

Міністерство освіти і науки України

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА
АКАДЕМІЯ»**

Факультет інформатики

Кафедра інформатики

Магістерська робота

Освітній ступінь: магістр

**На тему: «ОПТИЧНЕ РОЗПІЗНАВАННЯ ПОЗНАЧОК ПРИ
ОЦІНЮВАННІ КРЕДИТОСПРОМОЖНОСТІ ФІЗИЧНОЇ ОСОБИ»**

Виконав: студент 2 року навчання

Спеціальності

121 Інженерія програмного забезпечення

Білокінь Данило Тарасович

Керівник

Нагірна А.М.

кандидат фіз.-мат. наук

Рецензент _____

Магістерська робота захищена

з оцінкою _____

Секретар ЕК С.А. Мелешенко

«___» _____ 2023

Київ 2023

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА
АКАДЕМІЯ»

Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ

Зав.кафедри інформатики,
к.ф.-м.н.

_____ С. С.Гороховський
(підпис)

„____” _____ 2022 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на магістерську роботу

студента 2 р.н. магістерської програми Інженерія програмного забезпечення Білокія Данила Тарасовича

Оптичне розпізнавання позначок при оцінюванні кредитоспроможності фізичної особи.

Зміст текстової частини до магістерської роботи:

Зміст

Анотація

Вступ

1 Дослідження предметної області

2 Методи та засоби розв’язання поставлених завдань

3 Опис реалізації програми

Висновки

Список літератури

Додатки

Дата видачі „____” листопада 2022 р.

Керівник

(підпис)

Завдання отримав

(підпис)

Тема: “Оптичне розпізнавання позначок при оцінюванні кредитоспроможності фізичної особи”

Календарний план виконання роботи:

/п	Назва етапу дипломного проекту (роботи)	Термін виконання етапу	Примітка
1.	Отримання завдання на дипломну роботу.	02.11.2022	
2.	Огляд технічної літератури за темою роботи.	25.11.2022	
3.	Виконати аналіз існуючих методів розпізнавання позначок	25.11.2022	
4.	Програмування розробленого алгоритму	15.01.2022	
5.	Застосування розробленого алгоритму для розпізнавання анкет	15.02.2022	
6.	Написання пояснювальної роботи.	01.04.2023	
7.	Створення слайдів для доповіді та написання доповіді.	14.05.2023	
8.	Аналіз отриманих результатів з керівником, написання доповіді та попередній захист магістерської роботи.	15.05.2023	
9.	Корегування роботи за результатами попереднього захисту.	18.05.2023	
10.	Остаточне оформлення пояснювальної роботи та слайдів.	05.06.2023	
11.	Захист магістерської роботи (проекту)	13.06.2023	

Студент **Білокінь Данило Тарасович**

Керівник **Нагірна Алла Миколаївна**

“ _____ ”

АНОТАЦІЇ

Білокінь Данило Тарасович, кваліфікаційна робота магістра на тему: “Оптичне розпізнавання позначок при оцінюванні кредитоспроможності фізичної особи”. Напрямок підготовки 121 “Інженерія програмного забезпечення”, НаУКМА, Київ 2023. В даній кваліфікаційній роботі магістра проводиться дослідження процесу оптичного розпізнавання позначок, методи та способи розв’язання поставленої задачі та необхідні інструменти. Розробляється програмне забезпечення, яке буде генерувати таблиці для відповідей, оптично розпізнавати зображення та оцінювати кредитоспроможність фізичної особи на основі анкет. Результат роботи програмного забезпечення будуть виводитись в вигляді файлу з виділеними відповідями і оцінкою на основі відповідей і коефіцієнтів важливості питань.

Ключові слова:

Оптичне розпізнавання позначок, оцінювання кредитоспроможності, фізична особа, OCR, OMR, QR коди, OpenCV, шаблони розпізнавання, методи дескрипторів, машинне навчання, комп’ютерне зорове сприйняття, алгоритми обробки зображень, зчитування даних, класифікація, точність розпізнавання, навчальний набір даних, препроцесинг зображень, сегментація, локалізація об’єктів, відповідність шаблону, аналіз результатів, автоматизація процесу, ефективність.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	6
ВСТУП	7
1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ	10
1.1 Поняття та методи оцінюванні кредитоспроможності фізичної особи	10
1.2 Анкетування як метод оцінюванні кредитоспроможності фізичної особи	11
Висновки до розділу 1	12
2. МЕТОДИ ТА ЗАСОБИ РОЗВ'ЯЗАННЯ ПОСТАВЛЕНИХ ЗАВДАНЬ.....	13
2.1 Оптичне розпізнавання знаків	13
2.2 Алгоритми і методи оптичного розпізнавання позначок	14
2.3 Шаблони в оптичному розпізнаванні позначок.....	16
2.4 Маркери для накладання шаблонів.....	17
2.5 Оптичне розпізнавання позначок при оцінюванні кредитоспроможності.....	18
Висновки до розділу 2	19
3. ОПИС РЕАЛІЗАЦІЇ ПРОГРАМИ	20
3.1 Формування та аналіз вимог.....	20
3.2 Проектування програмного забезпечення.....	24
3.3 Генерація сітки з відповідями для документу анкети.....	27
3.4 Алгоритм розпізнавання місцезнаходження сітки з відповідями ...	28
3.5 Алгоритм визначення відповідей респондента	30
3.6 Тестування	33
3.7 Використані технології.....	37
3.8 Напрямки для вдосконалення.....	39
Висновки до розділу 3	40
ВИСНОВКИ	41
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	42

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

1. ПЗ – програмне забезпечення.
2. ООП – об’єктно-орієнтоване програмування.
3. OCR – Optical Character Recognition.
4. API – Application Public Interface.
5. LBP – локальні бінарні шаблони.
6. SVM – метод опорних векторів.
7. SIFT – Scale-Invariant Feature Transform.
8. QR – quick response.
9. CNN – згорткові нейронні мережі.
10. WPF – Windows Presentation Foundation.
11. UI – User Interface – інтерфейс користувача.
12. XML - eXtensible Markup Language.
13. OOXML – Office Open XML.
14. ОС – операційна система.
15. RAIІ – Resource Acquisition Is Initialization.
16. SDK – software development kit – комплект для розробки програмного забезпечення.

ВСТУП

Актуальність теми

Оцінка кредитоспроможності фізичних осіб є однією з ключових задач у сучасній фінансовій системі. Банки, фінансові установи та інші кредитні організації мають великий інтерес у визначенні ризиків, пов'язаних з видачею кредитів, а також в розрахунку вірогідності повернення вкладених коштів вчасно і безперешкодно. Для досягнення цих цілей використовуються різні методи та підходи, зокрема оптичне розпізнавання позначок, яке витікає з швидкого розвитку сучасних технологій, особливо в галузі штучного інтелекту і комп'ютерного зору.

Традиційні методи оцінки кредитоспроможності, такі як аналіз фінансової інформації та кредитний історія, мають свої обмеження. Вони можуть бути часом та працевитратними, а також залежати від достовірності наданих даних. Оптичне розпізнавання позначок може забезпечити автоматизований та ефективний процес перевірки та аналізу документів, що сприятиме швидкому та точному оцінюванню кредитоспроможності фізичних осіб.

Отже, актуальність теми зумовлена потребою у швидкому, точному та автоматизованому процесі збору та аналізу інформації, необхідної для оцінки кредитоспроможності. Використання оптичного розпізнавання позначок може покращити ефективність і надійність процесу, сприяючи прийняттю кращих рішень у сфері кредитування.

Мета і завдання досліджень

Мета:

Дослідження і розробка системи оптичного розпізнавання позначок для оцінювання кредитоспроможності фізичних осіб з використанням технологій обробки зображень.

Завдання досліджень:

Проаналізувати поточний стан існуючих методів оцінювання кредитоспроможності фізичних осіб та існуючих систем оптичного розпізнавання позначок.

Розробити алгоритм для оптичного розпізнавання позначок, що дозволить визначати кредитоспроможність фізичної особи.

Зібрати відповідний набір даних, що містить зображення позначок та відповідну інформацію про фізичних осіб.

Реалізувати систему оптичного розпізнавання позначок на основі обробки зображень.

Об'єкт і предмет досліджень

Об'єкт досліджень:

Оптичне розпізнавання позначок при оцінюванні кредитоспроможності фізичної особи.

Предмет досліджень:

Алгоритми та системи оптичного розпізнавання позначок, засновані на обробці зображень

Практична значимість отриманих результатів

Отримані результати дозволяють програмно реалізувати систему оптичного розпізнавання позначок в анкетах, використаних для оцінювання кредитоспроможності фізичних осіб.

1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Поняття та методи оцінюванні кредитоспроможності фізичної особи

Оцінювання кредитоспроможності фізичної особи є важливим процесом для фінансових установ, таких як банки, кредитні організації і кредитні кооперативи. Цей процес дозволяє визначити, наскільки ймовірно, що фізична особа буде здатна виплачувати кредитні зобов'язання вчасно і в повному обсязі.

Оцінка кредитоспроможності зазвичай здійснюється на основі різних методів та критеріїв. Установи збирають і аналізують інформацію про фінансовий стан та платоспроможність клієнта, включаючи кредитну історію, доходи та інші фактори.

