{S}^\circ}\prod_{n=1}^{N(\bar{s})} p(s_n)\cdot p(x|\, s_n; e_{a(s_n)}) = F_0(W,L) + F_1(W,L). \quad (18)$$

Подібними ж міркуваннями можна виразити Z через величини B_0 та B_1:

$$Z = B_0(0,0) + B_1(0,0).$$

Величини F_0 та F_1 можуть розглядатись як суми штрафів за шляхи на деякому графі. Розглянемо граф, вершинами якого є координати стовпчиків $i = \overline{0,W}$ поля зору зображення, а ребра відповідають сегментам на цьому зображенні: сегменту $s = (i,a)$ відповідає ребро $(i \to i + d(a))$ з іменем a, яке ми позначатимемо $\varepsilon(i, i+d(a), a)$. При цьому на графі присутні ребра для тих і лише тих сегментів, які зустрічаються в сегментаціях з $\bar{S}(\bar{k})$. З кожним ребром $\varepsilon(i, i+d(a), a)$ на графі пов'язано штраф $f(i,a)$. Штрафом за шлях є добуток штрафів його ребер. Ребро називатимемо значущим, якщо відповідний йому сегмент — значущий. Тоді величина $F_1(i,l)$ дорівнює сумарному штрафу за всі шляхи на графі від вершини 0 до вершини i, які містять l значущих ребер і останнім ребром яких є значуще ребро; $F_0(i,l)$ дорівнює сумарному штрафу за всі шляхи на графі від вершини 0 до вершини i, які містять l значущих ребер і останнім ребром яких є незначуще ребро. Відповідно, штраф за всі сегментації Z дорівнює сумарному штрафу за всі шляхи від вершини 0 до вершини W, які містять L значущих ребер.

Аналогічно величини B_0 та B_1 на цьому графі є штрафами за всі шляхи від кінцевої вершини W до інших вершин: $B_1(i,l)$ — сумарний штраф за шляхи від W до i, які містять $L-l+1$ значущих ребер і останнім ребром яких є значуще ребро $\varepsilon(i, i+d(k_l), k_l)$; $B_0(i,l)$ — сумарний штраф за шляхи від W до i, які містять $L-l$ значущих ребер і останнім ребром яких є незначуще ребро $\varepsilon(i, i+1, \kappa)$.

Насамкінець наведемо еквівалентне рекурсивне означення величин F_0 та F_1, що визначає ефективний алгоритм їхнього обчислення.

Нехай i приймає всі можливі значення координат стовпчиків зображення включно з фіктивним стовпчиком з координатою W: $i = \overline{0,W}$, а l — номер літери рядка $\bar{k}$: $l = \overline{0,L}$. Тоді величини $F_0(i,l)$ та $F_1(i,l)$ визначаються наступними рекурсивними співвідношеннями:

$$\begin{cases} F_0(0,0) = 1, \\ F_1(0,0) = 0, \\ F_0(i,l) = f(i-1,\kappa)\cdot(F_0(i-1,l) + F_1(i-1,l)), \quad l = \overline{0,L}, i = \overline{1,W}, \\ F_1(i,l) = f(i-d(k_l),k_l)\cdot(F_0(i-d(k_l),l-1) + F_1(i-d(k_l),l-1)), \quad l = \overline{1,L}, i = \overline{d(k_l),W}. \end{cases} \tag{19}$$

Очевидно, що введені таким чином величини F_1 та F_0 збігаються з наведеними у (13) та (16).

Аналогічно дамо рекурсивне визначення величин $B_0(i,l)$ та $B_1(i,l)$:

$$\begin{cases} B_0(W,L) = 1, \\ B_1(W,L) = 0, \\ B_0(i,l) = f(i,\kappa)\cdot(B_0(i+1,l) + B_1(i+1,l)), \quad l = \overline{0,L},\ i = \overline{0,W-1}, \\ B_1(i,l) = f(i,k_l)\cdot(B_0(i+d(k_l),l+1) + B_1(i+d(k_l),l+1)), \quad l = \overline{0,L-1},\ i = \overline{0,W-d(k_l)}. \end{cases} \tag{20}$$

Обчислені таким чином B_1 та B_0 збігаються з наведеними у (14) та (17).

