

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСОБІВ
РОЗПІЗНАВАННЯ ШТРИХОВИХ КОДІВ
RESEARCH AND SOFTWARE IMPLEMENTATION OF
BARCODE RECOGNITION TOOLS

Виконав: студент 2 курсу, групи КНТ-112м

Спеціальності 121 Інженерія програмного

(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

КОВАЛЬ М.О.

(ПРІЗВИЩЕ та ініціали)

Керівник ОЛІЙНИК А.О.

(ПРІЗВИЩЕ та ініціали)

Рецензент КОЦУР М.І.

(ПРІЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти магістр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2023 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

КОВАЛЯ Максима Олександровича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація засобів розпізнавання штрихових кодів. Research and Software Implementation of Barcode Recognition Tools

керівник проєкту (роботи) д.т.н., професор ОЛІЙНИК Андрій Олександрович,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “ 14 ” листопада 2023 року № 437

2. Строк подання студентом проєкту (роботи) 01 грудня 2023 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ОЛІЙНИК А.О., доцент		
Нормоконтроль	КАМІНСЬКА Ж.К., асистент		

7. Дата видачі завдання “ 16 ” жовтня 2023 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій	2 тиждень	Розділ 2
	розробки		
4	Розробка структури програми.	3 тиждень	Розділ 3
5	Розробка програми.	4-7 тижні	Розділи 3,4
6	Тестування та експериментальне дослідження	8 тиждень	Розділ 4
	програмного забезпечення.		
7	Оформлення пояснювальної записки та документів	9-10 тижні	Додатки
	до неї.		
8	Нормоконтроль та рецензування.	11 тиждень	
9	Захист роботи.	12 тиждень	

Студент(ка)

(підпис) Максим КОВАЛЬ
(Імя ПРИЗВИЩЕ)

Керівник проєкту (роботи)

(підпис) Андрій ОЛІЙНИК
(Імя ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
90 с., 35 рис., 3 табл., 3 дод., 23 джерела.

C#, MICROSOFT VISUAL STUDIO, МУЛЬТИАГЕНТНИЙ МЕТОД,
ОПТИМІЗАЦІЯ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, РОЗПІЗНАВАННЯ
ОБРАЗІВ, ШТРИХОВИЙ КОД.

Об'єкт дослідження – процес розробки програмного забезпечення для
розпізнавання образів.

Предмет дослідження – методи та програмні засоби для розв'язання
завдань розпізнавання штрихових кодів.

Метою роботи є дослідження та розробка програмного забезпечення
для розпізнавання штрихових кодів.

Матеріали, методи та технічні засоби: мова програмування C#,
середовище розробки Microsoft Visual Studio.

Результати. Розроблено програмне забезпечення для розв'язання
завдань розпізнавання штрихових кодів за допомогою мови програмування C#
та середовища розробки Microsoft Visual Studio.

Практична цінність роботи полягає у розробці програмного
забезпечення, що реалізує методи розпізнавання образів та дозволяє
розпізнавати штрихові коди.

Висновки. Виконано проєктування програмного забезпечення для
розпізнавання штрихових кодів. Розроблено програмне забезпечення для
розпізнавання штрихових кодів. Здійснено тестування розробленого
програмного забезпечення для розпізнавання штрихових кодів.

Галузь використання – програмні засоби для розпізнавання штрихових
кодів.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 90 pages, 35 figures, 3 tables, 3 appendixes, 23 sources.

C#, MICROSOFT VISUAL STUDIO, MULTI-AGENT METHOD, OPTIMIZATION, SOFTWARE, PATTERN RECOGNITION, BAR CODE.

The object of research is the process of developing software for pattern recognition.

The subject of the research is software for solving barcode recognition tasks.

The purpose of this work is to develop the software for barcode recognition.

Materials, methods and technical tools: C# programming language, Microsoft Visual Studio development environment.

Results. The software tools for solving barcode recognition tasks using C# programming language and Microsoft Visual Studio development environment have been developed.

The practical value of the work lies in the development of software that implements pattern recognition methods and allows you to solve problems of recognition of barcodes.

Conclusions. The software for barcode recognition has been designed. The software for barcode recognition has been developed. The developed software for barcode recognition has been tested.

Scope of use – software tools for barcode recognition.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	8
Вступ	9
1 Аналіз предметної області	11
1.1 Задача розпізнавання штрихових кодів	11
1.2 Аналіз програмного забезпечення для розпізнавання штрихових кодів	24
1.3 Висновки за розділом 1	32
2 Матеріали і методи	34
2.1 Вибір мови програмування	34
2.2 Вибір середовища розробки для розпізнавання штрихових кодів	36
2.3 Методи та алгоритми аналізу даних для розпізнавання штрихових кодів	38
2.4 Висновки за розділом 2	42
3 Опис програми	43
3.1 Структура програмного забезпечення для розпізнавання штрихових кодів	43
3.2 Функціонування програмного забезпечення для розпізнавання штрихових кодів	44
3.3 Проектування інтерфейсу програмного забезпечення для розпізнавання штрихових кодів	47
3.4 Висновки за розділом 3	49
4 Експлуатація, тестування та експериментальне дослідження програми	50
4.1 Призначення й умови застосування програми	50
4.2 Характеристики програми для розпізнавання штрихових кодів	51
4.3 Інструкція по експлуатації програми	51
4.3.1 Звернення до програми	51
4.3.2 Вхідні й вихідні дані	51
4.3.3 Повідомлення	51

4.4 Виконання програмного забезпечення для розпізнавання штрихових кодів	52
4.5 Тестування програмного забезпечення для розпізнавання штрихових кодів	54
4.6 Висновки за розділом 4	58
Висновки.....	59
Перелік джерел посилання	62
Додаток А Технічне завдання.....	64
Додаток Б Фрагмент тексту програми	68
Додаток В Слайди презентації	83

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

CV – Computer Vision;

EAN – European Article Number;

QR – Quick Response;

UPC – Universal Product Code;

ПЗ – програмне забезпечення;

ШІ – штучний інтелект.