Один з ключових аспектів для прийняття рішення - це аналіз кредитної історії фізичної особи, на основі звітів, попередніх кредитів, інформації про кредитні картки, можливих способів погашення зобов'язань та наявності прострочених платежів. Також враховуються дохід та зайнятість клієнта: перевіряють джерела доходу та стабільність зайнятості особи, розраховується співвідношення зобов'язань до доходів для визначення платоспроможності. Крім того, враховується інформація про майновий стан та фінансові зобов'язання. Банки можуть запитати про наявність активів, таких як нерухомість або автомобіль, відкриті іпотеки або інші кредити.

Комплексне оцінювання кредитоспроможності особи допомагає приймати рішення щодо надання кредиту, визначення його розміру та встановлення відповідних умов, таких як процентна ставка та строк погашення. Мета цього процесу - забезпечити як клієнтів, так і фінансові установи оптимальними умовами для успішного виконання зобов'язань.

Основні методи оцінювання кредитоспроможності особи включають:

1. **Аналіз кредитного звіту:** Фінансові установи можуть замовити кредитний звіт, який містить інформацію про кредитну історію. Кредитний звіт надає відомості про кредитні картки, позики, кредитні зобов'язання та спосіб погашення кредитів.
2. **Доходи та зайнятість:** Кредитори оцінюють дохід та стабільність зайнятості клієнта. Можуть бути розглянутими документи, що підтверджують дохід (наприклад, податкові декларації, платіжні витяги, трудові книжки, тощо) для підтвердження спроможності особи виплачувати кредит зі свого поточного рівня доходів.
3. **Вартість житла:** При заявці на іпотечний кредит, банки можуть оцінювати вартість нерухомості, яку фізична особа планує придбати. Вони враховують ринкову вартість нерухомості, поточну іпотечну ставку та фінансові здатності.
4. **Матеріальний стан:** Банки аналізують матеріальний стан особи, враховуючи вже існуючі кредитні зобов'язання перед іншими установами, наявність активів, вкладів, майна та інших фактори, що впливають на забезпеченість людини.

1.2 Анкетування як метод оцінюванні кредитоспроможності фізичної особи

Анкетування є одним з методів оцінювання кредитоспроможності фізичної особи. При цьому банк або фінансова установа надсилає клієнту анкету або частіше проводить особистий інтерв'ю для отримання додаткової інформації про його фінансовий стан та здатність погасити кредит.

Анкета може включати наступні питання:

1. **Особисті дані:** ім'я, адреса, контактна інформація.
2. **Зайнятість:** інформація про місце роботи, посаду, стаж, рівень доходів.

3. Фінансовий стан: джерела доходів, наявність заощаджень, активів, зобов'язань, іпотеки, кредитних карток.
4. Кредитна історія: попередні кредити, їх розмір, термін погашення, наявність прострочених платежів, банкрутства або інших фінансових проблем.
5. Витрати: орієнтовані витрати на житло, транспорт, їжу, медичне обслуговування та інші регулярні витрати.
6. Інші фактори: сімейний стан, наявність дітей, стан здоров'я тощо.

Результати анкетування можуть вплинути на рішення щодо надання кредиту, його розміру та умов. Варто зауважити, що анкетування може бути трудомістким процесом, потребуючим втручання співробітників для проведення анкетування, подальшого аналізу і підрахунків результатів.

Висновки до розділу 1

Проаналізувавши предметну область, було обрано метод анкетування, як найбільш зручніший метод для вирішення поставлених задач. Зі сторони оцінювання кредитоспроможності була виявлена наступна послідовність дій: на основі заповненої анкети кредитор аналізує отриману інформацію, проводить внутрішній підрахунок, враховуючи всі ризики та порівнює її зі своїми критеріями достатності кредитоспроможності.

2. МЕТОДИ ТА ЗАСОБИ РОЗВ'ЯЗАННЯ ПОСТАВЛЕНИХ ЗАВДАНЬ

2.1 Оптичне розпізнавання знаків

Оптичне розпізнавання знаків (OCR) - це процес автоматичного визначення, розпізнавання та інтерпретації тексту на зображень, що містять позначки або символи. OCR-системи використовуються для перетворення тексту, написаного або надрукованого на фізичних документах (паперові документи, книги, чеки, паспорти, фотографії) в цифровий формат, який може бути оброблений комп'ютером.

Процес OCR зазвичай включає кілька етапів. Спочатку зображення аналізується для виявлення текстових областей, які потенційно містять позначки. Потім розпізнавач визначає індивідуальні символи або групи символів у цих областях. Нарешті, розпізнані символи перетворюються на текстовий формат, який може бути збережений або переданий для подальшої обробки і роботи з ним.

OCR-технологія має широкий спектр застосувань. Вона використовується для автоматичного розпізнавання автомобільних номерних знаків, сканування та конвертації книг у електронний формат, перетворення рукописного тексту на комп'ютерно-читабельний формат, обробки чеків, банківських документів, накладних, а також для автоматичного розпізнавання позначок на фотографіях або відео. [10]

Оптичне розпізнавання знаків використовується в багатьох сферах, де потрібно швидко і точно обробляти великий обсяг інформації, що міститься на фізичних документах або зображеннях.

Прикладами реальних продуктів оптичного розпізнавання символів є:

1. Tesseract - один із найпопулярніших OCR-двигунів з відкритим вихідним кодом. Він розроблений Google і підтримується

спільнотою розробників. Tesseract використовується для розпізнавання тексту у багатьох додатках, включаючи сканери документів, системи електронного документообігу та розпізнавання автомобільних номерних знаків.[2]

2. Abbyy FineReader Engine - комерційний OCR-двигун, розроблений компанією Abbyy. Він забезпечує високу точність розпізнавання тексту та підтримує багато мов. Abbyy FineReader Engine використовується в різних сферах, включаючи автоматичне розпізнавання позначок у документах, системи електронного архівування та розпізнавання рукописного тексту.
3. OCRopus - ще один відкритий OCR-двигун, який використовується для розпізнавання тексту. Він базується на алгоритмах та методах машинного навчання. OCRopus використовується в проектах з документації, бібліотеках та дослідницьких роботах з обробки тексту.
4. Microsoft OCR API - хмарний OCR-двигун, розроблений компанією Microsoft. Цей сервіс надає можливість виконувати оптичне розпізнавання позначок через API-запити до хмарної інфраструктури Microsoft. Він може бути використаний в додатках, що потребують розпізнавання тексту з фотографій і зображень.

2.2 Алгоритми і методи оптичного розпізнавання позначок

Існує багато алгоритмів та методів роботи з OCR, які застосовуються на певних етапах роботи з оптичним розпізнаванням, основними із них можна виділити:

1. Перетворення зображення - цей етап включає підготовку зображення, таку як видалення шуму, вирівнювання, бінаризація та зменшення розміру. Етап може включати в себе наступні методи:

Фільтрація шуму за допомогою медіанного фільтру або фільтра Гауса, підвищення контрастності за допомогою гістограмного розтягнення або адаптивного рівня сірого.[15]

2. Сегментація - етап на якому проводиться розділення зображення на окремі області, які відповідають окремим символам або компонентам. Сегментація може бути заснована на порогованні, з'єднанні компонентів або використанні алгоритмів виявлення контуру, такі як алгоритм Кенні або алгоритм Кітту. Також, методи засновані на детекторах країв, наприклад - Собел.[15]
3. Витягування ознак - після сегментації кожен окремий символ або компонент потрібно описати за допомогою певних ознак. Це можуть бути такі ознаки, як форма, текстурні характеристики, структура ліній. Різні методи витягування ознак можуть використовувати фільтри Габора, густинні градієнти, локальні бінарні шаблони (LBP) для опису текстурних характеристик, алгоритм скелетизації для визначення форми символів та інші.[15]
4. Класифікація - етап, що відповідає за проведення класифікація окремих символів на основі їх витягнутих ознак. Могуть застосовуватися методи машинного навчання, такі як метод опорних векторів (SVM), нейронні мережі (наприклад: згорткові нейронні мережі), методи кластеризації, статистичні моделі і т.д.
5. Пост-обробка - після класифікації можуть бути застосовані методи пост-обробки для поліпшення результатів OCR. Це може включати корекцію помилок різними методами, такими як метод Левенштейна для виправлення невідповідностей індивідуальних символів. Використання контекстної інформації або моделей мови для покращення розпізнавання цілісних слів або фраз, виявлення та коригування спотворень та відновлення пропущених символів.

Також, методи які потребують окремої уваги незалежно від етапів роботи з оптичним розпізнаванням:

1. Шаблонне співставлення - метод використовує зразки або шаблони позначок, які порівнюються зі зображеннями для виявлення і розпізнавання позначок. Шаблони можуть бути надані наперед або вибрати зображення позначки для порівняння.
2. Локалізація та сегментація - методи дозволяють виділити області зображень, де розташовані позначки. Вони можуть використовувати алгоритми детекції об'єктів або методи, які використовують границі або інші характеристики позначок для їх виявлення.
3. Методи дескрипторів - методи, що використовують особливі описи або характеристики позначок для їх розпізнавання. Наприклад, метод SIFT (Scale-Invariant Feature Transform) створює дескриптори позначок, які потім можуть бути порівняні з дескрипторами на вхідних зображеннях.
4. QR-коди - для розпізнавання QR-кодів існують спеціалізовані алгоритми, які використовують методи бінаризації, детекції маркерів та декодування для виявлення та інтерпретації QR-кодів.