Із співвідношень (9), (12) та (15) випливає остаточний вираз для значення апостеріорної ймовірності $\alpha(x,s)$ довільного сегмента $s = (i,a)$:

$$\alpha(x,s) = \begin{cases} \frac{1}{Z}\sum\limits_{\substack{1\le l\le L:\\ k_l = a}} \frac{F_1(i+d(a),l)\cdot B_1(i,l)}{f(i,a)}, & \text{якщо } a \in A_0; \\ \frac{1}{Z}\sum\limits_{l=0}^{L} \frac{F_0(i+1,l)\cdot B_0(i,l)}{f(i,a)}, & \text{якщо } a = \kappa, \end{cases} \tag{21}$$

де F_1 та F_0 обчислені згідно (13) та (16), B_1 та B_0 — згідно (14) та (17), а Z — згідно (18).

2.4 Алгоритм навчання задачі розпізнавання

Тепер, коли вказано спосіб обчислення всіх величин, що необхідні для ефективної реалізації алгоритму самонавчання, ми сформулюємо його ще раз, в цілісному вигляді. Але спочатку звернемо увагу читача на початкові значення еталонів символів E^0, апріорних ймовірностей сегментів $p^0(\bar{s}),\ \bar{s} \in \bar{S}$, що встановлюються на першому кроці роботи алгоритму, та критерій зупинки алгоритму. Їх вибір та вплив на роботу алгоритму описано в розділі, присвяченому експериментам.

Алгоритм 1 Навчання задачі розпізнавання

1: Обираємо початкові значення параметрів $(E^0, p^0(\bar{s}))$.

2: Користуючись отриманими параметрами $(E^r, p^r(\bar{s}))$ як істинними, обчислюємо числа F_0, F_1 та B_0, B_1 згідно (19) та (20).

3: За F_0, F_1 та B_0, B_1 обчислюємо $\alpha^r(x^m, s)$ згідно (21).

4: Згідно (10) та (11), оцінюємо нові значення параметрів $(E^{r+1}, p^{r+1}(s))$.

5: Якщо виконується критерій закінчення роботи, то кінець. Інакше переходимо до кроку 2.

3 Експериментальна перевірка

3.1 Тестові приклади

Експерименти проводилися наступним чином: на вхід алгоритму подавалась вибірка, що складалась з зображень текстових рядків та відповідних їм текстових рядків, а також розміри еталонів всіх літер алфавіту і величина дисперсії σ^2 гаусового розподілу (1). Експериментальну вибірку було розбито на дві частини: навчальну та тестову. Перша частина використовувалась для побудови еталонів символів на основі описаного в статті підходу, друга ж частина вибірки слугувала для оцінки якості розпізнавання за допомогою побудованих еталонів.

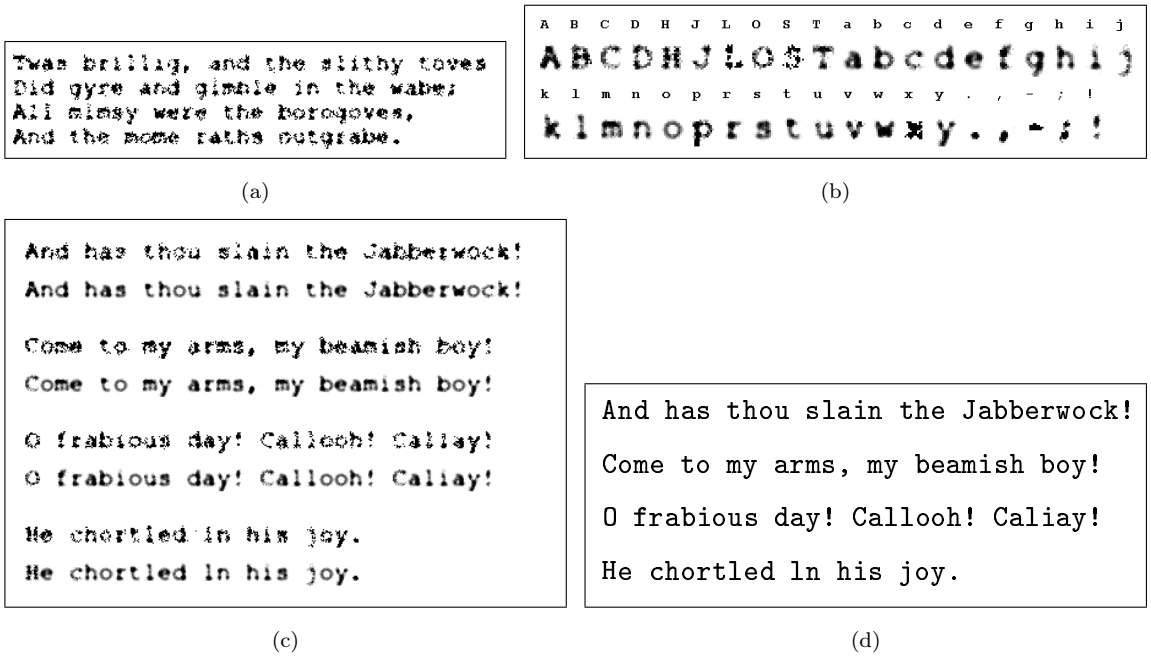
В цьому підрозділі ми розглянемо чотири різних приклади. Перший був створений штучним шляхом, три інших є реальними з різними видами спотворення зображень.

