

ЕЛЕКТРОННІ ІНФОРМАЦІЙНІ РЕСУРСИ: СТВОРЕННЯ, ВИКОРИСТАННЯ, ДОСТУП

ЗБІРНИК МАТЕРІАЛІВ

Міжнародної науково-практичної Інтернет-конференції

20-21 листопада 2023 р.

Міністерство освіти і науки України
Вінницький національний технічний університет
Національна академія Державної прикордонної служби України
ім. Богдана Хмельницького
Вінницький національний медичний університет ім. М.І. Пирогова
КЗВО «Вінницька академія безперервної освіти»
КЗ «Сумський обласний інститут післядипломної педагогічної освіти»
Інститут комп'ютерних систем і технологій "Індустрія 4.0"
ім. П. Н. Платонова
Люблінська політехніка (Польща)
Університет Бельсько-Бяльський (Польща)

**«ЕЛЕКТРОННІ ІНФОРМАЦІЙНІ
РЕСУРСИ: СТВОРЕННЯ, ВИКОРИСТАННЯ,
ДОСТУП»**

ЗБІРНИК МАТЕРІАЛІВ

Міжнародної науково-практичної Інтернет-конференції
20-21 листопада 2023 р.

Суми/Вінниця
НІКО/КЗВО «Вінницька академія безперервної освіти»
2023

УДК 004
ББК 32.97
Е50

Рекомендовано до видання Вченою радою КЗВО «Вінницька академія безперервної освіти» (протокол № 8 від 20.11.2023 р.)

Електронні інформаційні ресурси: створення, використання, доступ.
Збірник матеріалів Міжнародної науково-практичної Інтернет конференції 20-21 листопада 2023 р. – Суми/Вінниця: НІКО/КЗВО «Вінницька академія безперервної освіти», 2023. – 336 с.

ISBN 978-617-7422-23-4

Збірник містить матеріали Міжнародної науково-практичної Інтернет конференції «Електронні інформаційні ресурси: створення, використання, доступ. Матеріали збірника подано у авторській редакції. Автори опублікованих матеріалів несуть повну відповідальність за підбір, точність наведених фактів, цитат, статистичних даних, власних імен та інших відомостей, Матеріали відтворюються зі збереженням змісту, орфографії та синтаксису текстів, наданих авторами.

УДК 004
ISBN 978-617-7422-23-4

© КЗВО «Вінницька академія безперервної освіти», 2023
© Вид-во Суми, НІКО, 2023

Гронюк Р.О., Ліщинська Л.Б.	Порівняльний аналіз методів і програмних засобів автоматизації відносин з клієнтами	89
Гуралюк А.Г.	Система інтеграції електронних ресурсів ONTOS.	91
Дика А.І.	Тестування штучного інтелекту: ключові виклики, стратегії вдосконалення	93
Дідик В.І.	Гейміфікація	95
Дмитрієва О.А., Зіборов Д.Ю.	Багатокристувацька інформаційна система управління нотатками	96
Доценко Д.В., Романюк О.Н., Котлик С.В., Чехмestрук Р.Ю., Майданюк В.П.	Використання нейронних мереж для аналізу складності ігрових ситуацій у комп'ютерних іграх	98
Єжова Є. О.	Нейронна мережа аутентифікації користувача за клавіатурним почерком	100
Завальнюк Є.К.	Розробка плагінів для 3DS MAX	103
Завальнюк Є.К., Романюк О.Н.	Аналіз процедур розпаралелення рендерингу графічних сцен	105
Зарічний В. М., Романюк О. Н.	Аналіз графічного двигуна SOURCE для розробки комп'ютерних ігор	107
Захарчук М. Д., Романюк О. Н., Мельник О. В., Романюк С. О., Прозор О. П.	Аналіз технології OLED	109
Зінько П.О.	Система генерації портрету підозрюваного на основі наявного фоторобота за допомогою GAN	110
Кавка О.О., Майданюк В.П.	Аналіз алгоритмів стиснення зображень із втратами на основі дискретного косинусного перетворення	112
Кирнасюк Є. С., Майданюк В.П.	Розробка клієнтської частини тестувальної системи з фотоконтролем	113
Ковальський С.В., Тужанський С.Є.	Оцінювання та вимірювання успіху освіти з використанням цифрових інструментів	116
Ковтун Б.В., Романюк О.В.	Розробка методу розпізнання суми проплати з чеків різних банків	117