2.3 Шаблони в оптичному розпізнаванні позначок

В оптичному розпізнаванні позначок шаблони використовуються для виявлення і розпізнавання конкретних символів або патернів у зображеннях. Шаблони можуть бути використані для розпізнавання цифр, літер, спеціальних символів або навіть повних слів. Існує кілька підходів до використання шаблонів в OCR, і основні з них включають:

Порівняння шаблонів: в цьому підході створюються шаблони для кожного символу або патерну, які потім порівнюються зі зображенням, яке потрібно розпізнати. Порівняння може включати обчислення схожості шаблону з частинкою зображення за допомогою алгоритмів, таких як кореляція або шаблонне співставлення.[12]

Машинне навчання: використання методів машинного навчання, наприклад - згорткові нейронні мережі (CNN), для навчання системи OCR

розпізнавати символи на основі набору попередньо підготовлених зображень та шаблонів. Так система навчається визначати закономірності та ознаки, що відрізняють один символ від іншого і його положення.

Використання геометричних функцій: зображення символів можуть бути представлені як набір геометричних функцій, таких як кількість кутів, пряма, крива та інші. Ці функції можуть використовуватися для розпізнавання символів шляхом порівняння їх з відомими шаблонами або маркерами на документах.

Використання шаблонів у OCR може бути поєднано з іншими методами і алгоритмами для досягнення більш точного та ефективного розпізнавання символів. Швидким простим методом для покращення накладання шаблонів є маркування.

2.4 Маркери для накладання шаблонів

Маркери для накладання шаблонів визначають області на зображенні, де очікується наявність конкретних символів.

Розберемо декілька із найрозповсюдженіших маркерів для накладання шаблонів є:

Прямокутні маркери - використовуються для виділення областей зображення, які містять символи або текст. Прямокутний маркер визначає верхню ліву та нижню праву точки прямокутника, який охоплює область з цільовим текстом.

Овальні маркери - використовуються для виділення символів, які мають овальну або круглу форму. Вони визначають центр символу та його радіус або половину діагоналі.

Мультиполігони - дозволяють вибрати складні форми символів або групи символів. Вони складаються з набору точок, які визначають контур символу.

Маркери на основі ключових точок - використовуються для визначення особливих ключових точок на символах або тексті. Наприклад, можна

використовувати маркери на основі кутів, кінців ліній або вигинів для визначення положення символів.

Ці маркери можуть бути використані для побудови шаблонів, які в подальшому використовуються для пошуку і розпізнавання позначок. Вибір певного типу маркера на пряму залежить від розташування, потреб і характеру позначок, які необхідно розпізнати.

2.5 Оптичне розпізнавання позначок при оцінюванні кредитоспроможності

Процес використання OCR для оцінювання кредитоспроможності, за аналогію подібних задач, має включати такі кроки:

1. Підготовка анкет - анкети створюються заздалегідь з позначками, які представляють певні відповіді або параметри, пов'язані з оцінкою кредитоспроможності. Наприклад, можуть бути позначки для категорій доходу, кредитної історії, зайнятості тощо.
2. Сканування анкет - анкети скануються за допомогою оптичного сканера, що перетворює їх на цифровий формат. Можливий варіант з фотографуванням анкет, але його надійність та доцільність має бути протестована.
3. Розпізнавання позначок - програмне забезпечення аналізує цифрове зображення анкети та розпізнає позначки, які були зроблені на аркуші під час заповнення анкети. Це може включати визначення наявності або відсутності позначки, а також аналізу відповідних параметрів.
4. Обробка результатів - після розпізнавання позначок, отримані дані можуть бути оброблені для оцінювання кредитоспроможності. Це може включати обчислення сум та відсотків, встановлення категорій, коефіцієнтів певних відповідей або оцінок, аналіз згідно з заданими правилами тощо.

Висновки до розділу 2

Використання OCR дозволяє автоматизувати процес обробки анкет і значно прискорити оцінювання кредитоспроможності. Воно зменшує ризик людських помилок, які можуть виникнути при ручному введенні даних з анкет. Проте, точність розпізнавання позначок залежить від якості сканування, чистоти аркушів та інших факторів, тому важливо використовувати якісне обладнання та правильно налаштовувати процес роботи для досягнення найкращих результатів.

3. ОПИС РЕАЛІЗАЦІЇ ПРОГРАМИ

3.1 Формування та аналіз вимог

Первинні вимоги:

- Створене програмне забезпечення оптичного розпізнавання позначок повинно обробляти зображення і видавати оцінку кредитоспроможності особи на основі розпізнаних результатів пройденої анкети.

Детальні вимоги:

- Користувач повинен мати спосіб обрати шлях до файлу текстового документу з зіставленою анкетною;
- Користувач повинен мати можливість задавати правильні або не правильні відповіді на запитання в анкеті;
- Користувач повинен мати можливість обрати коефіцієнти важливості для запитань в анкеті;
- Користувач повинен обрати шлях до файлу текстового документу з заповненою анкетною;
- Користувач повинен обрати шлях для збереження вихідного файлу;
- Користувач повинен мати змогу відразу побачити результат анкетування.

Функціональні вимоги:

- Програмне забезпечення повинно приймати файли в форматах .docx для генерації таблиці;
- Програмне забезпечення повинно приймати файли в форматах .jpg, .jpeg, .png для розпізнавання анкети;
- Програмне забезпечення повинно знаходити таблицю відповідей на зображенні за допомогою маркерів;

- Програмне забезпечення повинно розпізнавати в будь-якому виді позначки в межах комірок таблиці;
- Програмне забезпечення повинно виділяти комірки, які розпізнані відповіддю;
- Програмне забезпечення повинно підрахувати фінальну оцінку на основі правильних і неправильних відповідей та коефіцієнтів важливості запитань;
- Програмне забезпечення повинно зберігати файл опитування з виділеними відповідями і кінцевою оцінкою на зображенні за вказаним місцем користувача;
- Програмне забезпечення повинно експортувати кінцевий файл в форматі .png;
- Програмне забезпечення повинно відображати кінцеву оцінку опитування на екрані застосунку.

Нефункціональні вимоги:

- Кінцевий продукт повинен буде скомпільований в файл формату .exe;
- Кінцевий продукт повинен працювати на операційній системі Windows без встановлення додаткового програмного забезпечення, бібліотек, модулів або розширень;
- Кінцевий продукт повинен мати інтуїтивний та лаконічний користувацький інтерфейс;
- Час виконання кінцевого продукту повинен бути оптимізований і швидкодіючий.

Організаційні вимоги:

- Продукт повинен бути розроблений мульти-парадигмовою мовою C# включно з використанням додаткових бібліотек;
- Інтерфейс повинен бути розроблений на Windows Presentation Foundation або Windows Forms UI-фреймворках.

Зовнішні вимоги:

- Програмне забезпечення не повинно збирати, зберігати або розповсюджувати будь які дані (конфіденційну інформацію, дані комп'ютера, дані про користувача, дані про результати анкетування, дату, час та геолокацію);
- Програмне забезпечення не повинно порушувати авторські права, патенти, права власності та будь-які закони актуального законодавства України.

Аналіз вимог за допомогою діаграми прецедентів:

Діаграма прецедентів або Use Case Diagram - це графічне зображення можливих взаємодій користувача з системою. Діаграма варіантів використання показує різні варіанти використання та різні типи користувачів системи і часто супроводжується діаграмами інших типів. Прецеденти (варіанти використання) позначаються колами або еліпсами, а актори (користувачі) - значком людини.

Проаналізувавши дані вимоги та основні процеси програмного забезпечення, були побудовані діаграми, за допомогою яких було охоплено всі існуючі функції та повний список можливостей використання продукту.

За діаграмою прецедентів (*Рисунок 3.1*), на початку роботи з застосунком, користувач повинен мати вибір між режимами роботи: генерація таблиці для зіставленої анкети чи розпізнавання відповідей-позначок.

Якщо обраний режим – генерація таблиці, користувач повинен вказати шлях до документу анкети за допомогою впливаючого вікна-форми вибору розташування файлу. Наступний крок – заповнення сітки коефіцієнтів анкети. Користувач вводить значення відповідно до кожного із запитань і варіантів відповіді в таблицю полів введення. Після цього користувач повинен обрати рівень прохідного балу – мінімальна оцінка достатня для вдалого проходження анкети. Значення задається в полі введення. Після виконання всіх кроків цього

варіанту взаємодії з застосунком необхідно зберегти встановлені значення за натисканням кнопки.

Якщо обраний режим – розпізнавання відповідей, користувач повинен вказати шлях до зображення, що має бути розпізнаним. Після цього вказати шлях для збереження результату – вже розпізнаної анкети, з виділеними позначками в таблиці і підрахованим результатом. Для обох кроків використовується впливаюче вікно-форма вибору розташування файлу. В першому випадку задається сам файл, а в другому – папка. Наступним кроком користувач повинен задати ім'я файлу з результатом розпізнавання. Ім'я задається в полі введення. Процес розпізнавання повинен запускатись по натисканню кнопки.

