УДК 004.9

Д.К. Ковальов, С.М. Вовк

КОМП'ЮТЕРНА СИСТЕМА ДЛЯ ЗЧИТУВАННЯ ДРУКОВАНОГО ТЕКСТУ ТА ЙОГО КОРИГУВАННЯ

Анотація. Пропонується комп'ютерна система для зчитування та коригування друкованого тексту на основі бібліотек з відкритим програмним кодом Tesseract-OCR, EmguCV та NHunspell. Система розроблена на базі програмної платформи Microsoft .NET Framework та дозволяє вводити зображення з друкованим текстом (фото, відсканована сторінка тощо) у стандартних графічних форматах, виконувати його розпізнавання, перевіряти отриманий текст на орфографічні помилки та виправляти їх. Система підтримує автоматичний режим коригування тексту та інтерактивний режим для самостійного виправлення помилок.
Ключові слова: розпізнавання тексту, перевірка орфографії, Tesseract-OCR, EmguCV, NHunspell.

Вступ

Інтелектуальна обробка друкованих текстів є одним із найважливіших напрямків у галузі сучасних інформаційних технологій, що пов'язані з документообігом та автоматизацією відповідних процесів. Актуальність та важливість розвитку даного напрямку обумовлена як необхідністю створення цифрових бібліотек зі старих друкованих книг й видань, так і необхідністю покращення візуальної якості документів з низьким рівнем контрасту та, в цілому, підвищення ефективності процесів документообігу. В поданій роботі пропонується комп'ютерна система для зчитування та коригування друкованого тексту, яка створена основі бібліотек з відкритим програмним кодом Tesseract-OCR, EmguCV та NHunspell.

Постановка задачі

Постановка задача полягає у створенні комп'ютерної системи з відкритим програмним кодом, яка дозволяє вирішувати задачі введення та попередньої обробки зображення з друкованим текстом, розпізнавання тексту на зображенні й роботи зі словником для перевірки та коригування отриманого тексту.

Основна частина

Розв'язок поставленої задачі вимагає наявності двох основних програмних компонентів, які повинні опрацьовувати вхідне зображення та отриманий з нього текст, відповідно.

Перший програмний компонент може бути розроблений у такий спосіб. Для введення зображень з друкованим текстом є доцільним використання стандартного компоненту System.Drawing платформи .NET Framework 4.0, який дозволяє вводити зображення за форматами JPEG, BMP, TIF, PNG, GIF тощо. Наступним кроком опрацювання вхідного зображення є його попередня обробка з метою вирівнювання контрасту та зменшення шуму. Для виконання попередньої обробки зображень можна використати функції бібліотеки комп'ютерного зору EmguCV, яка є адаптованою версією бібліотеки OpenCV. Це є доцільним і тому, що EmguCV може працювати з .NET-сумісними мовами, такими як C#, VB, VC++ і т.д. та працює з Windows, Linux, Android і Windows Phone. Використання цієї бібліотеки дозволяє спростити архітектуру комп'ютерної через використання готових до роботи алгоритмів попередньої обробки зображень. Останнім кроком попередньої обробки є виконання бінаризації зображення. Одним з найкращих алгоритмів бінаризації вважається алгоритм Отцу [1].

На етапі розпізнавання тексту на зображенні та його конвертації в електронний вигляд традиційним є використання технології OCR (Optical Character Recognition). В даній роботі використовувалася безкоштовна OCR-бібліотека Tesseract від Google [2], [3]. Ця бібліотека формує блоби (з англ. «blob») – текстові рядки, в яких визначається фіксований розмір кроку між символами. Саме визначення дистанцій між символами дозволяє правильно відрізняти пробіли між словами від стандартної дистанції між літерами одного слова. Далі в одному рядку формуються символьні комірки з контурами літер (одна літера в комірці). Після виконання зазначеного перетворення, розпізнавання проводиться в два етапи. На першому етапі кожне слово розпізнається шляхом порівняння контурів його літер з еталонами в допоміжних файлах Tesseract. В ході розпізнавання, для кожної літери формується її еталон в межах поточного зображення. Після цього проводиться другий етап, яким є нове розпізнавання, де враховуються поточні еталони. Це покращує якість розпізнавання контуру

 ISSN 1562-9945

літери та дає більш точний результат. На другому етапі перевіряються області між комірками, в яких можуть міститися знаки пунктуації, крапки літер на зразок умляут (ь, ц, д), тощо.