Попри досягнені успіхи, невирішеними залишаються питання стосовно об'єктивності алгоритмів автоматизованої оцінки та їх здатності враховувати індивідуальні особливості студентів. Покращення у цьому напрямку може полягати в розробці більш складних алгоритмів та врахуванні контексту навчання. Крім того, важливо надавати учням засоби для оскарження результатів та отримання зворотного зв'язку.

Висновки

Усі ці аспекти вказують на важливість подальших досліджень у сфері оцінювання в дистанційному навчанні. Забезпечення об'єктивності та чесності в оцінюванні залишається важливим завданням. Подальші дослідження повинні спрямовуватися на розвиток більш точних та надійних алгоритмів оцінювання та забезпечення відкритого спілкування між студентами та викладачами.

З одного боку, оцінка на основі ШІ може забезпечити більш точне, об'єктивне та ефективне оцінювання, звільнення для вчителів/викладачів зосередитись на більш значущому, як наприклад взаємодії з учнями. Він також може ідентифікувати області слабкості та сили учнів, що дозволить викладачам адаптувати методи навчання відповідно індивідуальним потребам. Однак, оцінка на основі ШІ не є панацеєю і не повинна повністю замінити людське судження.

Список використаних джерел

1. Advancing artificial intelligence research – MIT EECS, 2020
2. EURASIA Journal of Mathematics, Science and Technology Education, 2023 – Exploring the potential of AI tools in educational measurement and assessment
3. Artificial Intelligence in Education: Promises and Implications for Teaching and Learning, Darrell M. West, Center for Technology Innovation at Brookings, 2019.

УДК 004.93

КОВТУН Б.В., РОМАНЮК О.В.

Вінницький національний технічний університет

РОЗРОБКА МЕТОДУ РОЗПІЗНАННЯ СУМИ ПРОПЛАТИ З ЧЕКІВ РІЗНИХ БАНКІВ

Анотація. Розроблено метод розпізнання оплати з чеків різних банків. Розроблено блок-схему алгоритму розпізнавання суми оплати з чеків. Наведено фрагмент лістингу програмної реалізації запропонованого методу. Ключові слова: розпізнавання зображень, нейромережі, патерн стратегія.

Розробка методу розпізнання суми оплати з чеків різних банків важлива для автоматизації фінансового обліку та збільшення ефективності бізнесу. Цей метод дозволяє швидко та точно визначити суму, що була сплачена, незалежно від банку, який видав чек. Це полегшує обробку фінансової інформації, зменшує ризик помилок та автоматизує облік. Бізнесам це допомагає зекономити час та ресурси, сприяє точному фінансовому аналізу та забезпечує зручність для клієнтів [1].

Розпізнання тексту на зображеннях чеків на сьогоднішній день є точним як ніколи раніше. Існує велика кількість нейронних мереж, що розпізнають текст на зображеннях з великою точністю. Також нейронні мережі можуть розпізнавати текст для різної величини шрифтів та для різних шрифтів [2]. Але звичайне розпізнання тексту зображень не буде достатньо, оскільки після розпізнання тексту, текст стає перемішаним і втрачає свою логічну структуру. Отримання потрібної інформації з розпізнаного тексту стає практично неможливо, оскільки зазвичай у такому тексті немає унікального патерну для потрібної інформації.

Таким чином, актуальною є задача розробки нового методу для розпізнавання сум проплати з чеків різних банків.

