

Розробка та дослідження програмного забезпечення для розпізнавання текстів

Бачеріков В'ячеслав

Вінницький Національний Технічний Університет

Вінниця, 2019

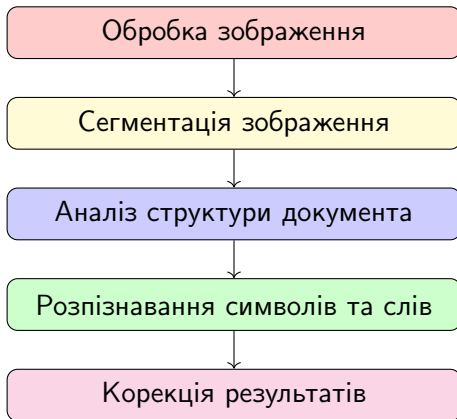
Сучасний розвиток методів глибинного навчання став поштовхом різноманітних досліджень комп'ютерного зору та оптичного розпізнавання символів. Сучасні методи дозволяють розпізнавати не лише друковані символи на білому фоні, а навіть рукописні символи та текст у реальних умовах, де фон може бути будь-яким і розташовуватись різноманітним чином.

Мета розробки та предмет дослідження

- Головною метою є підвищення достовірності розпізнавання та сегментації друкованого та рукописного тексту.
- Об'єктом дослідження є методи розпізнавання символів у різноманітних умовах, сегментація тексту та попередня обробка зображення перед розпізнаванням.
- Предметом дослідження є програмне забезпечення для оптичного розпізнавання текстів.

Задачі дослідження:

- 1 Виконати аналіз існуючих методів та підходів для розпізнавання символів та слів.
- 2 Виконати аналіз існуючих методів сегментації зображення на слова та літери.
- 3 Побудувати нейронну мережу для розпізнавання друкованих та рукописних символів.
- 4 Розробити програмне забезпечення для навчання нейронної мережі.
- 5 Створити програмне забезпечення для сегментації та розпізнавання тексту.
- 6 Провести тестування розробленого програмного забезпечення.



Оригінал

путь в P]. Пометьте в узле u метку T и перейдите к шагу $T2$.

закончительном шаге этого алгоритма слово "пометьте" означает, что во время обхода дерева при попадании в узел выполняется заранее предусмотренная операция. Алгоритм T по отношению к другим действиям основной программы является как подпрограмма, т. е. основная программа вызывает эту подпрограмму рекурсивно от одного узла к другому в заданном симметричном порядке. Конечно, и эта подпрограмма вызывает основную процедуру только в одном месте, она отличается от подпрограммы (см. раздел 1.4.2). В алгоритме T предусматривается, что в результате выполнения внешних действий в данном дереве не удаляется ни $NODE(P)$, ни любой другой его узел-предшественник.

Чтобы понять идею, лежащую в основе этого алгоритма, читатель может в любое удобное время применить алгоритм T к бинарному дереву (2). После полезного упражнения применить алгоритм T к бинарному дереву с корнем, значением шага $T3$ следует приступить к обходу бинарного дерева в соответствии с указателем P и указывать P . При этом основная идея заключается в сохранении указателя P в последующим обходом левого поддерева. После выполнения этих действий можно вернуться к шагу $T4$ и найти прежнее значение P в стеке. После обхода левого поддерева на шаге $T5$ остается только совершить обход правого поддерева. Алгоритм T типичен для многих других алгоритмов, которые будут рассмотрены в следующем абзаце. Докажем с помощью метода индукции, что алгоритм T

Глобальный трешхолдінг Адаптивний трешхолдінг

путь в P]. Пометьте в узле u метку T и перейдите к шагу $T2$.

закончительном шаге этого алгоритма слово "пометьте" означает, что во время обхода дерева при попадании в узел выполняется заранее предусмотренная операция. Алгоритм T по отношению к другим действиям основной программы является как подпрограмма, т. е. основная программа вызывает эту подпрограмму рекурсивно от одного узла к другому в заданном симметричном порядке. Конечно, и эта подпрограмма вызывает основную процедуру только в одном месте, она отличается от подпрограммы (см. раздел 1.4.2). В алгоритме T предусматривается, что в результате выполнения внешних действий в данном дереве не удаляется ни $NODE(P)$, ни любой другой его узел-предшественник.