Другий програмний компонент системи призначений для перевірки тексту та коригування орфографічних помилок. Для рішення цієї задачі доцільним є використання програмного додатку NHunspell [4]. NHunspell є вільною в застосуванні програмою-словником перевірки орфографії, що призначена для мов зі складною системою словотворення і великою морфологією. Для перевірки орфографії додаток NHunspell потребує два файли. Перший файл містить слова, а другий є файлом афіксів, який визначає значення спеціальних міток (прапорців) в словнику. Файл словника (.dic) містить список слів, по одному слову в рядку, а файл афіксів (.aff) може містити необов'язкові атрибути. Наприклад, атрибут SET визначає кодування символів файлів афіксів і словника.

На основі розглянутих пропозицій була розроблена відповідна комп'ютерна система у вигляді Windows-додатку, що має зручний графічний інтерфейс. У головному вікні програмного додатку користувач може обрати файл з зображенням для розпізнавання, застосувати обробку зображення та обрати режим коригування розпізнаного тексту. Програмне забезпечення реалізовано мовою C#. Як приклад, на рис. 1, зображено метод, який реалізує заміну неправильних слів (відсутніх в словнику обраної мови) в інтерактивному режимі.

На рис. 2 показано результат обробки фрагменту зображення поганої якості, але якість розпізнавання була задовільною.

На рис. 3 подано загальний вигляд програмного додатку. На ньому основне поле займає зображення з друкованим текстом, яке подається на обробку. Праворуч розташоване поле з електронним варіантом тексту, який був отриманий в результаті обробки, та перевірений на орфографічні помилки в автоматичному режимі. Результати тестування довели працездатність програмного додатку для тестових зображень, які отримані за допомогою камери мобільного телефону з аркушу паперу з друкованим текстом.

```
public static string Correct(string inputText, string inputAff, string inputDic)
{
    FillingSymbols(uncheckedSymbols);
    inputWords = SeparateWordsWithEnter(inputText);
    using (Hunspell hunspell = new Hunspell(inputAff, inputDic))
    {
        for (int i = 0; i < inputWords.Length; i++)
        {
            currentWord = inputWords[i];
            if (uncheckedSymbols.Contains(currentWord)) continue;
            if (hunspell.Spell(currentWord)) continue;
            else
            {
                try
                {
                    suggestions = hunspell.Suggest(currentWord);
                    rgx = new Regex(currentWord);
                    if (suggestions.Count == 0)
                    {
                        inputText = rgx.Replace(inputText, CreateAddingForm(currentWord), 1);
                        continue;
                    }
                    inputText = rgx.Replace(inputText, CreateChooseForm(currentWord, suggestions.ToArray()), 1);
                    continue;
                }
                catch (Exception e)
                {
                    continue;
                }
            }
        }
    }
    return inputText;
}
```