ВСТУП

Програмні засоби для розпізнавання штрихових кодів використовуються у багатьох сферах, таких як логістика, роздрібна торгівля, управління запасами, лікарська індустрія, продовольча промисловість та багато інших [1]-[4]. За допомогою програмних засобів для розпізнавання штрихових кодів можна підвищити швидкість та точність сканування кодів, що дозволяє покращити ефективність різних операцій, таких як ведення обліку товарів, відстеження поставок, оформлення замовлень тощо [3]-[6].

Ці програмні рішення спираються на засоби штучного інтелекту, машинного навчання та комп'ютерного зору, що забезпечують високу точність розпізнавання та прийнятну швидкість цього процесу [1]-[7].

Останнім часом розпізнавання штрихових кодів стає можливим шляхом використання різних пристроїв, зокрема, мобільних, що значно полегшує процес розпізнавання для споживачів та бізнесу, а також дозволяє спростити процеси покупки, отримання інформації про товари тощо [1]-[6].

Однак відомі програмні засоби для розпізнавання штрихових кодів можуть обмеженими у підтримці різних форматів штрихових кодів, що може викликати проблеми при розпізнаванні певних типів кодів. При використанні інших програмних засобів, особливо при обробці великих обсягів даних або на повільних пристроях, час, необхідний для розпізнавання, може бути значним. Крім того, багато програмних засобів для розпізнавання штрихових кодів вимагають чіткості та яскравості штрихового коду, внаслідок чого низька якість друку, пошкодження або розмивання штрихового коду можуть призвести до неправильного або некоректного розпізнавання. Також освітлення та кути зору можуть впливати на точність розпізнавання, відповідно, недостатнє або неправильне освітлення може призвести до помилок у розпізнаванні. Деякі програмні засоби для розпізнавання штрихових кодів можуть вимагати дорогого обладнання або платних

програмних рішень, що в деяких випадках обмежує їх застосування на практиці, особливо в невеликих організаціях [4]-[7].

Тому актуальною є розробка програмного забезпечення для розпізнавання штрихових кодів.

У дипломній кваліфікаційній роботі магістра розв'язується актуальне завдання розробки програмного забезпечення для розпізнавання штрихових кодів.

Для досягнення поставленої мети у кваліфікаційній роботі магістра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів для розпізнавання штрихових кодів;
- розробити метод розпізнавання штрихових кодів на основі інтелектуальних обчислень;
- здійснити проектування програмного забезпечення для розпізнавання штрихових кодів;
- створити програмне забезпечення для розпізнавання штрихових кодів;
- виконати тестування розробленого програмного забезпечення для розпізнавання штрихових кодів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Задача розпізнавання штрихових кодів

Штрихові коди є числовими та лінійними символами, які знаходяться на товарах та допомагають в їх ідентифікації. Проте їхнє використання виходить далеко за межі простої ідентифікації. Програмні системи оброблення штрих-кодів допомагають підприємствам в відстеженні товарів, управлінні цінами та запасами, що сприяє більш ефективному керуванню комп'ютерними програмами та підвищенню робочої продуктивності. Ці символи представляють інформацію у вигляді ліній та звичайних шаблонів, які легко зчитуються оптичними сканерами або читачами штрих-кодів, інформація потрапляє безпосередньо до комп'ютерних систем. Використання штрих-кодів допомагає уникнути помилок, які можуть виникнути при ручному введенні даних, зменшуючи час, що витрачається на цей процес. Штрих-коди, починаючи з простих одновимірних зображень, тепер існують у різних формах і розмірах, більшість з яких можна зчитувати за допомогою мобільних пристроїв та інших гаджетів [1]-[8].

Існують два основних види штрих-кодів: лінійні, відомі також як 1D, та двовимірні – 2D. Універсальний код товару є прикладом 1D штрих-коду, який має дві складові: сам штрих-код і 12-значний номер. Перші шість цифр штрих-коду визначають виробника, наступні п'ять – конкретний продукт, а остання цифра – контрольна, яка гарантує правильність сканування [1]-[3].

У лінійних штрих-кодів може бути кілька типів тексту. У свою чергу, двовимірні штрих-коди більш складні, мають можливість зберігати різноманітну інформацію: ціни, кількість, вебадреси чи навіть зображення. Лінійні сканери не можуть читати 2D коди, для цього потрібен сканер зображень. Однак, сучасні смартфони, такі як iPhone, Android та інші, обладнані камерою, можуть зчитувати 2D штрих-коди. Створення двовимірних штрих-кодів розширило їхні можливості: тепер вони використовуються для обміну інформацією зі споживачами через QR-коди, що

можуть містити до 7000 чисел або 4000 символів. Ці коди застосовуються в різноманітних сферах, від веб-сайтів до медичних систем оцінки та обміну даними в організаціях охорони здоров'я. Вони також використовуються в програмах, як MS Office, MS SQL Server та інших базах даних. Безкоштовний генератор QR-кодів допоможе створити QR-код для вашого бізнесу [3]-[6].

Штрихові коди представляють собою знаки, що можна сканувати за допомогою електронних пристроїв, використовуючи технології на базі лазерів або зображень [1]-[8].

Штрихові коди служать для кодування важливих даних, таких як ідентифікаційні номери (продукту, відправлення, місцезнаходження і т. ін.) і ключові характеристики (серійний номер, номер партії/лоту, дата і т. ін.), використовуючи стандартний синтаксис, що включає звичайні рядки цифрових посилань [1]-[8].

Штрихові коди грають важливу роль у ланцюгах поставок, дозволяючи автоматично ідентифікувати та відстежувати продукцію від роздрібних торговців до виробників, транспортників і лікарень під час переміщення по ланцюгу поставок [1]-[8].