Фінальним кроком роботи з застосунком, незалежно від обраного варіанту взаємодії, повинне бути ознайомлення з результатами – зображенням, яке повинно відкриватись в новому вікні програми.

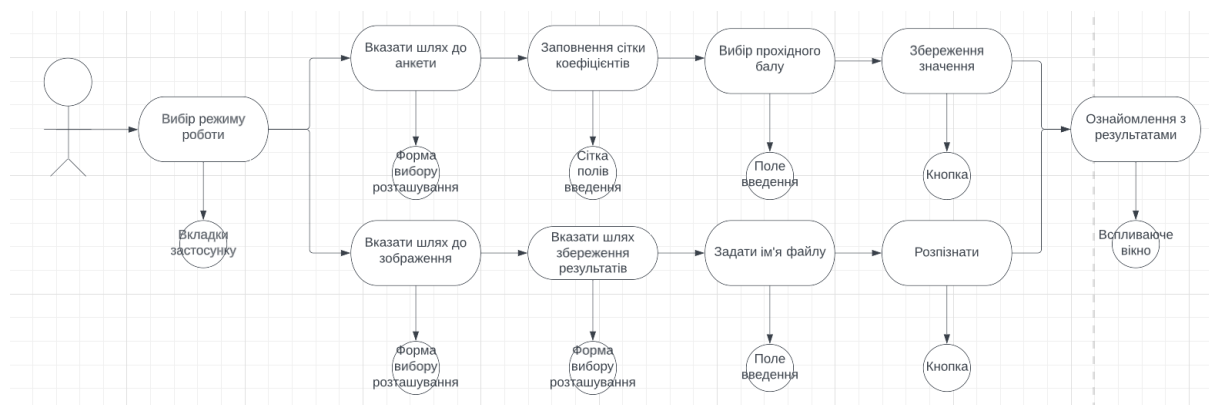


Рисунок 3.1 - Діаграма прецедентів

3.2 Проектування програмного забезпечення

Діаграма діяльності:

На діаграмі (Рисунок 3.2) зображена очікувана послідовність дій користувача при роботі над визначенням балу кредитоспроможності особи. На вибір альтернативного шляху впливає чи вже готова і заповнена анкета для розпізнавання. Дії користувача відрізняються на генерацію сітки, визначення коефіцієнтів і проведення анкетування. Важливо зауважити, проведення анкетування не покривається роботою програмного забезпечення що розробляється.

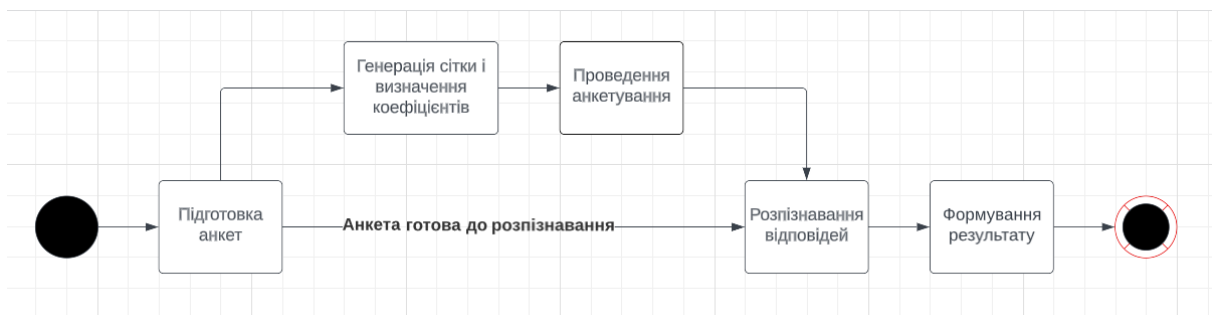


Рисунок 3.2 - Діаграма діяльності

Діаграма скінченного автомату:

На діаграмі (Рисунок 3.3) зображена скінченна кількість станів в якій може знаходитись система та переходи між ними. На стілочках позначено що спричиняє перехід від одного стану, до іншого.

Діаграма починається з початкового псевдостану і переходить в запуск додатку, а з якого в стан введення всіх необхідних даних, і ініціалізує перевірку на їх коректність. Якщо дані коректні, починається процес розпізнавання. Якщо дані некоректні, виводиться відповідне повідомлення про помилку, яка може бути критичною або некритичною. В разі критичної помилки стан відразу переходить в кінцевий, в іншому випадку починається процес обробки. Після формування і збереження кінцевого файлу, настає стан очікування, в якому можна продовжити роботу, повернувшись до стану введення даних або закінчити роботу – перейти в кінцевий стан.

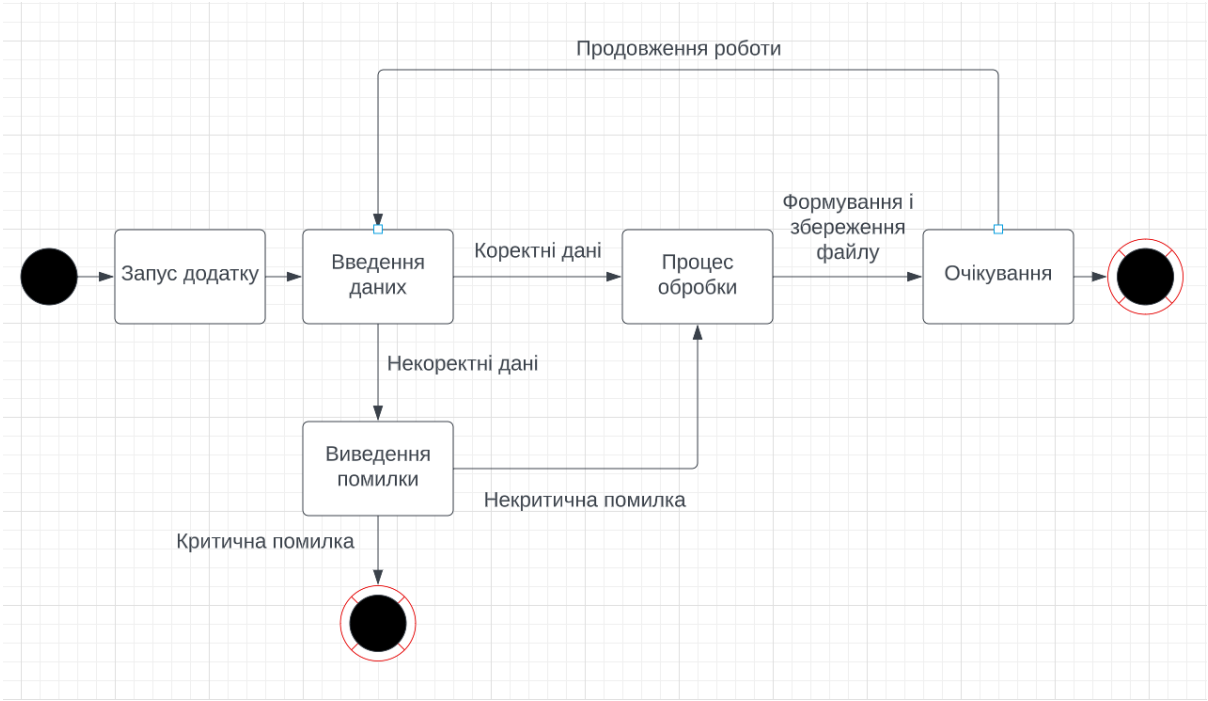


Рисунок 3.3 - Діаграма скінченного автомату

Розробка інтерфейсу:

Інтерфейс був побудований лаконічним та достатньо дружнім для користувача (user-friendly).

Для побудови макету було використано Figma.