3.1.1 Приклад „Jabberwocky“

В цьому прикладі зображення були створені штучним шляхом із застосуванням моделі деградації зображень, описаної у [2]. Ця модель відображає зашумлення зображень чорно-білого друкованого тексту після численних копіювань. Такий вид зашумлення відносно добре описується статистичною моделлю, прийнятою в даній роботі. На рис. 1 наведено вхідні зображення та результати роботи алгоритму. На зразках тестових вхідних зображень (рис. 1(a)) можна оцінити характер зашумлення. Рис. 1(b) містить зображення еталонів символів, отриманих при навчанні. На рис. 1(c) наведено пари зображень текстових рядків, що демонструють результат розпізнавання текстових зображень. В кожній парі перше зображення — це вихідний рядок тестової вибірки, а друге — зображення рядка, склеєного з еталонів літер текстового рядка, що є результатом розпізнавання цього зображення. Нарешті, на рис. 1(d) наведено розпізнаний алгоритмом текст.

Навчальна вибірка мала об’єм близько 600 літер. Помилка при розпізнаванні склала близько 1%. Власне, в результатах розпізнавання присутні 2 помилки і в обох випадках переплутано літери „l“ та „i“. Еталони літер досить схожі, і шум легко робить більш ймовірним неправильну літеру.

Навчальна вибірка розпізнавалась без помилок.



(a) (b)

(c) (d)

Рис. 1: Приклад „Jabberwocky“. (a) — приклад вхідного зображення, (b) — побудовані еталони, (c), (d) — результати розпізнавання

3.1.2 Приклад „Book Intro“

Другий приклад містить зображення рядків відсканованої сторінки книги. Результати роботи алгоритмів наведено на рис. 2. Помилка розпізнавання тестової вибірки склала 0.8%. Типова помилка, яка спостерігалась у цьому експерименті — плутання літер „п“, „н“, „и“, еталони яких досить схожі один з одним.