Для підвищення ефективності розпізнання тексту доцільно розпізнавати текст лише у певній області, де потрібна інформація буде мати унікальний патерн. Після розпізнання чеку

зображення лише у певній області буде отримано неструктурований текст, де потрібна інформація буде мати унікальний патерн і цю інформацію можна буде отримати за допомогою Regex.

Недолік розпізнання тексту зображення лише у певній області полягає в тому, що неможливо розпізнати потрібну інформацію на чеках різних банків. Чеки різних банків мають різну структуру тексту і тому потрібна інформація знаходиться в різних областях зображення. Для вирішення цієї проблеми потрібно використати патерн стратегії і для кожного типу чеку банку розробити свою стратегію.

Патерн стратегія – це один з п'яти основних патернів проектування в програмуванні. Він дозволяє визначити набір алгоритмів, упакувати їх у вигляді окремих об'єктів і забезпечити можливість обміну алгоритмами без зміни коду клієнта. Цей патерн розділяє поведінку від контексту і дозволяє вибирати потрібну стратегію в реальному часі. Він сприяє полегшенню розширення і зміни системи, підвищенню її гнучкості і покращенню її читабельності. Патерн стратегія використовується для вирішення завдань, де існує багато різних варіантів алгоритмів і вибір потрібного залежить від контексту [3].

Кожна стратегія буде реалізовувати метод розпізнання необхідної інформації, а та стратегія, що розпізнає найбільшу кількість інформації, буде використовуватись при надсиланні результату розпізнання користувачу.

Для реалізації розпізнання тексту на зображенні доцільно використовувати бібліотеку Tesseract, оскільки вона повністю безкоштовна та забезпечує можливість розпізнання тексту зображення лише у певній області, але можна і використовувати інші бібліотеки розпізнання тексту зображень.

Блок-схема алгоритму розпізнання суми чеку проплати для різних банків зображено на рисунку 1.

1. Початок.
2. Завантажити чек оплати в змінну image.
3. Декларація змінних maxRecognitions та paymentInfoResult.
4. Запустити цикл, що буде перебирати стратегії розпізнання. Якщо стратегії закінчилися – перейти до кроку 18, якщо ні – до кроку 5.
5. Декларація змінної recognizeResult та приствоєння змінній engine об'єкту для розпізнання тексту зображення, а саме бібліотеки Tesseract.
6. Використовуючи стратегію розпізнання, розпізнати кінцевий номер рахунку та присвоїти результат destinationCardNumber.
7. Розпізнати номер квитанції проплати та приствоїти результат checkPaymentId.
8. Розпізнати суму проплати на чеку та записати результат у змінну checkPaymentPrice.
9. Акумуляувати результат розпізнання у змінній checkPaymentInfo.
10. Перевірити чи destinationCardNumber не дорівнює null, якщо так – перейти до кроку 11, якщо ні – до кроку 12.
11. Додати до recognizeResult одиницю.
12. Перевірити чи checkPaymentId не дорівнює null, якщо так – перейти до кроку 13, якщо ні – до кроку 14.
13. Додати до recognizeResult одиницю.
14. Перевірити чи checkPaymentPrice не дорівнює null, якщо так – перейти до кроку 15, якщо ні – до кроку 16.
15. Додати до recognizeResult одиницю.
16. Перевірити чи recognizeResult більше maxRecognitions, якщо так – перейти до кроку 17, якщо ні – до кроку 4.
17. Присвоїти змінній maxRecognitions значення recognizeResult та присвоїти змінній paymentInfoResult значення checkPaymentInfo.
18. Кінець.