Існують різні типи штрих-кодів, які відповідають різним потребам бізнесу. Лінійні штрих-коди часто використовуються на товарах, використовують різні ширини ліній і інтервалів для кодування інформації, для ідентифікації конкретного продукту [1]-[8].

У двовимірних штрих-кодах для кодування використовуються різноманітні форми, такі як квадрати, шестикутники, точки і інші. Оскільки в такому випадку дані кодуються у двовимірному форматі, ці штрих-коди можуть містити значно більше інформації, ніж лінійні, навіть за фізично менших розмірів [1]-[8].

Штрихові коди складаються з символів, які сканери можуть розпізнати і перетворити на корисну інформацію, яка зазвичай містить дані про походження об'єкта, його тип, вид і місце розташування. Після сканування штрих-коду, збережені дані автоматично вносяться в систему, часто у базу

даних. Цей інструмент відкриває безліч переваг для бізнесу. Він забезпечив основу для глобальних систем постачання і дистрибуції, допомігши компаніям, наприклад, WalMart, переконатися в належному обладнанні продукції та встановленні правильних цін по всьому світу. Штрих-коди стали необхідним інструментом для малих та середніх підприємств, а також для лікарень і громадських організацій, допомагаючи відстежувати активи і підвищувати ефективність [1]-[8].

Використання штрихових кодів дозволяє розв'язувати різні завдання. Система відстеження запасів програмного забезпечення складається з програм та пристроїв для сканування штрих-кодів або ноутбуків. Товари, матеріали або сировина у вашому запасі мають спеціальні мітки із штрих-кодами. При вибутті товару зі складу ви просто скануєте його штрих-код, що автоматично вводить дані у ваше програмне забезпечення для відстеження запасів, уникаючи ручного введення в базу даних. Це допомагає оптимізувати управління запасами [1]-[8].

Також ця система може бути використана для керування активами. Усі компанії мають свої ІТ-активи та основні засоби. Мітки з штрих-кодами прикріплюються до кожного окремого активу і можуть бути скановані для збереження або отримання даних про них у програмному забезпеченні відстеження активів. Це ефективний спосіб підвищити звітність та спростити перевірки [1]-[8].

Штрих-коди також можуть застосовуватися у взаємному листуванні. Для цього використовується реєстр штрих-кодів, який відповідає серійному номеру товару, що дозволяє миттєво визначати, які серійні номери були зареєстровані. Це також звільняє клієнтів від необхідності пошуку та переписування тривалих серійних номерів [1]-[8]. Також можна включити штрих-коди до електронного листа. У випадку, якщо компанія організовує захід, до запрошень можна додати штрих-коди, щоб відслідковувати відповіді без необхідності розшифровування чийогось почерку [1]-[8].

Додавши штрих-коди до рахунків-фактур, можна включити номер клієнта або номер рахунку-фактури, щоб легко здійснювати пошук клієнтських облікових записів або рахунків при поверненні з оплатою. Відповідне програмне забезпечення допомагає уникати можливих непорозумінь, таких як неправильне використання облікових записів клієнтів або помилкова оплата рахунків [1]-[8].

Штрих-коди легко можна впровадити в бізнес, використовуючи відповідне програмне забезпечення. Запитання генерації штрих-кодів можна вирішити дистанційно, використовуючи доступні у мережі інструменти, не витрачаючи значних інвестицій [1]-[8].

Зазвичай штрих-коди являють собою чорні вертикальні лінії різної товщини та інтервалів, що формують квадрат або прямокутник, кодуючи інформацію про конкретний продукт. Сканери можуть розпізнати ці коди та отримувати інформацію про продукт через правильну комбінацію цифр, пробілів і ліній [1]-[8].

Штрих-коди зараз використовуються не лише на товарах у магазинах, а й на водійських посвідченнях, автомобілях в оренду, багажі на реєстрацію та медичних браслетах. Вони служать для ідентифікації продукту або особи та дозволяють кодувати важливі характеристики [1]-[8].

Штрих-коди є способом закодувати дані про товар за допомогою ліній і символів, що спрощує процес отримання товару в магазині або відстеження його на складі [1]-[8].

Поміж швидкістю і простотою, ключові комерційні переваги штрих-кодів полягають у точності, контролі над запасами та економії коштів [1]-[8].

Штрих-коди поділяються на два основні типи: лінійні коди, які включають поширені формати, і матричні коди, наприклад Quick Response коди (QR-коди) [1]-[8].

Для використання штрих-кодів необхідне мінімальне обладнання: принтер, сканер і базове програмне забезпечення для управління запасами компанії [1]-[8]. Штрих-коди економлять час і гроші, тому що їх можна

зчитувати сканером, як на портативному пристрої, так і на касовому апараті, без необхідності працівнику вводити інформацію про продукт вручну [1]-[8].

Існують два основних типи штрих-кодів: одновимірний, який можна побачити на упаковках продуктів або пляшках шампуню, та двовимірний, наприклад, QR-коди в рекламі, які ведуть користувача на вебсайт компанії. Більшість сканерів здатні зчитувати лише одновимірні або лінійні штрих-коди, які є найпоширенішим форматом [1]-[8]. Найбільш популярними типами лінійних штрих-кодів є Universal Product Code (UPC, універсальний товарний код) у США та European Article Number (EAN, європейський номер товару) у Європі [1]-[8].

При розпізнаванні штрихових кодів важливо розуміти, що ширина чорних смуг у штрих-коді зазвичай представляє числа 0 або 1, а їхнє розташування визначає числа від 0 до 9. Комп'ютер, який приєднаний до сканера, зберігає повну інформацію про продукт, яка пов'язана з унікальною послідовністю смуг і пробілів. Програмне забезпечення може обробляти, множити або ділити ці числа для отримання правильного значення, яке відображається на екрані [1]-[8].