Чтобы понять идею, лежащую в основе этого алгоритма, читатель может в любое удобное время применить алгоритм T к бинарному дереву (2). После полезного упражнения применить алгоритм T к бинарному дереву с корнем, значением шага $T3$ следует приступить к обходу бинарного дерева в соответствии с указателем P и указывать P . При этом основная идея заключается в сохранении указателя P в последующим обходом левого поддерева. После выполнения этих действий можно вернуться к шагу $T4$ и найти прежнее значение P в стеке. После обхода левого поддерева на шаге $T5$ остается только совершить обход правого поддерева. Алгоритм T типичен для многих других алгоритмов, которые будут рассмотрены в следующем абзаце. Докажем с помощью метода индукции, что алгоритм T

путь в P]. Пометьте в узле u метку T и перейдите к шагу $T2$.

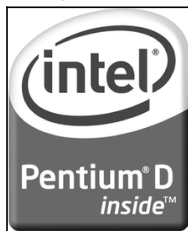
закончительном шаге этого алгоритма слово "пометьте" означает, что во время обхода дерева при попадании в узел выполняется заранее предусмотренная операция. Алгоритм T по отношению к другим действиям основной программы является как подпрограмма, т. е. основная программа вызывает эту подпрограмму рекурсивно от одного узла к другому в заданном симметричном порядке. Конечно, и эта подпрограмма вызывает основную процедуру только в одном месте, она отличается от подпрограммы (см. раздел 1.4.2). В алгоритме T предусматривается, что в результате выполнения внешних действий в данном дереве не удаляется ни $NODE(P)$, ни любой другой его узел-предшественник.

Чтобы понять идею, лежащую в основе этого алгоритма, читатель может в любое удобное время применить алгоритм T к бинарному дереву (2). После полезного упражнения применить алгоритм T к бинарному дереву с корнем, значением шага $T3$ следует приступить к обходу бинарного дерева в соответствии с указателем P и указывать P . При этом основная идея заключается в сохранении указателя P в последующим обходом левого поддерева. После выполнения этих действий можно вернуться к шагу $T4$ и найти прежнее значение P в стеке. После обхода левого поддерева на шаге $T5$ остается только совершить обход правого поддерева. Алгоритм T типичен для многих других алгоритмов, которые будут рассмотрены в следующем абзаце. Докажем с помощью метода индукции, что алгоритм T

Обробка зображення

Недоліки адаптивного трешхолдінгу

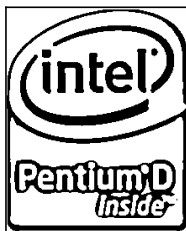
Оригінал



Глобальний



Адаптивний



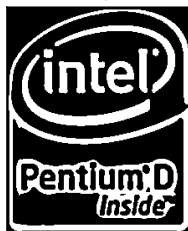
Інверсія



Сегментація зображення

Blob detection

Бінарне



Границі



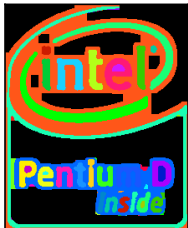
Маска



Блоби



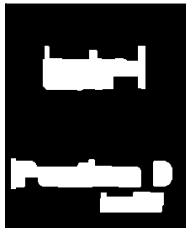
Блоби



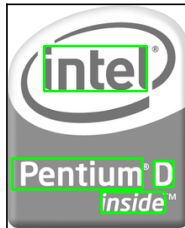
Відфільтр. блоки



Морф. закриття



Слова



- **kNN** класифікатор та модифікації (*k-d tree* та *інші*);
- Лінійна регресія;
- Логістична регресія;
- Метод опорних векторів (*SVM*);
- Нейронні мережі.