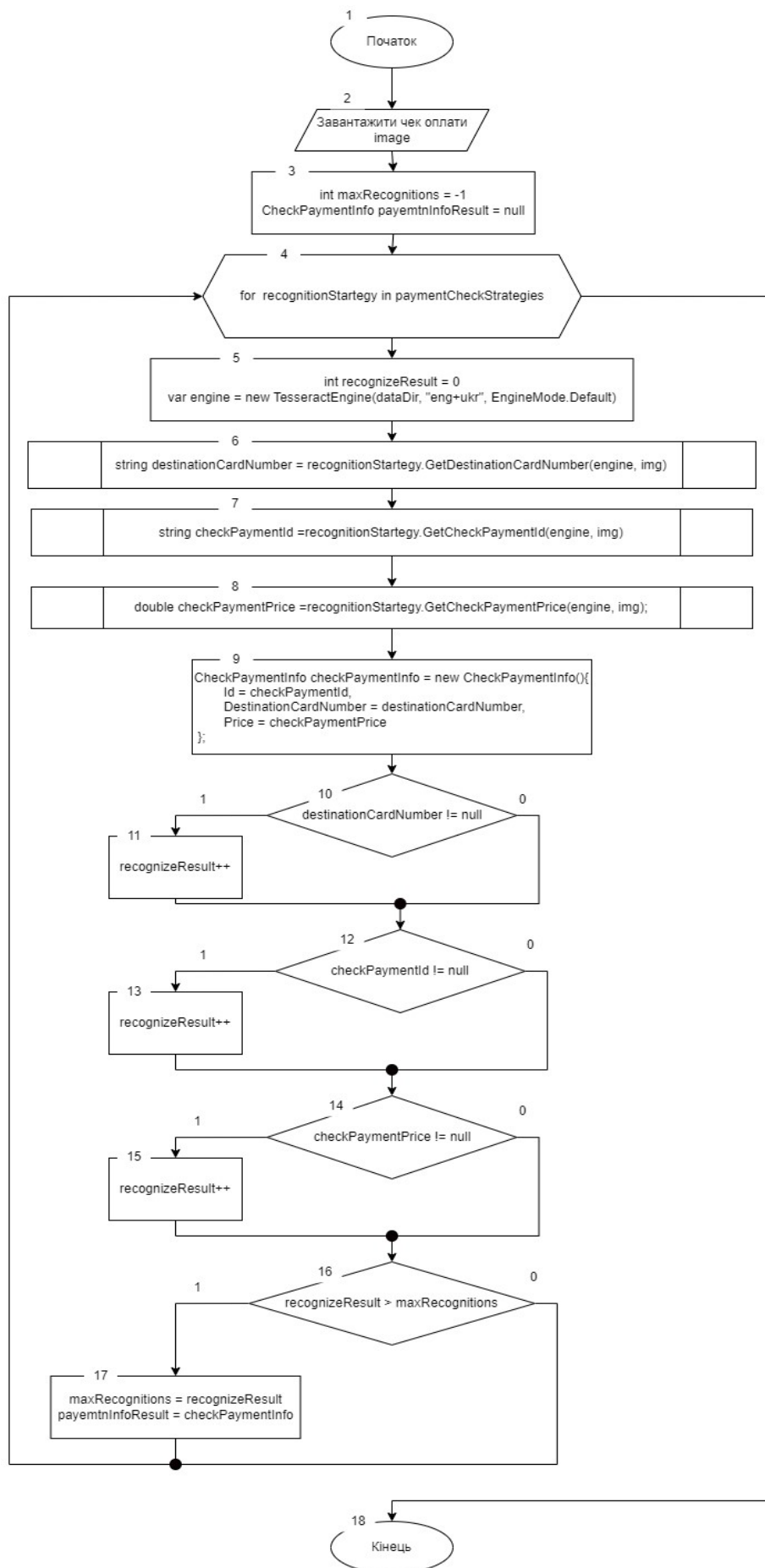


Рисунок 1 – Блок-схема алгоритму розпізнання суми чеку проплати
Розглянемо покроково запропонований алгоритм:

Лістинг коду алгоритму розпізнання суми проплати на чеку для різних банків мовою С# зображено на рисунку 2. Також лістинг коду для стратегії розпізнання суми десктопного чеку проплати для банку «ПриватБанк» мовою С# зображено на рисунку 3.

```
private List<IPaymentCheckStrategy> paymentCheckStrategies = new List<IPaymentCheckStrategy>()
{
    new PrivatBankDesktopPaymentCheckStrategy(),
    new MonoBankDesktopPaymentCheckStrategy(),
};

Ссылка: 2
public CheckPaymentInfo RecognizeCheckPaymentInfo(byte[] image)
{
    CheckPaymentInfo payemtnInfoResult = null;
    int maxRecognitions = -1;
    foreach (var recognitionStartegy in paymentCheckStrategies)
    {
        int recognizeResult = recognitionStartegy.Recognize(image, out CheckPaymentInfo payemtnInfo);
        if (recognizeResult > maxRecognitions)
        {
            payemtnInfoResult = payemtnInfo;
            maxRecognitions = recognizeResult;
        }
    }
    return payemtnInfoResult;
}
```

Рисунок 2 – Лістинг алгоритму розпізнання суми чеку для різних банків

```
Ссылка: 1
public class PrivatBankDesktopPaymentCheckStrategy : IPaymentCheckStrategy
{
    Ссылка: 2
    public int Recognize(byte[] image, out CheckPaymentInfo checkPaymentInfo)
    {
        string dataDir = @"./tesseract";
        // Initialize Tesseract engine
        using (var engine = new TesseractEngine(dataDir, "eng+ukr", EngineMode.Default))
        {
            // Load an image (replace with your image path)
            using (var img = Pix.LoadFromMemory(image))
            {
                string destinationCardNumber = GetDestinationCardNumber(engine, img);
                string checkPaymentId = GetCheckPaymentId(engine, img);
                double checkPaymentPrice = GetCheckPaymentPrice(engine, img);
                checkPaymentInfo = new CheckPaymentInfo()
                {
                    Id = checkPaymentId,
                    DestinationCardNumber = destinationCardNumber,
                    Price = checkPaymentPrice
                };
            };
            int recognitionCount = 0;
            if (destinationCardNumber != null)
            {
                recognitionCount++;
            }
            if (checkPaymentId != null)
            {
                recognitionCount++;
            }
            if (checkPaymentPrice != null)
            {
                recognitionCount++;
            }
            return recognitionCount;
        }
    }
}

Ссылка: 1
private string GetDestinationCardNumber(TesseractEngine engine, Pix img)...

Ссылка: 1
private string GetCheckPaymentId(TesseractEngine engine, Pix img)...

Ссылка: 1
private double GetCheckPaymentPrice(TesseractEngine engine, Pix img)...
```

Рисунок 3 – Лістинг коду стратегії чеку для банку «Приват Банк»

Отже, було розроблено метод розпізнання суми проплати з чеків різних банків. Було визначено, що доцільно використовувати нейронні мережі для розпізнання тексту на зображенні. Для реалізації розпізнання тексту було обрано бібліотеку Tesseract. Також було визначено, що доцільно розпізнавати потрібно інформацію лише у певній області зображення де потрібна інформація буде мати унікальний патерн. Для забезпечення можливості розпізнання суми проплати чеків для різних банків потрібно використовувати патерн стратегії

та для кожного чеку банку потрібно розробляти стратегію. Було наведено лістинг коду реалізації розпізнання суми чеку проплати для різних банків. Також було зображено блок-схему алгоритму розпізнання суми чеку проплати для різних банків.

Список використаних джерел

1. Ковтун В.Б. Розпізнавання чеку проплати за допомогою технології OCR // В.Б. Ковтун, О.В.Романюк / Матеріали XVI міжнародної науково-практичної конференції «Інформаційні технології і автоматизація – 2023», Одеса, 19-20 жовтня 2023 р. - Одеса, Видавництво ОНТУ, 2023 р. – С.348-349.
2. Stephen V. Rice. Optical Character Recognition: An Illustrated Guide to the Frontier. – Springer, 1999. – 203 с.
3. Erich Gamma. Design Patterns: Elements of Reusable Object-Oriented Software. – Addison-Wesley Professional, 1994. – 416 с.