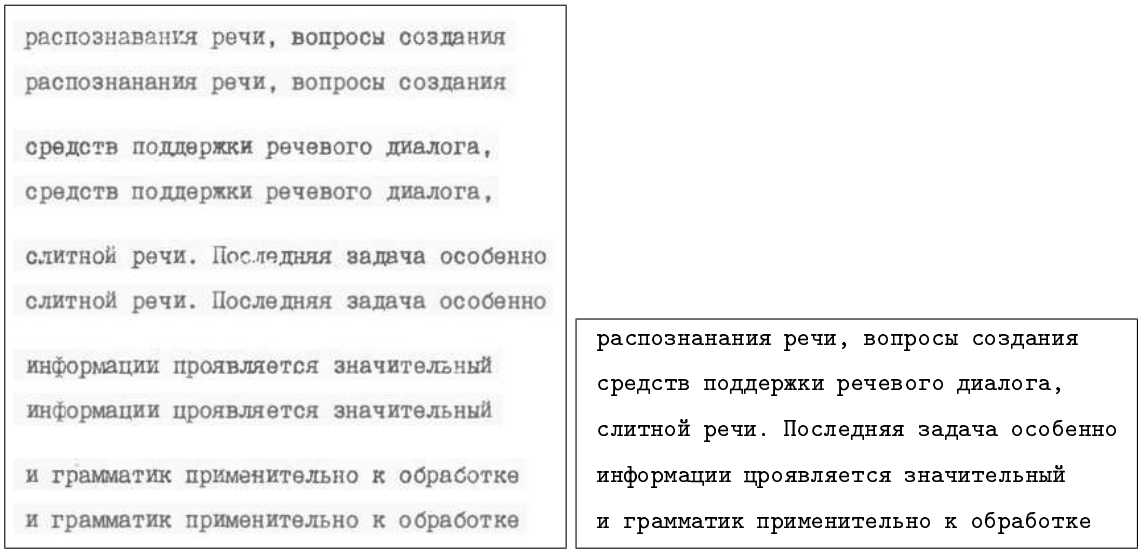
Навчальна вибірка мала об’єм 520 літер. При її розпізнаванні була зроблена одна помилка (0.8%), викликана значним спотворенням зображення літери.

3.1.3 Приклад „Gothic“

Наступний приклад є реальним зображенням з сильною деградацією. Результати розпізнавання наведено на рис. 3. Помилка розпізнавання тестової вибірки склала 3%, причому близько половини помилок виникло із-за того, що необхідні літери не були представлені в навчальній вибрці. При розпізнаванні навчальної вибірки, яка мала довжину 350 літер, помилка склала 2%.

3.1.4 Приклад „BigBrother“

Останній приклад також є реальним. Він був отриманий скануванням тексту, роздрукованого на лазерному принтері з дефектним барабаном. Як видно з рис. 4(a), характер зашумлення зображення значно відрізняється від прийнятої моделі, в першу чергу нерівномірністю насиченості фону. Тим не менш, підбором дисперсії σ^2 вдалось досягти розпізнавання тестової вибірки з рівнем помилок 5%. В цьому прикладі навчальна виборка довжини 1250 літер розпізнається з помилкою 5.3%. Більша кількість помилок при навчанні в порівнянні з розпізнаванням пояснюється тим, що в тестову



(a) (b)

Рис. 2: Приклад „Book Intro“. Результати розпізнавання

вибірку відбирались зображення, які мають не занадто велику зашумленість.

3.2 Початкові значення параметрів алгоритму

На першому кроці роботи алгоритму встановлюються початкові значення еталонів символів та апріорних ймовірностей сегментів. Крім того, перед початком роботи необхідно задати значення дисперсії σ^2 гаусового розподілу (1).

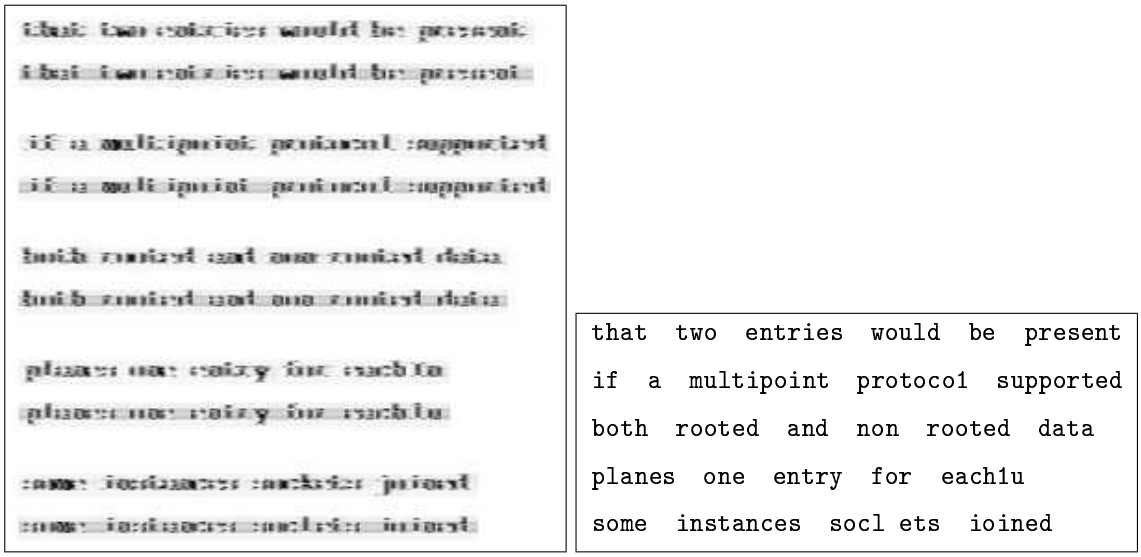
Вибір початкових значень параметрів має певний вплив на роботу алгоритму. Так, якщо покласти початкові еталони кольору тексту для літер та кольору фону для вставки, то це призводить до більш впевненої роботи алгоритму навчання, у тому сенсі, що алгоритм збігається швидше. Тим не менш, саме такі початкові значення еталонів не є обов'язковими. Повністю прийнятними є, наприклад, еталонні зображення, заповнені випадковими значеннями пікселів.