Розпізнавання символів

Набір символів для розпізнавання

- Цифри: 0 1 2 3 4 5 6 7 8 9
- Латинські літери¹:
Aa Bb Cc Dd Ee Ff Gg Hh Ii Jj Kk Ll Mm Nn Oo Pp Qq Rr Ss
Tt Uu Vv Ww Xx Yy Zz
- Кирилиця (російські та українські літери)²:
Aa Бб Вв Гг Дд Ее Ёё Жж Зз Ии Йй Кк Лл Мм Нн Оо Пп Рр Сс
Тт Уу Фф Хх Цц Чч Шш Щщ Ъъ Ыы Ьь Ээ Юю Яя Ёё Іі Іі
- Додаткові символи³: % ! \$ @ + - = . , : ; ()
- Загальна кількість символів: 153

¹52 символи (великих та маленьких)

²78 символів (великих та маленьких)

³13 символів

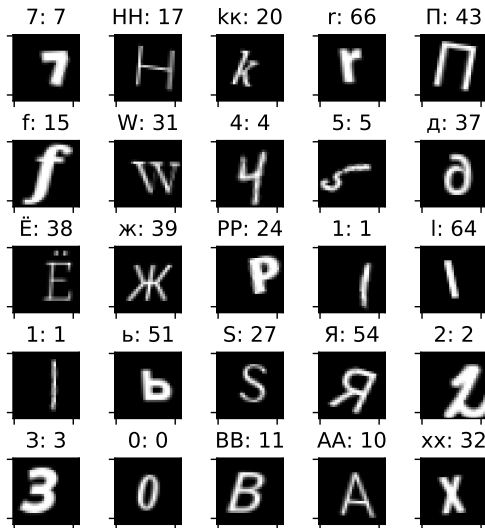
Розпізнавання символів

Складнощі розпізнавання

- Схожість між різними друкованими символами: $0 \rightarrow O$, $C \rightarrow C$, $i \rightarrow i$, $6 \rightarrow б$
- Схожість між регістрами: $Z \rightarrow z$, $W \rightarrow w$, $\Gamma \rightarrow \gamma$ (*більшість українських та російських літер*)
- Схожість між рукописними символами: $1 \rightarrow l$, $i \rightarrow j$
- Різноманітність накреслень одного символу (капітель, курсив): $a \rightarrow A$, $\ddot{e} \rightarrow \ddot{E}$, $\tau \rightarrow m$, $d \rightarrow \mathcal{d}$
- Лігатури: $fi \rightarrow fi$, $\mathbf{AE} \rightarrow \mathbf{\AE}$, $ff \rightarrow ff$, $oe \rightarrow \mathbf{\oel}$ $fl \rightarrow fl$
- Відхилення від базової лінії: $\text{L}\text{T}\text{E}\text{X}$
- Шуми, артефакти та викривлення на зображенні

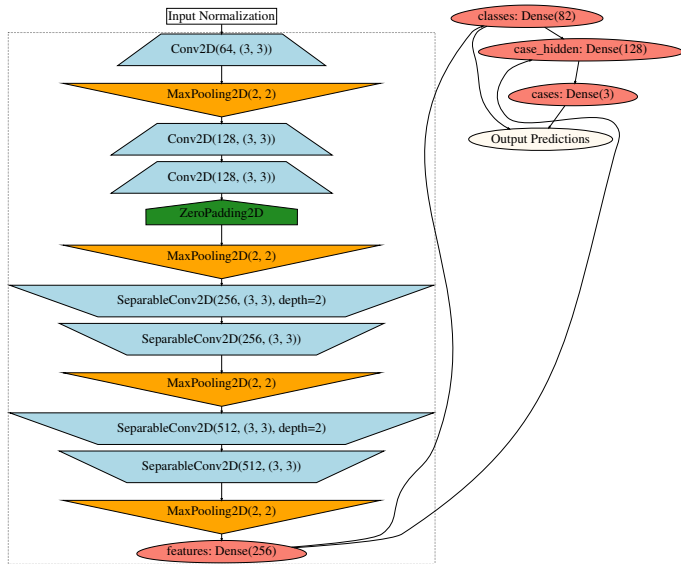
Розпізнавання символів

Приклад зібраних та аугментованих символів



Розпізнавання символів

Архітектура нейронної мережі



$$\text{Loss} = H(s, \hat{s}) + \beta H(c, \hat{c})$$

$$H(y, \hat{y}) = - \sum_i y_i \log \hat{y}_i \quad \text{та} \quad S(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}}$$