На складі штрих-код може кодувати різні параметри продукту, такі як розмір, колір, а також його місцезнаходження. Це дозволяє компанії детально відслідковувати наявність товарів, замовлення та запаси, або швидко здійснити їхнє завершення. У роздрібній торгівлі ця інформація може містити назву продукту та його ціну, яку співробітник повинен підтвердити перед клієнтом. Організації можуть використовувати штрих-коди для відстеження продукції на всіх етапах її життєвого циклу, від виробництва до розповсюдження, купівлі, обслуговування та ремонту [1]-[8].

Компоненти штрих-коду потрібно створювати точно й однаково, щоб сканери могли їх зчитувати та передавати закодовану інформацію на комп'ютер. Ці компоненти можуть також вказувати на країну походження, категорію товару та виробника [1]-[8].

Проаналізуємо складові штрих-коду UPC [1]-[8]. Безшумна зона цього коду є тонкою білою областю, яка оточує штрих-код, необхідною для зчитування етикетки сканером [1]-[8].

Номер системи нумерації: перше число вказує на категорію продукту за кодом UPC. Наприклад, роздрібні товари зазвичай починаються з 0 або 1, медикаменти з 3, а купони з 5 [1]-[8].

Перші символи після цифри вказують на конкретного виробника. Глобальна організація стандартизації, яка контролює UPC, надає унікальний код кожному виробнику. Наступна послідовність символів ідентифікує конкретний товар, створений виробником. Цифра перевірки: цей номер підтверджує точність інформації, пов'язаної з штрих-кодом, і виявляє можливі помилки [1]-[8].

Проаналізуємо переваги використання у програмному забезпечення штрихових кодів для бізнесу [1]-[8]. Штрих-коди стали популярними завдяки їх чіткості та миттєвості повернення вкладень. Штрих-коди дозволяють уникнути ручного введення даних про продукт, знижуючи тим самим ймовірність помилок. Дані в режимі реального часу: кожне сканування штрих-коду автоматично оновлює інвентаризацію та номери продажів у системі управління ресурсами підприємства або системі управління компанією. Це надає компанії постійний доступ до свіжих даних для швидких розрахунків, таких як оборотність запасів, потенційна вартість запасів чи тижневі продажі за одиниці [1]-[8].

Штрих-коди та сканери є зрозумілими інтуїтивно, що спрощує адаптацію нових співробітників. Вони також значно зменшують потребу в запам'ятовуванні та великих знаннях. Наприклад, у продуктовому магазині працівнику не потрібно запам'ятовувати коди популярних продуктів для їх обробки [1]-[8].

Використання штрих-кодів сприяє оптимізації контролю за запасами та зменшенню рівня використання товарів. Це дозволяє персоналу завжди мати актуальну інформацію щодо стану запасів та попиту, що уможливорює більш

точне прийняття рішень про закупівлю та розпродажі. Такий підхід зменшує витрати на запаси і уникнення застарілих товарів, що в свою чергу сприяє покращенню довгострокової прибутковості [1]-[8].

Витрати на їх початкове створення програмної системи підтримки процесу розпізнавання штрихових кодів не такі значні у порівнянні з перевагами. Підприємства можуть недорого створювати декілька штрих-кодів для внутрішнього використання, а вартість супутніх технологій лишається низькою, оскільки попит на них зростає [1]-[8].

Різноманітність штрих-кодів стимулює компанії розробляти різні види програмного забезпечення для оброблення цих кодів. Проаналізуємо деякі з найпоширеніших типів штрихових кодів [1]-[8].

Штриховий код UPC (Universal Product Code) має або 12 або 7 цифр, є одним з найпоширеніших. Він використовується для різноманітних товарів у США, від продуктів харчування до товарів для дому і канцелярських товарів. Штриховий код EAN (European Article Number) має 13 або 8 цифр і, в основному, використовується в Європі для ідентифікації споживчих товарів, схожий на UPC, але більш поширений в Європі [1]-[8].

Штриховий код Code 128 може включати всі 128 символів ASCII, часто використовується у секторі продажу товарів, дистрибуції та транспорті. Його можна побачити в різних закупівлях та транспортуванні товарів. Штриховий код Codebar може містити цифри та літери A-D і все ще використовується деякими бібліотеками, банками крові та службами доставки [1]-[8].

QR-коди набувають все більшої популярності у продажах та маркетингу. Більшість осіб вже користувалися QR-кодами або хоча б знають про них. Нові моделі телефонів використовують вбудовану камеру для сканування цих кодів [1]-[8].

Використання штрих-кодів пов'язано з використанням різних типів програмного забезпечення, що може значно підвищити ефективність та точність в різних щоденних операціях. Проаналізуємо програмне забезпечення, що пов'язано з використанням штрихових кодів [1]-[8].

Програмне забезпечення моніторингу запасів дозволяє підприємствам мати інформацію про актуальний рівень запасів, а також про статус кожного товару. Штрих-коди допомагають у вирішенні цієї проблеми управління запасами: вони періодично скануються персоналом складу при прибутті нових товарів та відвантаженні замовлень. Сканери, що встановлені на автоматизованих лініях, можуть автоматично сканувати товари під час їх руху по конвеєру. Кожному товарному коду (SKU) необхідний власний штрих-код, і коди повинні бути пов'язані з базою даних, що містить повну інформацію про товар. Після впровадження системи управління запасами запаси завжди залишаються надійними та актуальними [1]-[8].

Програмне забезпечення відстеження активів також використовує штрихові коди. Таке програмне забезпечення дозволяє відстежувати авто і комп'ютери, а також інше обладнання та активи, які компанії використовують у своїй повсякденній діяльності. У поєднанні з програмним забезпеченням для відстеження активів, штрих-коди допомагають підприємствам визначати розташування та стан цих активів, фіксуючи всі відповідні записи, пов'язані з їх обслуговуванням чи ремонтом. Це критично, оскільки, на відміну від інвентаризації, організації зберігають основні засоби на довгі періоди та часто використовують їх повторно. Для рухомих об'єктів, таких як комп'ютери чи транспортні засоби, штрих-коди можуть вказати, хто востаннє використовував предмет і коли. Таке програмне забезпечення також сприяє керівникам підприємств зрозуміти використання та стан різних активів при плануванні майбутніх інвестицій [1]-[8].