Вплив початкових значень апріорних ймовірностей $p^0(s)$ не такий наочний. Так, задання ймовірностей певних сегментів рівними нулю „забороняє“ використання цих сегментів. Тим не менш, у загальному випадку, найкращими початковими значеннями апріорних ймовірностей є однакові значення для всі сегментів.

В розглянутих експериментах початкові параметри $(E^0, p^0(\bar{s}))$ були вибрані наступними: однакові ймовірності для всіх сегментів, еталонні зображення чорні для всіх літер та білі для вставки. Такі еталони є природними для темного тексту на світлому фоні.

Крім описаних початкових еталонів, також проводились експерименти з початковими зображеннями, заповненими випадковими значеннями пікселів. Відмінність при цьому від описаних вище початкових еталонів в основному полягає у збільшенні кількості ітерацій для отримання того ж результату і мало помітна при якісному порівнянні результатів навчання та розпізнавання.

Дисперсія σ^2 є характеристикою зашумленості зображень, що подаються на вхід алгоритму. При збільшенні цього параметру зменшується різниця між штрафами за різні літери для одного



(a) (b)

Рис. 3: Приклад „Gothic“. Результати розпізнавання

прямокутного фрагменту. При зменшенні — ця різниця зростає, але при цьому збільшується вплив нерівномірності шуму. Оптимальне значення дисперсії залежить від зображень, проте оптимальні значення для досліджуваних зображень не сильно різняться.

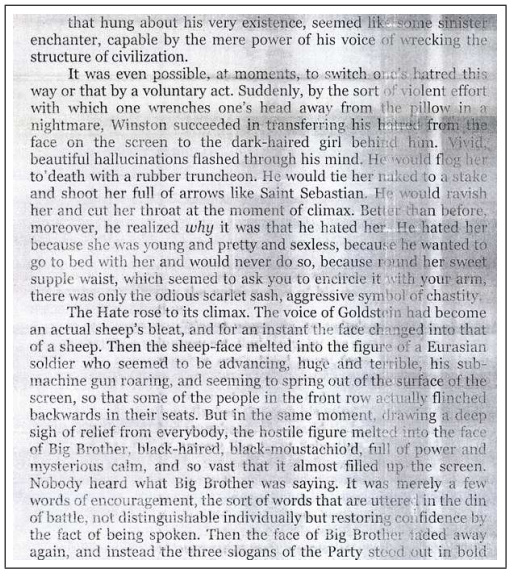
Експерименти показали, що невеликі відхилення значень параметрів від емпіричних оптимальних значень не мають значного впливу на результати навчання. На відміну від цього, алгоритм є дуже чутливим до зміни фіксованих ширин еталонів літер. Задання ширин, більших за істинні, може значно погіршити результати навчання та розпізнавання. Причини цього лежать у постановці задачі розпізнавання, що забороняє накладання еталонів сусідніх літер.