Рисунок 1 - Програмний код функції виправлення тексту

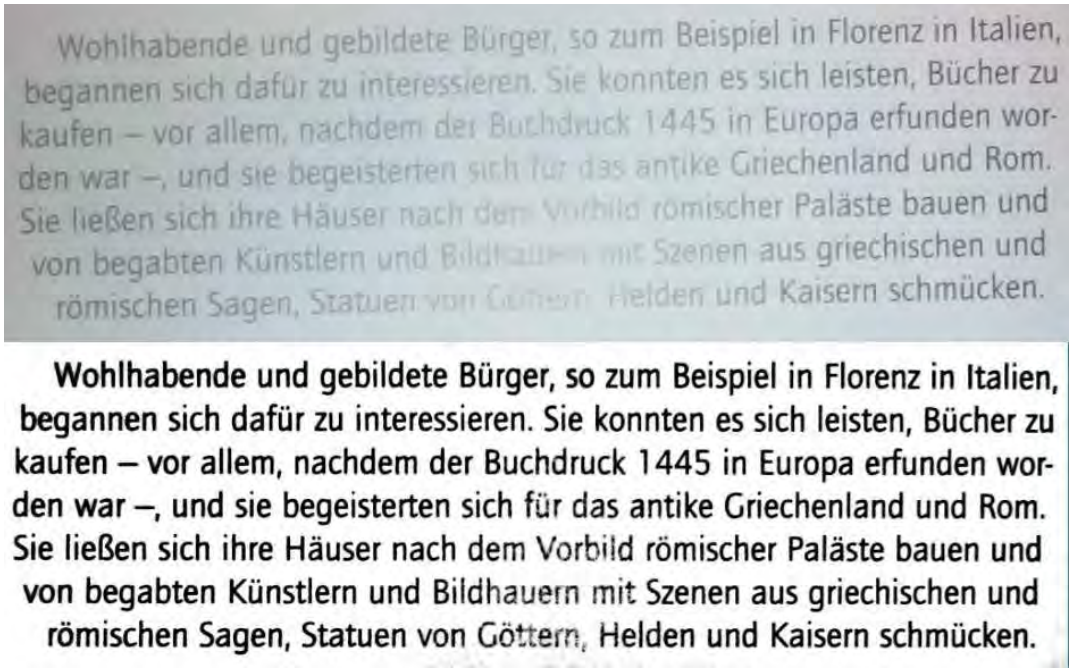

Рисунок 2 - Вхідне зображення (вгорі)
та зображення після обробки (знизу)

 ISSN 1562-9945

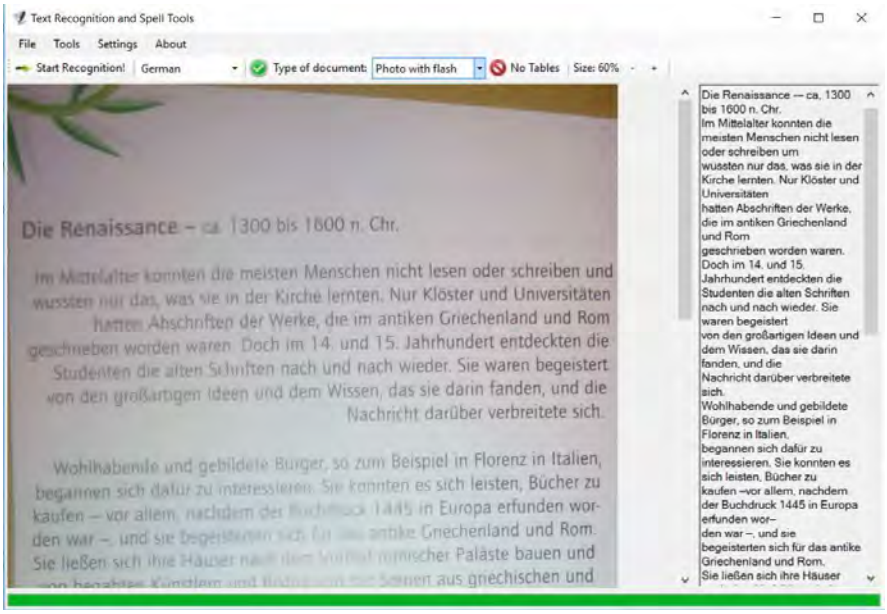

Рисунок 3 - Розроблений програмний додаток з результатом тестування

Висновки

Розроблена комп'ютерна система має відкритий програмний код та дозволяє зчитувати друкований текст з зображень, перетворювати його в електронну форму та перевіряти орфографію й виправляти помилки.

Подальший розвиток розробленої системи є доцільним у вигляді free online-сервісу або мобільного додатку.

ЛІТЕРАТУРА

1. Shi J., Ray N., Zhang H. Shape Based Local Thresholding for Binarization of Document Images / J. Shi, N. Ray, H. Zhang // Pattern Recognition Letters. – 2012. – Elsevier. – P.1-6
2. Smith R. An Overview of the Tesseract OCR Engine / R. Smith // IEEE Computer Society. – 2007. –P. 629-633
3. Patel C., Patel A., Patel D. Optical Character Recognition by Open Source OCR Tool Tesseract / C. Patel, A. Patel, D. Patel // A Case Study International Journal of Computer Applications. – 2012. – V. 55. – №.10. – P.50
4. Pirinen T. A., Lindйn K. Creating and Weighting Hunspell Dictionaries as Finite-State Automata / T. A. Pirinen, K. Lindйn // Investigationes Linguisticae. – 2010. –V. 21. – P. 3-6