Багато організацій використовують програмне забезпечення, яке додає штрих-коди до рахунків-фактур, щоб спростити відстеження кредиторських та дебіторських рахунків. Компанії можуть встановлювати зв'язок з конкретними клієнтами, додаючи штрих-коди до рахунків-фактур. Під час оплати клієнтом, працівник може сканувати штрих-код, щоб переконатися, що він зараховує кошти на правильний рахунок, і такий же процес може використовуватися при виставленні рахунків. Штрих-коди на рахунках можуть також спростити

процес доставки. Працівник складу може роздрукувати рахунок-фактуру для замовлення та сканувати код, щоб визначити, які товари та звідки потрібно забрати, знижуючи ймовірність помилок. Після того, як працівник збирає всі продукти для замовлення, штрих-код також може бути сканований, щоб переконатися, що всі правильні товари в упаковці [1]-[8].

Програмне забезпечення обробки штрих-кодів також може використовуватися для моніторингу усіх відправлень через пошту та кур'єрські послуги. Вони можуть сканувати листи та пакети перед їх відправленням кур'єрам, а також пов'язувати відомості про відстеження з конкретним замовленням, яке потім надсилають клієнтам, щоб вони могли перевірити статус своїх покупок. Після повернення товару продавцю, він може сканувати штрих-код для швидкого визначення, до якого клієнта слід звернутися для вирішення проблеми [1]-[8].

Створення штрих-кодів може бути здійснене компаніями за допомогою кількох методів. Ті, кому необхідним є обмежений обсяг штрих-кодів, можуть скористатися прямою поштовою розсилкою або використати безкоштовні інструменти для генерації штрихових кодів та їхнього друку на звичайному принтері. Проте цей підхід може стати неефективним з плином часу, коли зростають ваші потреби [1]-[8].

Більш оптимальним варіантом є використання програмного забезпечення для автоматизованого обліку та створення штрихових кодів. Навіть базові програмні системи для обліку товарів часто мають функцію створення штрихових кодів, що дозволяє легко пов'язати кожен штрих-код з конкретними продуктами у вашій базі даних. Для більшості компаній було б корисно інтегрувати цю інформаційну технологію з принтером штрих-кодів, який призначений спеціально для друку етикеток і може коштувати відносно невелику суму [1]-[8].

При використанні програмного забезпечення обробки штрих-кодів для зовнішніх цілей, таких як відстеження продажів через сторонні роздрібні мережі, необхідно зареєструватися у спеціальних реєстрах та отримати

унікальний ідентифікатор, який буде включений у всі штрих-коди. Ця підписка також дає доступ до центру даних, де є можливість створювати та управляти штрих-кодами, а також експортувати їхні файли на принтер [1]-[8].

Програмні системи обробки штрих-кодів забезпечують простий, ефективний та надійний спосіб відстеження запасів, що призводить до значної економії як доходів, так і витрат для багатьох підприємств [1]-[8].

Для нових компаній або тих, хто ще не впровадив штрих-коди, важливо вибрати програмне забезпечення та тип штрих-коду, що найкраще підходять для їхніх потреб, і переконатися, що вони застосовуються і скануються послідовно. Штрих-коди є ключовим елементом для надання підприємствам необхідної видимості для контролю витрат і забезпечення відмінного взаємодії з клієнтами [1]-[8].

Використання програмних засобів оброблення штрихових кодів вимагає декількох технічних рішень, які можуть допомогти підприємствам почати їх використання [1]-[8]. Проаналізуємо складові таких програмно-апаратних засобів [1]-[8]. Сканери штрихових кодів надають компаніям можливість скористатися перевагами штрих-кодів. Існує різні типи сканерів: лазерні, що є найпопулярнішими, вони доступні та здатні сканувати коди на великій відстані; сканери, які використовують спеціальні лампи для зчитування кодів, є точнішими, але мають менший радіус дії. Для двовимірних штрих-кодів потрібний сканер зображень, який може зчитувати ці коди за допомогою камери [1]-[8].

Централізована база даних підвищує ефективність використання програмних засобів оброблення штрих-кодів, потребує зв'язку цих унікальних ідентифікаторів з продуктами. Тому необхідна централізована база даних, яка встановлює зв'язок між кожним штрих-кодом та конкретним продуктом, забезпечуючи правильну інформацію після сканування коду. Ці дані зазвичай зберігаються у програмних додатках, які використовуються разом з системами та пристроями зберігання [1]-[8].

Програмне забезпечення для управління запасами є важливим для максимального використання можливостей, які пропонують штрих-коди. Ця система є центральним місцем зберігання всієї інформації про продукти, включаючи їх штрих-коди, і може діяти як база даних, що автоматично надсилає інформацію на комп'ютер під час сканування штрих-коду [1]-[8].

Використання штрих-кодів у поєднанні з програмними системами управління запасами дозволяє компаніям завжди мати точну та повну інформацію про поточні запаси та їх рух. При кожному переміщенні товару сканування штрих-коду дозволяє автоматично оновлювати дані в програмній системі управління запасами. Це уникає несподіваного вичерпання запасів у момент очікування нового замовлення, що може спричинити затримки у продажах. Крім того, система може відстежувати місцезнаходження кожного конкретного товарного запасу, що прискорює обробку та зменшує витрати на оплату праці [1]-[8].