$$\hat{s} = S(a^{(s)}) \quad \hat{c} = S(a^{(c)})$$

де s – коректний клас символу в унітарному коді;

c – коректний регістр символу в унітарному коді;

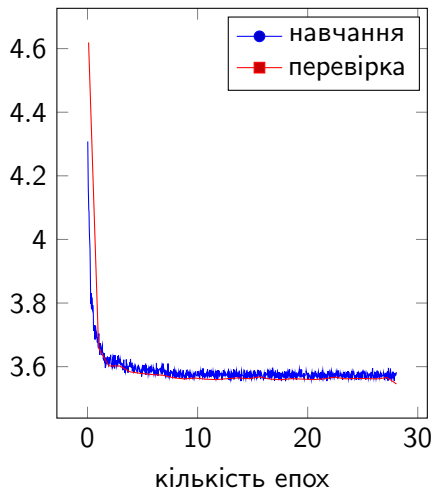
$a^{(s)}$ – результуючий вектор з шару symbols нейронної мережі;

$a^{(c)}$ – результуючий вектор з шару cases нейронної мережі;

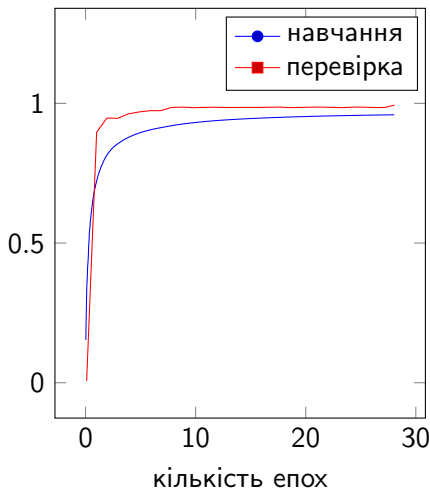
β – константа.

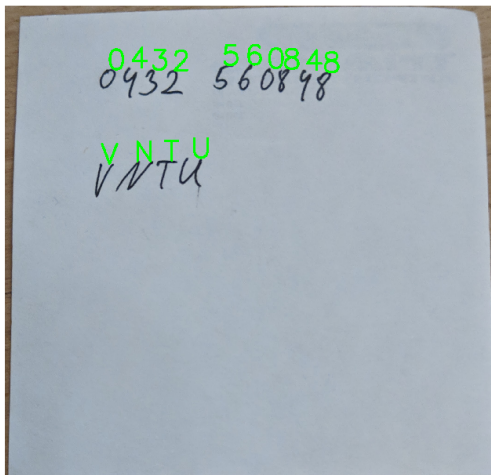
Навчання нейронної мережі

loss



accuracy





- Використання рекурентних нейронних мереж разом з CNN (*LSTM*, *GRU*, *CTC*);
- Збір більшої кількості зображень символів;
- Покращення якості зібраних символів;
- Підтримка рукописного курсиву (скоропису);
- Покращення сегментації на слова.

У ході виконання роботи розроблено систему оптичного розпізнавання текстів, яка підтримує 3 мови: англійську, російську та українську. Загальна кількість підтримуваних символів 153. Система працює як з друкованими символами так і рукописними символами. У ході виконання було зібрано більш ніж 3 мільйони зразків символів, котрі використовувались для навчання нейронної мережі. Для зменшення кількості параметрів та підвищення швидкодії нейронна мережа використовує глибинно розділенні згортки (*depthwise separable convolutions*).

Дякую за увагу