УДК 004.4

КОЖЕВНИКОВ В.С., РОМАНЮК О.В.

Вінницький національний технічний університет

ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ МЕТОДІВ ПОШУКУ ФАЙЛІВ ТА ЇХ УПОРЯДКУВАННЯ

Анотація: У роботі розглянуто питання підвищення ефективності пошуку файлів та їх упорядкування. Запропоновано метод автоматичного створення міток на основі метаданих файлів для спрощення подання запиту пошуку файлів.

Ключові слова: програмне забезпечення, пошук файлів, метод рекурсивного пошуку, метод індексного пошуку, індексування метаданих

Abstract: The paper discusses the study of the effectiveness file search and their ordering. A method of automatically creating labels based on file metadata is proposed to simplify preparing of a file search query.

Keywords: software, file search, recursive search method, index search method, metadata indexing

Пошук та впорядкування файлів – важливий аспект роботи з комп'ютером та організації даних. Це може стосуватися фалів документів, фотографій, музики та інших видів файлів. Підтримувати порядок та забезпечити ефективний пошук, серед великої кількості не впорядкованих даних – це комплекс задач, який може вимагати багато часу, зусиль та наявність відповідних навичок.

Для пошуку файлів застосовується рекурсивний метод пошуку [1]. Це метод, за яким система або програма починає пошук у заданій директорії, а потім рекурсивно переходить до всіх піддиректорій в цій директорії та їхніх піддиректоріях і так далі, до знаходження всіх файлів, що відповідають певним критеріям пошуку. Цей метод дозволяє знайти всі файли, включаючи ті, які знаходяться у вкладених папках. Це ефективний метод пошуку, але при великій кількості файлів та вкладених папок може бути повільним. Кожен виклик функції зумовлює витрату часу на виконання та повернення [2].

Для збільшення ефективності пошуку застосовують індексний метод пошуку. Метод при якому під час пошуку файлів створюється індекс або база даних, яка містить інформацію про місцезнаходження та інші атрибути файлів [3]. Індекс дозволяє значно прискорити процес пошуку, оскільки не потрібно прочитувати або аналізувати кожен файл безпосередньо. Замість цього, можна шукати файли за допомогою індексу, який зберігає важливу інформацію про файли та їх властивості. Для підтримання індексу в актуальному стані треба застосовувати заплановані задачі для сканування файлів на зміни, або слухачі подій змін файлів. На рис. 1 наведено схему пошуку фалів з допомогою міток та індексів пошуку.

Наявність швидкого пошуку це важлива складова, але цього не достатньо для впорядкування невідомих даних. У роботі [3] було запропоновано метод пошуку файлів з використанням інструментів для довготривалого зберігання даних, який дозволяє прискорити час пошуку файлів порівняно з традиційними методами, однак його недоліком є необхідність

**ЕЛЕКТРОННІ ІНФОРМАЦІЙНІ РЕСУРСИ:
СТВОРЕННЯ, ВИКОРИСТАННЯ, ДОСТУП:**

Збірник матеріалів
Міжнародної науково-практичної Інтернет-конференції
20-21 листопада 2023 р.

Редактор С.А.Пойда, М.С. Ніколаєнко
Комп'ютерне верстання С.А.Пойда, М.С. Ніколаєнко

Підписано до друку 15.11.2023 Гарнітура Times New Roman
Формат 60x84/16 Папір офсетний
Друк цифровий Ум. друк. арк. 19,4
Тираж 300 пр. Зам. № 2/23

Видавництво НІКО
м.Суми, вул.Харківська, 54
Свідоцтво про внесення до Державного реєстру
суб'єктів видавничої справи України
серія СМв № 044
від 15.10.2012
E-mail: ms.niko@i.ua
Телефон для замовлень: +38(066) 270-64-68