Розпізнавання штрихових кодів відноситься до завдання розпізнавання образів [5]-[10]. Розпізнавання штрих-коду за допомогою програмного забезпечення передбачає сканування його в горизонтальному напрямку для отримання послідовності двійкових кодів, набір стрічок різної ширини та кольору, що містить інформацію про сам штрих-код. Декодування змісту штрих-коду можливе за допомогою різних методів розпізнавання образів та відповідного програмного забезпечення [5]-[10].

У наш час використовуються різні бібліотеки розпізнавання штрих-кодів, що являють собою набір програм або об'єктів, які дозволяють створювати зображення штрих-кодів та розпізнавати машинно-зчитані тексти/дані зі сканованих або знятих зображень із вбудованими штрих-кодами. Ці програмні бібліотеки можуть підтримувати два режими: створення та розпізнавання штрих-кодів [8]-[10].

Сучасні технології та програмні засоби оброблення штрих-кодів дозволяє додавати машинне зчитування додаткових даних до об'єктів реального світу за ціною менше одного цента. Це дозволяє використовувати

будь-який пристрій з камерою для ідентифікації додаткових даних про об'єкти. Поєднання технології штрих-кодів та програмних засобів оброблення штрихових кодів відкриває можливості для програм автоматизованої обробки документів, систем відстеження, а також розширеної реальності [8]-[10].

Використання технологій глибокого навчання для розпізнавання та аналізу штрих-кодів показало вражаючі результати і привернуло велику увагу у різних галузях. Порівняно з іншими методами, підходи, що базуються на глибокому навчанні, можуть значно покращити швидкість та точність виявлення та декодування штрих-кодів. Недавні досягнення в цій області підкреслили переваги та обмеження методів глибокого навчання у розпізнаванні зображень, і хоча їхні переваги очевидні, є ще певні аспекти, які потребують подальшого дослідження [8]-[10].

Унаслідок розвитку технології комп'ютерного зору (computer vision, CV) протягом останніх десятиліть, з'явилися нові методи динамічного розпізнавання та декодування штрих-кодів. За принципом CV, процес розпізнавання штрих-коду можна умовно розділити на дві основні фази: локалізацію та декодування самого штрих-коду. Для досягнення цих цілей системи розпізнавання штрих-кодів мають п'ять основних компонентів [8]-[10]. Проаналізуємо компоненти програмних засобів розпізнавання штрихових кодів [8]-[10].

Програмний модуль отримання зображень призначено для роботи з файлами цифрових зображень, які генеруються та отримуються з електронних пристроїв, таких як цифрові камери, телефони та сканери [8]-[10].

Програмні модулі попередньої обробки та сегментації зображень призначені для поліпшення якості зображення та підвищення точності аналізу наступних процесів. У цих модулях реалізуються методи покращення зображення, роздільної здатності та його якості [8]-[10]. Також у програмних модулях реалізуються функції зменшення розмірності зображення шляхом видалення зайвих компонентів та виділення необхідних ознак зі сканованого зображення [8]-[10].

Модулі класифікації ознак призначені для точного розпізнавання зображення. У таких програмних модулях реалізуються спеціальні правила класифікації малорозмірних об'єктів, отриманих на попередньому кроці. Ці правила, як правило, є наслідком застосування відповідних методів та засобів розпізнавання графічних образів [8]-[10].

В системі розпізнавання штрих-кодів, отримане зображення або штрих-код декодується за допомогою відповідного алгоритму, що дозволяє аналізувати та визначати необхідну інформацію [8]-[10].

Навіть при значному покращенні розпізнавання штрих-кодів методом комп'ютерного зору, він все ще має свої недоліки та обмеження. Завдяки великій кількості класів і ознак, оптимальні характеристики для класифікації різних візуальних об'єктів потребують ручного вибору. Крім того, декілька параметрів потребують налаштування залежно від кожної конкретної функції. Операція вимагає високого рівня технічних навичок, що провокує тривалий період проб і помилок. Останнім часом були спроби вдосконалення ручних елементів традиційних методів комп'ютерного зору, що описано вище для розпізнавання штрих-кодів. Глибоке навчання є одним із провідних методів та ключових рішень, які широко використовуються за їх успішне використання у локалізації та декодуванні штрих-кодів [8]-[10].

Розпізнавання штрихових кодів використовує підходи на основі глибокого навчання і має кілька переваг порівняно з традиційними методами. Для вирішення цієї задачі використовуються два основні методи глибокого навчання, зокрема, багат шарові нейронні мережі прямого поширення та згорткові нейронні мережі. Однією з ключових можливостей згорткових нейронних мереж є вирішення проблеми втрати інформації при перетворенні двовимірних зображень на одновимірні вектори [8]-[10]. Розроблено декілька архітектур згорткових мереж для розв'язання завдань комп'ютерного зору, зокрема, розпізнавання штрихових кодів, таких як згорткові мережі на основі регіонів, одиночний детектор та You Only Look Once (YOLO). Результати попередніх досліджень показали, що програми для розпізнавання штрих-кодів

на основі методів глибокого навчання можуть допомогти точно, швидко та надійно виявляти та декодувати штрих-коди. Вони також краще розв'язують проблеми розпізнавання штрих-кодів, зокрема з розмитими зображеннями та спотвореннями [8]-[10].

Визначено, що штрихові коди є числовими та лінійними символами, які знаходяться на товарах та допомагають в їх ідентифікації. Показано, що програмні системи оброблення штрих-кодів допомагають підприємствам в відстеженні товарів, управлінні цінами та запасами. Проаналізовано різні види штрихових кодів. Відзначено, що штрихові коди легко можна впровадити в бізнес, використовуючи відповідне програмне забезпечення. Проаналізовано переваги використання програмного забезпечення оброблення штрихових кодів для бізнесу. Відзначено, що використання штрихових кодів у поєднанні з програмними системами дозволяє компаніям завжди мати точну та повну інформацію про поточні запаси та їх рух. Відзначено, що розпізнавання штрихових кодів відноситься до завдання розпізнавання образів. Проаналізовано компоненти програмних засобів розпізнавання штрихових кодів, а також інтелектуальні методи розпізнавання штрихових кодів.