3.3 Критерії зупинки алгоритму навчання

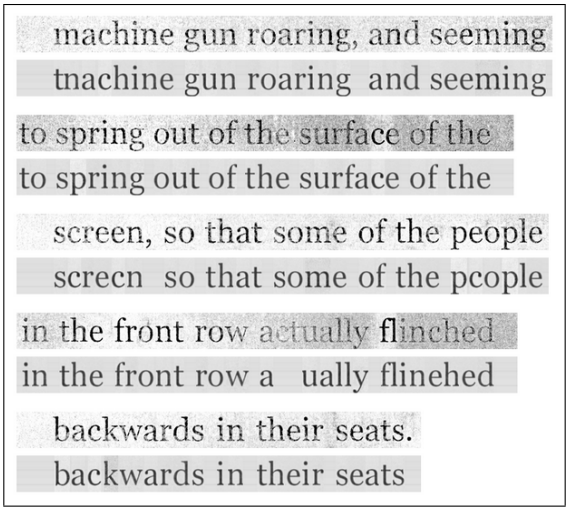
Очевидним критерієм зупинки алгоритму є правильне розпізнавання всіх навчальних зображень. Але постановка задачі у тому вигляді, як це було зроблено в цій роботі (пошук параметрів, що максимізують функцію правдоподібності), не гарантує збігання алгоритму до параметрів, які забезпечують правильне розпізнавання навчальної вибірки. Тому такий критерій, принаймні у чистому вигляді, не може бути застосований.

Найпростішим критерієм зупинки є досягнення певної кількості ітерацій. Проте такий критерій є кількісним і не забезпечує отримання певного якісного результату. Іншим простим критерієм може бути зменшення відносної зміни функції правдоподібності до деякого порогового значення.

Експерименти показали, що алгоритм збігається досить швидко — як правило, потрібно від 3 до 5 ітерацій для того, щоб отримати результат, який вже не покращується, у тому сенсі, що функція правдоподібності мало змінюється. Тим не менш, отриманий результат може виявитись локальним максимумом функції правдоподібності, а не глобальним, що відображається у неправильній побудові декількох еталонів і, відповідно, неправильному розпізнаванні навчальної вибірки. У та-



(a)



(b)

```
tnachine gun roaring and seeming
to spring out of the surface of the
screcn so that some of the pcople
in the front row a  ually flinehed
backwards in their seats
```

(c)

Рис. 4: Приклад „Big Brother“. (a) — вхідне зображення, (b), (c) — результати розпізнавання

кому випадку алгоритм можна виштовхнути з локального максимуму реініціалізацією неправильно побудованих еталонів та апріорних ймовірностей сегментів.

На підставі цих спостережень був розроблений евристичний алгоритм реініціалізації параметрів, який полягає у наступному. На кожному його кроці алгоритм навчання проходить декілька ітерацій, після чого навчальна вибірка розпізнається. Розпізнані рядки порівнюються з заданими у вибірці, і еталони літер, на яких алгоритм помилився, оголошуються „винними“ і скидаються до початкових значень. Після цього починається новий крок. Алгоритм працює доти, доки всі зображення з навчальної вибірки не будуть розпізнаватись правильно або ж до досягнення заданої кількості ітерацій.

3.4 Застосування алгоритму як допоміжного для алгоритмів настройки параметрів

Описаний підхід може використовуватись не лише як самостійний метод статистичного оцінювання еталонів, але й як допоміжний метод для інших алгоритмів навчання, наприклад алгоритмів настройки (див. [3, 4]).