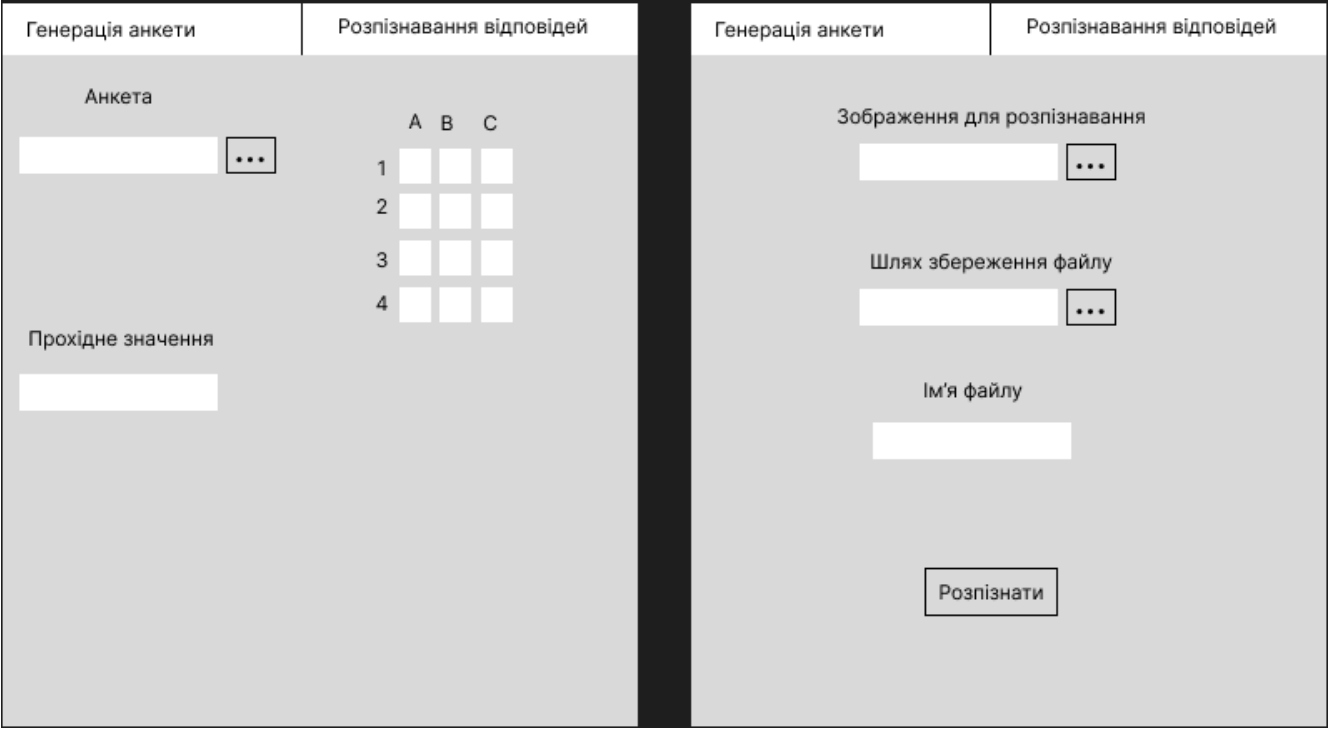


Рисунок 3.4 - Макет інтерфейсу

Інтерфейс був побудований на основі WPF. Основними блоками було використано стандартний набір інструментів, а саме: TextBox, Button, Label, TabControl. Для впливаючих вікон вибору розміщення файлів був обраний Microsoft.Win32.OpenFileDialog.

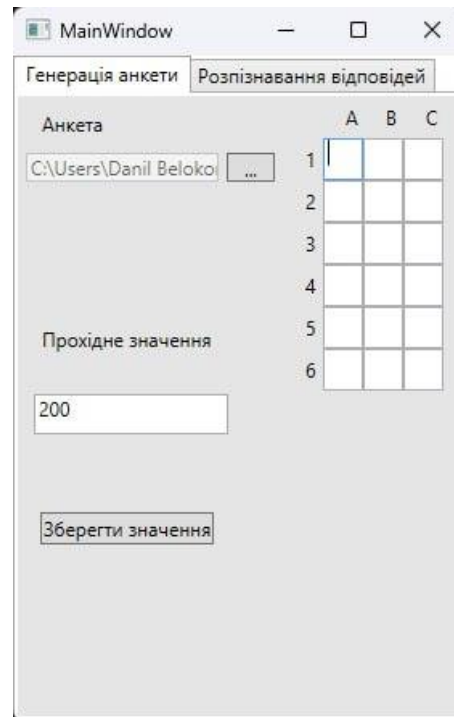


Рисунок 3.5 - Сторінка інтерфейсу "Генерація анкети"

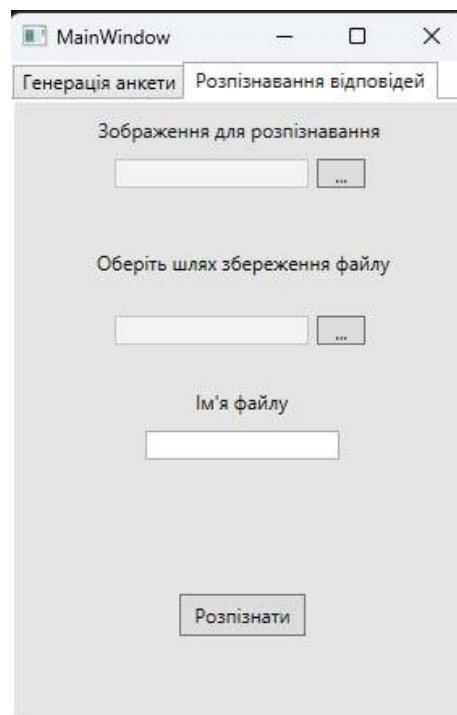


Рисунок 3.6 - Сторінка інтерфейсу "Розпізнавання відповідей"

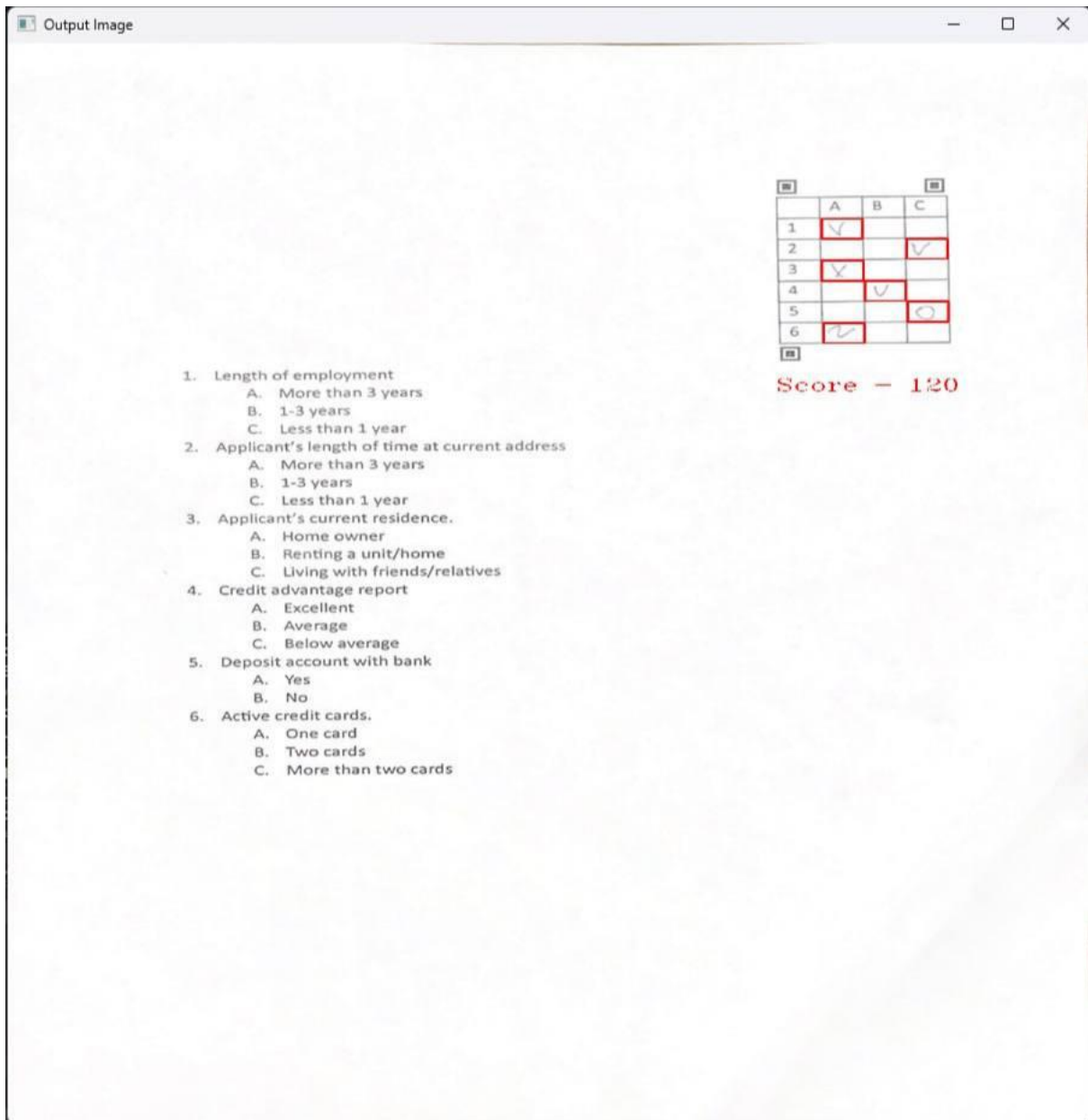


Рисунок 3.7 - Виведення результату в застосунку

3.3 Генерація сітки з відповідями для документу анкети

В програму імпортується файл документ в форматі .docx з визначеними питаннями та варіантами відповідей у форматі нумерованих списків. Питання позначаються цифрами, варіанти відповідей - латинськими буквами.

Використовуючи бібліотеку OpenXml, програмне забезпечення сканує вузли структури документу і визначає елементи нумерованого списку. Після

визначення рівню елементу списку на основі цього значення визначається чи є обраний елемент списку питанням чи варіантом відповіді. Значення кількості запитань передається в функцію побудови сітки і задає кількість рядків. Значення максимальної кількості варіантів відповідей для одного питання також передається в функцію побудови і задає кількість колонок в сітці.