1.2 Аналіз програмного забезпечення для розпізнавання штрихових кодів

Проаналізуємо програмне забезпечення підтримки для розпізнавання штрихових кодів [11]-[15].

Програмне забезпечення QR Code Desktop Reader (рис. 1.1) призначено для зчитування QR-кодів за допомогою веб-камери комп'ютера. Воно дозволяє користувачам швидко і легко сканувати QR-коди, які знаходяться на різних об'єктах або екрані комп'ютера. Після сканування програма інтерпретує інформацію, закодовану в QR-коді, і може відобразити цю інформацію на екрані або зберегти для подальшого використання. Такі програми зазвичай

дуже прості у використанні та можуть працювати на різних операційних системах, таких як Windows, macOS або Linux [11], [14].

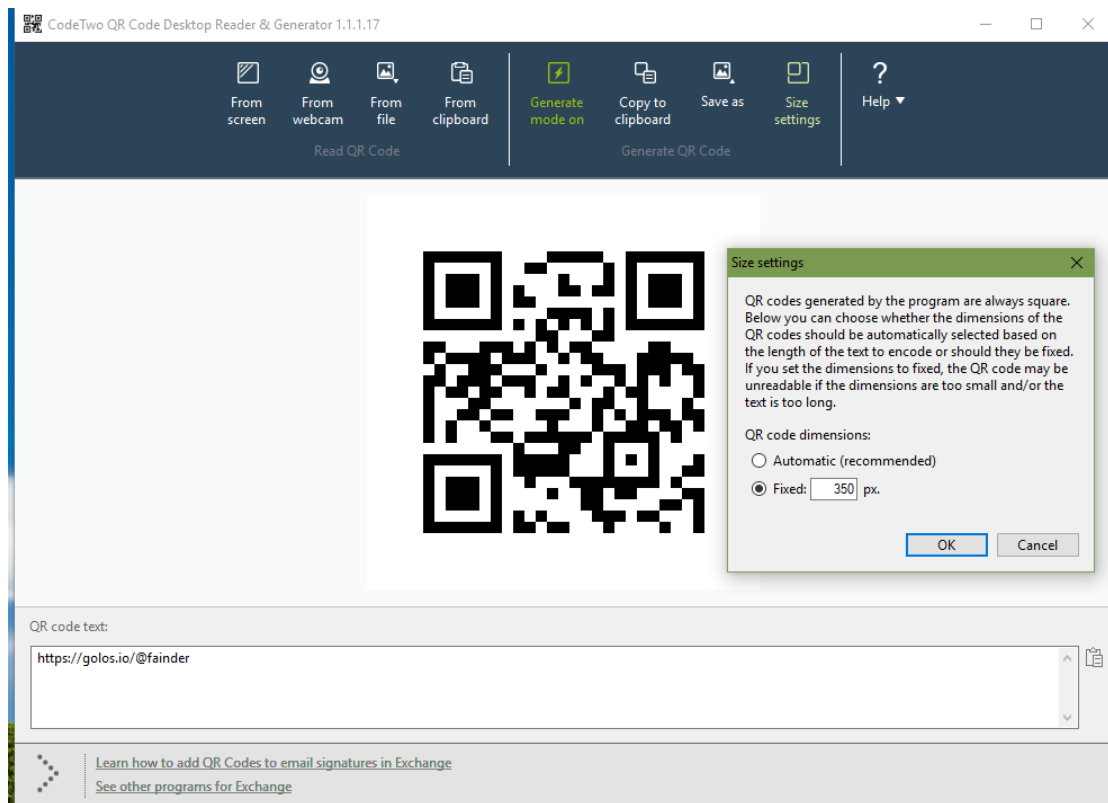


Рисунок 1.1 – Програмне забезпечення QR Code Desktop Reader [11], [14]

Головна функція програмного забезпечення QR Code Desktop Reader дозволяє розпізнавати штрихові коди на зображеннях. QR Code Desktop Reader може зчитувати штрих-код за допомогою кількох методів: від захоплення частини екрана до використання веб-камери, буфера обміну або завантаження файлу. Після обробки користувач отримує розшифровку тексту, закодованого у цьому штрих-коді [11], [14].

Сервіс має декілька переваг: безкоштовний, дає змогу користувачам створювати власні коди. Достатньо лише ввести текст у відповідне поле, і програма автоматично створить штрих-код. Потім зображення можна зберегти у форматі PNG або JPEG або скопіювати в буфер обміну [11], [14].

Серед особливостей програмного забезпечення QR Code Desktop Reader для розпізнавання штрихових кодів можна виділити такі [11], [14]:

- здатність до розпізнавання штрих-кодів: здатність розпізнавати QR-коди і штрих-коди з різних джерел, включаючи захоплення частини екрану, веб-камеру, буфер обміну або завантаження файлу [11], [14];

- доступність різних методів сканування: програма QR Code Desktop Reader пропонує кілька способів сканування штрих-кодів з різних джерел, що дозволяє користувачам вибрати найзручніший метод для них [11], [14];

- розшифрування інформації: після сканування програмне забезпечення QR Code Desktop Reader розшифровує текст, закодований у штрих-коді, і представляє його для користувача [11], [14];

- можливість створення власних кодів: QR Code Desktop Reader дає змогу користувачам створювати власні QR-коди, вводячи текст у відповідне поле, і отримувати готовий штрих-код для використання [11], [14];

- формати збереження і копіювання: після створення штрих-коду користувач може зберегти його у форматах PNG або JPEG або скопіювати в буфер обміну [11], [14];

- безкоштовне використання: програмне забезпечення QR Code Desktop Reader є безкоштовним для використання, що робить його доступним для широкого кола користувачів [11], [14].