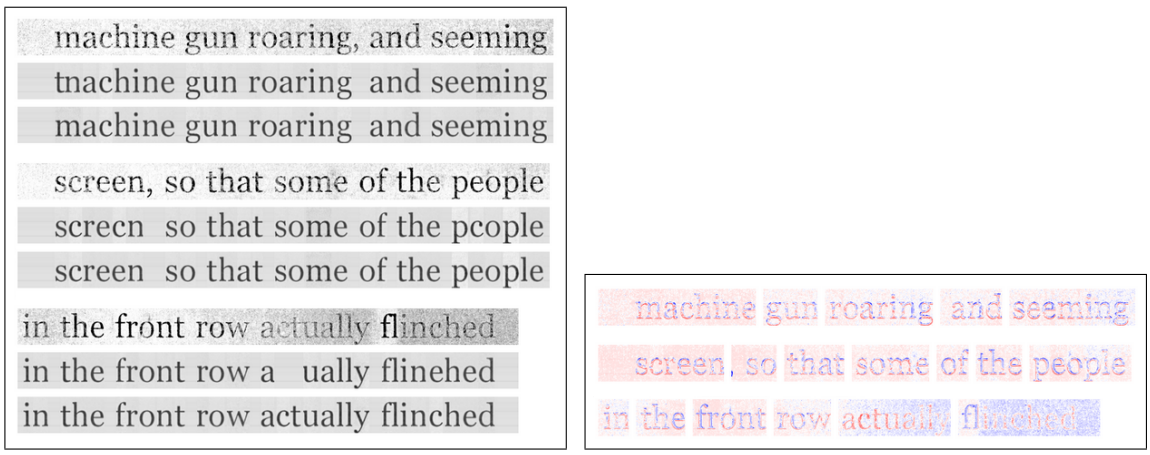
Алгоритми настройки не так сильно, як описаний нами, залежать від характеру зашумленості

зображень, тому при значному відхиленні характеру зашумлення від принятої в роботі гаусової моделі забезпечують кращі результати.

Недоліком алгоритмів настройки є вимога наявності в навчальній вибірці точної сегментації кожного з навчальних зображень. Як вже було неодноразово зазначено, побудова такої сегментації зазвичай потребує значних зусиль та часу оператора.

Водночас на основі описаного нами підходу навіть для сильнозашумлених зображень можуть бути отримані сегментації достатньої точності. Побудова сегментації складається з двох етапів: спочатку на основі текстового рядка та вхідного зображення будуються еталони літер так, як це описано в даній роботі. На другому етапі на основі отриманих еталонів знаходиться найімовірніша сегментація серед тих, що відповідають введеному на першому етапі текстовому рядку. Ця сегментація і приймається в якості вхідної для наступного алгоритму настройки.

Нижче наведено приклад роботи такого комбінованого налаштування еталонів на тексті з прикладу „Big Brother“. На рис. 5(a) зображено результати розпізнавання, де перший рядок кожної групи — це вхідне зображення, другий — результат звичайного розпізнавання, третій — найімовірніша сегментація серед тих, що відповідають текстовому рядку з навчальної вибірки. На рис. 5(b), на якому наведено попіксельні різниці першого та третього рядків кожної групи, можна оцінити точність створеної сегментації.



(a) (b)

Рис. 5: Автоматична побудова точних сегментацій. (a) — результати звичайного та модифікованого розпізнавання, (b) — той самий результат, отриманий як попіксельна різниця першого та третього рядків з попереднього рисунка

Висновки

Підхід до оцінювання еталонів літер для задачі розпізнавання тексту, запропонований в роботі, дозволяє суттєво зменшити обсяг ручної роботи при підготовці навчальної вибірки, оскільки замість точної сегментації навчальних зображень на окремі літери на вхід алгоритму подаються лише текстові рядки, що відповідають цим зображенням.

Перспективним напрямком подальших досліджень в цій галузі є алгоритми навчання, в ході роботи яких автоматично налаштовуються не лише кольори (градації сірого) еталонних зображень літер, але й розміри цих зображень.

Література

[1] М. Шлезингер and В. Главач. *Десять лекций по статистическому и структурному распознаванию.* Наукова думка, Киев, 2004.

[2] Prateek Sarkar, Henry S. Baird, and Xiaohu Zhang. Training on severely degraded text-line images. In *IAPR 7th International Conference on Document Analysis and Recognition (ICDAR03)*, pages 38–43, Edinburgh, Scotland, August 2003.

[3] Б. Д. Савчинський and О. В. Камоцький. Настройка алгоритму розпізнавання тексту. *Управляющие системы и машины*, (2):17–24, 2005.

[4] Bogdan Savchynskyy and Olexander Kamotsky. Character templates learning for textual images recognition as an example of learning in structural recognition. In *Second International Conference on Document Image Analysis for Libraries (DIAL'06)*, pages 88–95, Lyon, France, April 2006.