На основі отриманих даних про кількість питань та варіантів відповідей будується таблиця відповідного розміру в яку додається рядок та колонка з позначенням номеру питання та умовного позначення варіанту відповіді. В таблицю додаються шаблони для знаходження на сторінці, які генеруються автоматично за рахунок створення вкладених квадратів, один поверх одного.

3.4 Алгоритм розпізнавання місцезнаходження сітки з відповідями

Для розпізнавання місцезнаходження сітки з результатами анкети був обраний принцип пошуку маркерів QR коду за допомогою бібліотеки OpenCV.

QR-коди (Quick Response codes - коди швидкого реагування) — двовимірні штрих-коди, які складаються з чорних квадратів, розташованих на білому тлі. Чорні квадрати утворюють моделі позицій, які допомагають ідентифікувати та декодувати інформацію. Маркери позицій у QR-коді включають три характерні квадратні візерунки, які називаються шаблонами пошуку (Finding Pattern). Вони розташовуються в трьох кутах коду. Крім того, існує менший квадратний маркер - шаблон вирівнювання (Alignment Pattern), який розташовується біля шаблонів пошуку.

Основними шаблонами в QR-кодах є:

Finding Pattern:

У QR-коді є три шаблони пошуку, які розташовані у верхньому лівому, верхньому правому та нижньому лівому кутах. Кожен із маркерів складається з чорного заповненого квадрату, в який вкладено менший білий і менший чорний

квадрати. Вони допомагають пристрою або застосунку виявити, зчитати і правильно розшифрувати код.

Важливо зауважити, що маркери розташування в QR-коді відповідають певному макету та мають фіксовані розміри та пропорції.[16]

Alignment Pattern:

Шаблон вирівнювання — менший квадратний маркер, розташований біля шаблонів пошуку. Виглядає як заповнений чорний квадрат з вкладеним всередині білим заповненим квадратом і відцентрованою чорною крапкою всередині. Використовується для QR-кодів з вищою роздільною здатністю, щоб забезпечити точне декодування та сканування.

Timing Patterns:

Шаблони синхронізації не є маркерами позиції в тому ж сенсі, що шаблони пошуку та вирівнювання. Вони складаються з чергування чорних і білих пікселів, що розташовані горизонтально і вертикально по всьому коду. Шаблони синхронізації допомагають визначити розмір і структуру QR-коду.

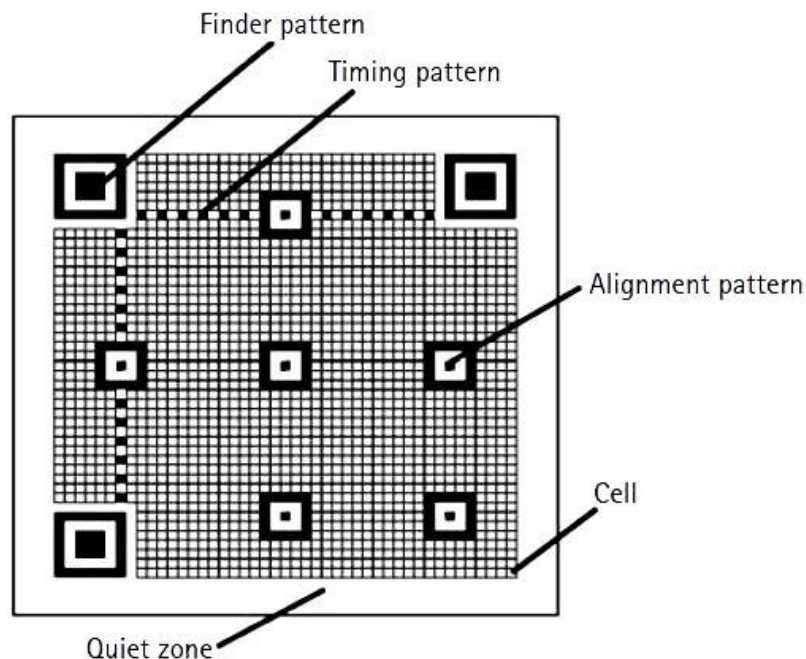


Рисунок 3.8 - Структура QR-коду

Для виконання поставленої задачі були обрані тільки шаблони пошуку. Алгоритм пошуку маркеру на зображенні включає в себе:

1. Перетворення отриманого зображення у градації сірого.
2. Всім пікселям, значення яких більше обраного значення, присвоюємо 255 - білий колір.
3. Всім пікселям, значення яких менше обраного значення присвоюємо 0 - чорний колір.

Після виконання попередніх кроків, отримуємо чорно-біле контрастне зображення, яке підвищує шанси коректного і правильного розпізнавання, навіть для неякісного вхідного зображення.

Наступним кроком знаходимо всі контури на чорно-білому зображенні. Серед знайдених контурів необхідно знайти три маркери, які будуть вказувати на грані необхідного контенту для подальшої роботи. Маркери - область коду утворена трьома вкладеними квадратами з відношенням сторін 7:5:3.

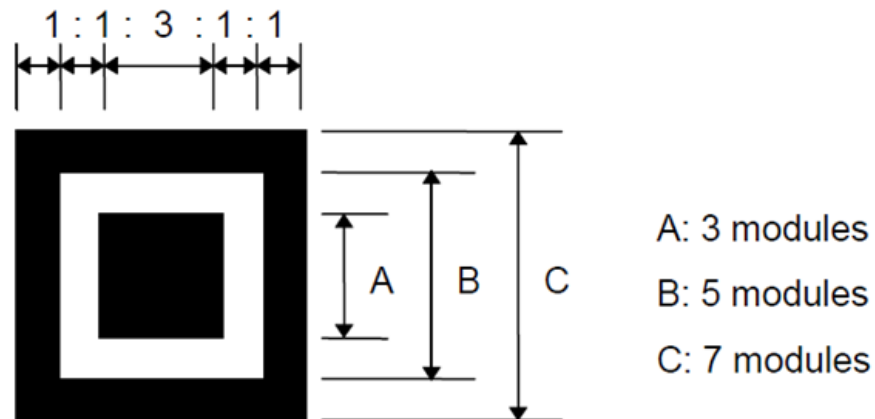


Рисунок 3.9 - Структура шаблону пошуку

3.5 Алгоритм визначення відповідей респондента

В першу чергу для визначення відповідей респондента зображення переводиться в бінарний вигляд за допомогою функції AdaptiveThreshold.[7] Цей метод автоматично визначає порогове значення для кожного пікселя на основі значень його сусідів. Використовується для того щоб картинка була більш чіткою для подальшої обробки.

Наступним кроком для картинки використовується розмиття по Гаусу для подальшого визначення контурів на зображенні.

Визначаються всі контури на зображенні за допомогою методу FindContours, при цьому використовується метод ApproxSimple. ApproxSimple - метод, який в основі використовує алгоритм Дугласа-Пьюкера, він спрощує контур, видаляючи при цьому непотрібні точки, зберігаючи основну форму.[1] На відміну від External методу він дає змогу знайти контури всередині таблиці.



	A	B	C
1	✓		
2			✓
3		✓	
4	✓		
5		✓	
6			✓

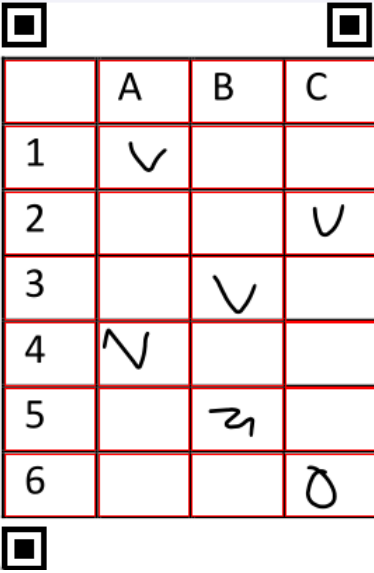
Рисунок 3.10 - Знаходження всіх контурів на зображенні

Наступним кроком спрощуються контури до більш простих багатокутних представлень. Для цього використовується алгоритм Дугласа-Пьюкера для зменшення точок кривої.

Для знаходження потрібних контурів комірок сітки відповідей фільтруються знайдені контури так що вони мають тільки 4 кути, що відповідає прямокутній формі комірки та фільтруються по розміру площі контуру, щоб відкинути занадто малі або, навпаки, занадто великі контури.

Існують випадки при яких контури дублюються з мінімальними змінами в координатах та позначка всередині комірки розпізнається як контур з 4 кутами. Для того щоб відфільтрувати подібні контури вони видаляються з результуючого списку контурів. Для цього знаходяться середні значення координат контурів та порівнюються з іншими. Для виключення випадків з позначками використовується функція PointPolygonTest, яка перевіряє кожен контур з

кожною з точок потенційного вкладеного контура.[1] Функція перевіряє чи є точка всередині контуру.



	A	B	C
1	✓		
2			✓
3		✓	
4	~		
5		~	
6			○

Рисунок 3.11 - Знаходження контурів комірок

Для визначення на яке саме питання респондент дав відповідь програма сортує знайдені контури комірок та розбиває їх на окремі рядки в правильному порядку. Для цього сортуються всі контури по координаті Y, розбиває їх на рядки на рядки, які відповідають кількості варіантів відповіді та заголовку рядка. В межах одного рядку контури відсортовуються по координаті X для розміщення їх в правильному порядку.

Маючи список з рядками та комірками всередині них визначається на яке саме питання була дана відповідь. Для цього з оригінальної картинки вирізається окремо комірка по координатам, які передані в функцію. За допомогою функції Threshold картинка комірки переводиться в бінарний вигляд для більш точного визначення наявності відповіді. Після чого за допомогою функції CountNonZero знаходиться кількість білих пікселів на картинці. Порівнявши кількість білих пікселів з розміром зображення визначається кількість чорних пікселів на картинці. Порівнявши значення кількості чорних пікселів для кожної комірки в межах одного рядку визначається, яку саме відповідь дав респондент.

	A	B	C
1	✓		
2			✓
3		✓	
4	✓		
5		✓	
6			✓

Рисунок 3.12 - Знаходження відповідей респондента

3.6 Тестування

Системне та приймальне тестування являють собою тестування завершеного та повністю інтегрованого програмного продукту. Ці тестування належить до тестувань методами чорної скриньки, де знання внутрішньої побудови коду або схеми роботи не є обов'язковою умовою. Зазвичай виконується командою тестування.

Перший етап тестування – тестування генерації сітки відповідей на основі анкети з питаннями. Вхідний документ зображений на Рисунок 3.13.

Результатом обробки документу є автоматично згенерована сітка відповідей в інтерфейсі програми відповідно до кількості питань та варіантів відповідей для заповнення балів за кожне з питань (Рисунок 3.14).

Після заповнення полів з балами та мінімальним значенням для проходження анкетування результати зберігаються в конфігураційний файл для подальшого використання під час розпізнавання результату.

1. Length of employment
 - A. More than 3 years
 - B. 1-3 years
 - C. Less than 1 year
2. Applicant's length of time at current address
 - A. More than 3 years
 - B. 1-3 years
 - C. Less than 1 year
3. Applicant's current residence.
 - A. Home owner
 - B. Renting a unit/home
 - C. Living with friends/relatives
4. Credit advantage report
 - A. Excellent
 - B. Average
 - C. Below average
5. Deposit account with bank
 - A. Yes
 - B. No
6. Active credit cards.
 - A. One card
 - B. Two cards
 - C. More than two cards

Рисунок 3.13 - Вхідний документ з анкетною

The screenshot shows a software window titled 'MainWindow' with two tabs: 'Генерація анкети' (Survey Generation) and 'Розпізнавання відповідей' (Answer Recognition). The 'Генерація анкети' tab is active, displaying a survey form. The form includes a text field for the user's name, currently showing 'C:\Users\Danil Beloko', and a table for recording answers. The table has columns for question numbers (1-6) and three options (A, B, C). Below the table, there is a section for 'Прохідне значення' (Threshold value) with a text input field containing '100'. At the bottom, there is a button labeled 'Зберегти значення' (Save values).

Анкета	A	B	C
1	10	20	30
2	10	20	30
3	10	20	30
4	10	20	30
5	10	20	30
6	10	20	30

Рисунок 3.14 - Згенерована сітка відповідей в інтерфейсі програми

Другий крок тестування — тестування зображень з різних ресурсів: документ формату PDF сформований програмно з документу, скан копія роздрукованого заповненого документу від руки та фотографія роздрукованого заповненого документу.

В випадку файлу формату PDF не виникає проблем при розпізнаванні позначок, сітка розташована рівно та маркери чіткі.

	A	B	C
1	✓		
2			U
3		✓	
4	✓		
5		2s	
6			0

1. Length of employment

A. More than 3 years

B. 1-3 years

C. Less than 1 year

2. Applicant's length of time at current address

A. More than 3 years

B. 1-3 years

C. Less than 1 year

3. Applicant's current residence.

A. Home owner

B. Renting a unit/home

C. Living with friends/relatives

4. Credit advantage report

A. Excellent

B. Average

C. Below average

5. Deposit account with bank

A. Yes

B. No

6. Active credit cards.

A. One card

B. Two cards

C. More than two cards

	A	B	C
1	✓		
2			U
3		✓	
4	✓		
5		2s	
6			0

Score = 120

1. Length of employment

A. More than 3 years

B. 1-3 years

C. Less than 1 year

2. Applicant's length of time at current address

A. More than 3 years

B. 1-3 years

C. Less than 1 year

3. Applicant's current residence.

A. Home owner

B. Renting a unit/home

C. Living with friends/relatives

4. Credit advantage report

A. Excellent

B. Average

C. Below average

5. Deposit account with bank

A. Yes

B. No

6. Active credit cards.

A. One card

B. Two cards

C. More than two cards

Рисунок 3.15 - Результат обробки файлу формату PDF

Розпізнавання скан-копії документу складніша через нечіткі маркери та не рівне розташування сітки на зображенні. За допомогою попередньої обробки зображення програмі вдається розпізнати відповіді правильно.

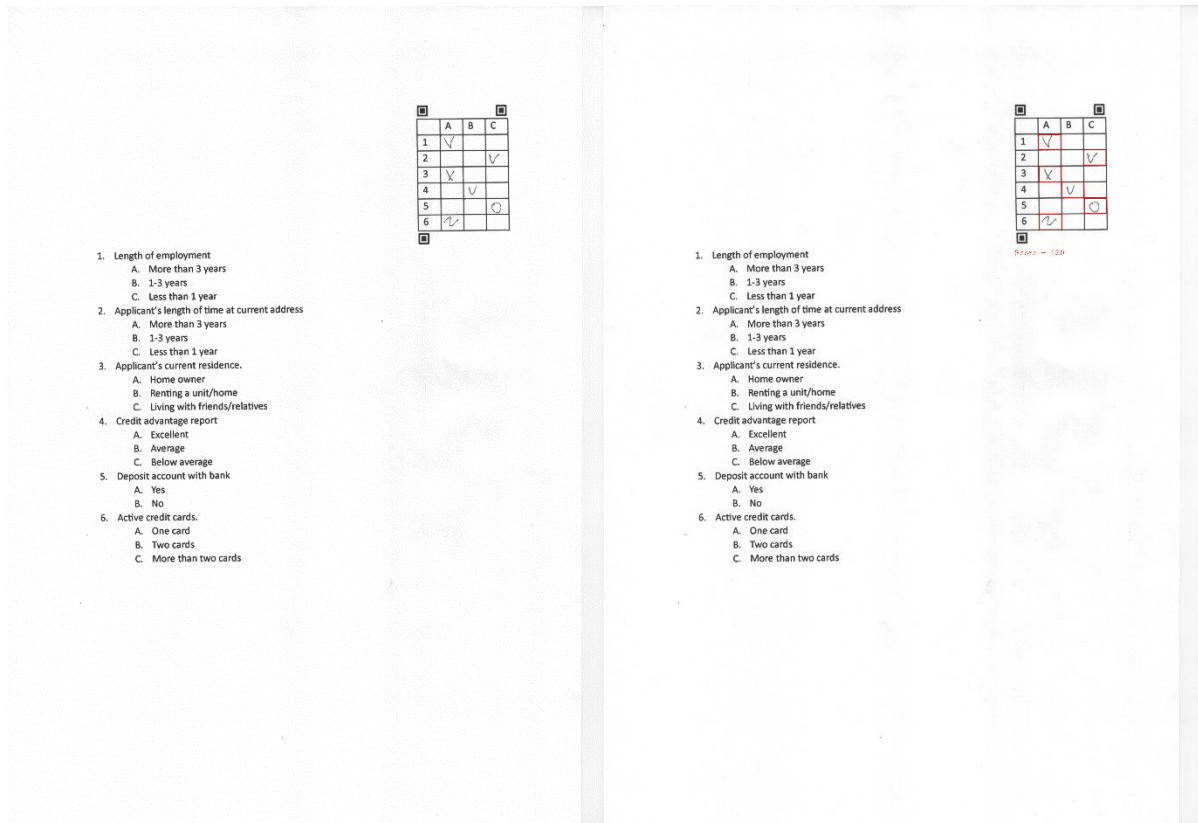


Рисунок 3.16 - Результат обробки скан-копії

Розпізнавання з фотографії було найбільш складним для тестувань. Окрім якості зображення, на яку впливає камера, також додатковими факторами виступають навколишнє освітлення і кут, під яким було зроблене фото. З проведених тестів було виявлено, що програма вдало розпізнає фотографії зроблені під прямим кутом відносно друкованого документу та на яких сітка відповідей не перекривається тінню, або іншими предметами.

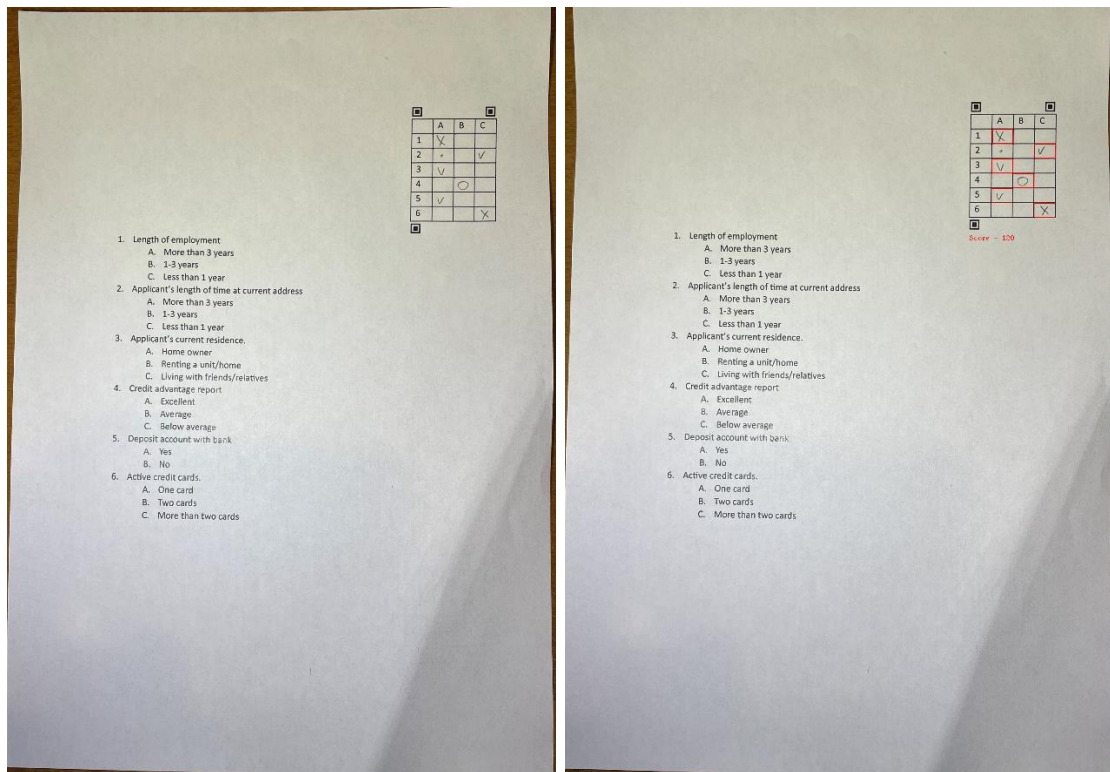


Рисунок 3.17 - Розпізнавання фотографії

Протестувавши роботу програми на трьох основних методах вводу зображення, були отримані задовільні результати, які повністю відповідали очікуванню і вказаним вимогам.

3.7 Використані технології

Для реалізації програмного забезпечення були обрані:

- Об'єктно орієнтована мова програмування C#;
- Система будування клієнтських форм на ОС Windows - Windows Presentation Foundation;
- Бібліотека алгоритмів комп'ютерного зору OpenCV;
- Бібліотека інструментів для роботи з документами OpenXML SDK.

Вибір C# як основного інструменту був обумовлений вимогами та спрощенням коду за рахунок перевантаження функцій та операторів, великою безпекою, можливістю використання об'єктно-орієнтованого підходу, керування ресурсами за допомогою RAII.

Вибір WPF (Windows Presentation Foundation) обумовлений тим, що платформа надає широкий і гнучкий набір графічних функцій, який має наведені нижче переваги:

- Незалежність графіки від дозволу та пристрою;
- Підвищення точності;
- Розширена підтримка графіки та анімації;
- Апаратне прискорення.

Вибір OpenCV як інструменту розпізнавання обумовлений універсальністю інструменту з широким спектром функцій і алгоритмів обробки зображень, який не обмежується завданнями розпізнавання символів або позначок. Основними перевагами було виявлено можливості сегментації зображення, низькорівнева обробка зображення (робота з зображенням на піксельному рівні), що допомагає в роботі зі складними, нестандартними форматами зображень або зображенням низької якості.

Вибір OpenXML SDK обумовлений необхідністю інструменту для роботи з файлами формату Office Open XML (OOXML). Порівняно з альтернативними інструментами, OpenXML SDK має наступні переваги:

- Нативність: інструмент розроблений Microsoft і забезпечує прямий доступ до основної файлової структури документів;
- Повний контроль: інструмент має повний доступ і контроль над вмістом, форматуванням і структурою документів Office;
- Ефективність: інструмент забезпечує високу швидкість створення та обробки документів, що робить його придатним для сценаріїв, де продуктивність має вирішальне значення, наприклад, великомасштабна обробка документів або створення документів на стороні сервера;
- Кросплатформеність: як і файловий формат, інструмент підтримується кількома платформами, включаючи Windows, macOS і різні дистрибутиви Linux;
- Інтегрованість з іншими інструментами.

3.8 Напрямки для вдосконалення

Програмне забезпечення має широкий потенціал подальшої розробки і вдосконалення в межах предметної області розпізнавання кредитоспроможності фізичних осіб і поза нею.

Для покращення автоматизації процесу визначення кредитоспроможності, можуть бути розроблені функції розпізнавання документів особи та зчитування значень важливих рядків, таких як номер паспорту або ПІН, для прискорення і полегшення процесу валідації документів зі сторони користувача.

Також, програмне забезпечення може бути інтегроване в багато інших предметних областей, які працюють на основі позначок і потребують інструменту їх автоматичного розпізнавання. Наприклад:

- **Виборчі голосування** - оптичне розпізнавання позначок може використовуватися для збору та обробки голосів під час виборчих процесів. Виборчі бюлетені зазвичай мають спеціальні поля з позначками.
- **Тестування** - оптичні форми з позначками використовуються для проведення масових тестів, таких як сертифікаційні іспити або стандартизовані тести (ЗНО, НМТ, тощо). Відповіді студентів реєструються за допомогою позначок в спеціальних бланках, а потім обробляються системою оптичного розпізнавання.
- **Анкети та опитування** - інструмент може бути використаний в будь якій іншій сфері для обробки анкет і опитувань. Респонденти заповнюють анкети, ставлячи позначки або штрихові коди відповідно до їх відповідей. Оптична система розпізнавання може автоматично зчитати ці відповіді та обробити їх, що дозволяє швидко та ефективно обробляти великі обсяги даних, отриманих з оптичних форм з позначками.

Висновки до розділу 3

У даному розділі було повністю описано процес проектування та реалізації програми, опис алгоритмів та опис взаємодії копристувача з застосунком. Описані технології, які використовувались та обґрунтування вибору. Були проведені дослідження ефективності запропонованого алгоритму та проведено тестування на реальних випадках застосування. Також було розглянуто напрямки для вдосконалення застосунку.

ВИСНОВКИ

В ході виконання кваліфікаційної роботи магістра було досліджено процес визначення кредитоспроможності фізичних осіб, процес оптичного розпізнавання позначок за допомогою програмного застосунку, загальні методи та засоби розпізнавань. Під час проведення теоретичних досліджень та досліджень існуючого інструментарію, були обрані методи оцінки кредитоспроможності анкетуванням та алгоритми розпізнавання позначок результатів.

Після розробки програмного забезпечення продовжувалось проведення теоретичних та практичних досліджень разом з тестуванням застосунку. В результаті яких було виявлено і вдосконалено обрані методи та алгоритми для роботи в різних умовах і з різним зображенням.

Розроблений програмний застосунок може бути використаний як самостійний інструмент розпізнавання позначок на зображенні та має високі перспективи подальшої розробки, додавання нового функціоналу або інтегрування в інші предметні області для розвитку дослідження та отримання нових результатів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. OpenCV - [Електронний ресурс]. - Режим доступу: <https://docs.opencv.org/>.
2. Tesseract OCR - [Електронний ресурс]. - Режим доступу: <https://tesseract-ocr.github.io/>
3. OpenXML - [Електронний ресурс]. - Режим доступу: <https://learn.microsoft.com/en-us/office/open-xml/open-xml-sdk>
4. Li, C., & Doermann, D. "Document image analysis: An overview." в книзі "Handbook of document image processing and recognition" - 2009.
5. Suen, C. Y. "Methods and algorithms in document image analysis." в книзі "Handbook of character recognition and document image analysis" - 1996.
6. Gonzalez, R. C., & Woods, R. E. Digital image processing - 2018.
7. Bradski, G., & Kaehler, A. "Learning OpenCV: Computer vision with the OpenCV library" - 2008.
8. van Hees, W., & Schomaker, L. "Optical mark recognition: A state-of-the-art survey." - 2018.
9. Zheng, L., & Doermann, D. "Document structure analysis algorithms: A literature survey." в книзі "Document Recognition and Retrieval XI" - 2004.
10. Farooq, F., & Trivedi, M. M. OpenSet OCR: A comprehensive benchmark - 2016.
11. Smith, R. An overview of the Tesseract OCR engine - 2019.
12. Lamiroy, B., & Thonnat, M. Template matching for document image analysis - 2004.
13. Singh, A., & Kaur, M. Hybrid text recognition system for identification of textual information from natural scene images - 2018.
14. Li, D., Guo, G., Li, L., & Zhang, D. Feature representation and metric learning for image retrieval - 2019.
15. Mishra, A., & Chauhan, P. An overview of optical character recognition (OCR) techniques - 2018.

16. L. Huo, J. Zhu, P. Kumar Singh, A. Pljonkin – Research on QR image code recognition system based on artificial intelligence algorithm – 2021.