Серед недоліків програмного забезпечення QR Code Desktop Reader можна відзначити те, що для деяких користувачів ця програма може бути важкою для експлуатації. Крім того, програмне забезпечення QR Code Desktop Reader не підтримує українську мову [11], [14].

Програмне забезпечення BarCode Descriptor дозволяє виконувати сканування звичайних штрих-кодів. Усі операції виконуються у вікні програмного забезпечення BarCode Descriptor. Користувачу потрібно лише ввести цифровий код, після чого він отримує зображення товару та додаткову інформацію, пов'язану з цим кодом [11], [15]. Програмне забезпечення BarCode Descriptor призначене для розпізнавання та аналізу штрих-кодів. Воно може використовуватися для сканування штрих-кодів, отримання інформації про продукти або відстеження облікових даних товарів. Програмне

забезпечення BarCode Descriptor (рис. 1.2) спрощує процес роботи з штрих-кодами, дозволяючи користувачам швидко отримати інформацію, пов'язану з цими кодами, і можливо, зображення товарів [11], [15].

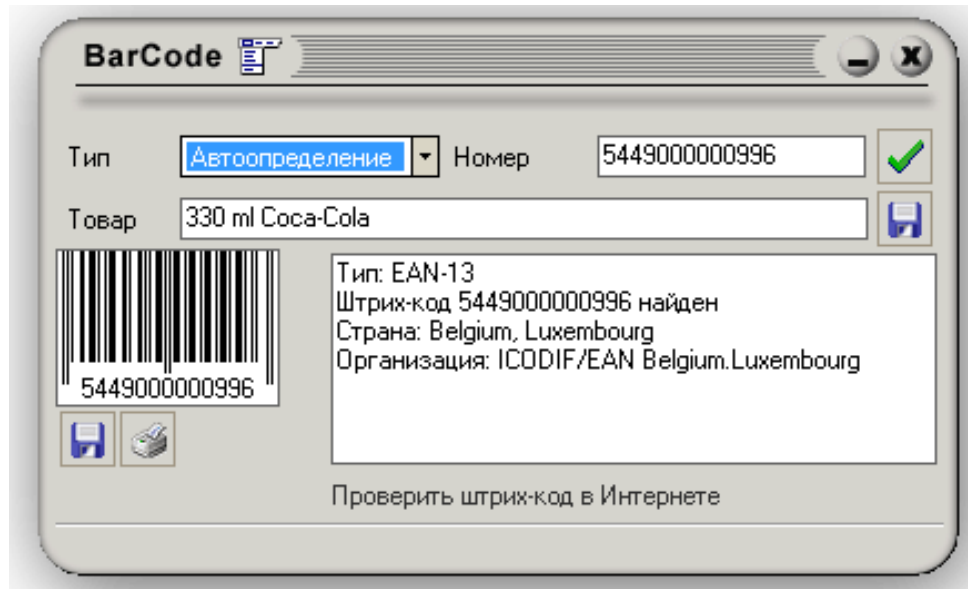


Рисунок 1.2 – Програмне забезпечення BarCode Descriptor [11], [15]

Основне завдання програми BarCode Descriptor полягає у розпізнаванні штрих-кодів, які знаходяться на зображеннях. Це програмне забезпечення вміє визначати штрих-коди за допомогою різних способів: від захоплення області екрану до використання веб-камери, буфера обміну або завантаження файлу. Після обробки цих даних користувач отримує розшифрований текст, який був закодований у цьому штрих-коді [11], [15].

Програмне забезпечення BarCode Descriptor має кілька переваг. Воно є безкоштовним і надає можливість користувачам створювати свої власні коди. Для цього достатньо лише ввести текст у спеціальне поле, і програма автоматично створить штрих-код. Після цього зображення можна зберегти у форматах PNG або JPEG або скопіювати його в буфер обміну [11], [15].

Серед особливостей програмного забезпечення BarCode Descriptor для розпізнавання штрихових кодів можна виділити такі [11], [15]:

– простота у використанні: інтерфейс програмного забезпечення BarCode Descriptor досить інтуїтивно зрозумілий, що робить процес створення та розпізнавання штрих-кодів більш зручним для користувача [11], [15];

– доступність різних методів сканування: програма BarCode Descriptor пропонує кілька способів сканування штрих-кодів з різних джерел, що дозволяє користувачам вибрати найзручніший метод для них [11], [15];

– здатність до розпізнавання штрих-кодів: здатність розпізнавати QR-коди і штрих-коди з різних джерел, включаючи захоплення частини екрану, веб-камеру, буфер обміну або завантаження файлу [11], [15];

– розшифрування інформації: після сканування програмне забезпечення BarCode Descriptor розшифровує текст, закодований у штрих-коді, і представляє його для користувача [11], [15];

– можливість створення власних кодів: BarCode Descriptor дає змогу користувачам створювати власні QR-коди, вводячи текст у відповідне поле, і отримувати готовий штрих-код для використання [11], [15];

– формати збереження і копіювання: після створення штрих-коду користувач може зберегти його у різних форматах або скопіювати в буфер обміну [11], [15];

– безкоштовний функціонал: програмне забезпечення BarCode Descriptor є безкоштовним програмним забезпеченням, що робить його доступним для широкого кола користувачів [11], [15].

Серед недоліків програмного забезпечення BarCode Descriptor можна відзначити те, що для певних користувачів вона може виявитися складною у використанні. Крім того, програмне забезпечення BarCode Descriptor не має підтримки української мови [11], [15].

Програмне забезпечення TNR CodeExpert (рис. 1.3) призначене для розпізнавання штрих-кодів. Це програмне забезпечення дозволяє розшифрувати або створити штрих-код у різних кодуваннях, здійснити пошук інформації про продукти за їхніми кодами, розпізнати символи на товарах і

одязі, а також визначити телефонний код міста або країни за номером телефону [11], [16].

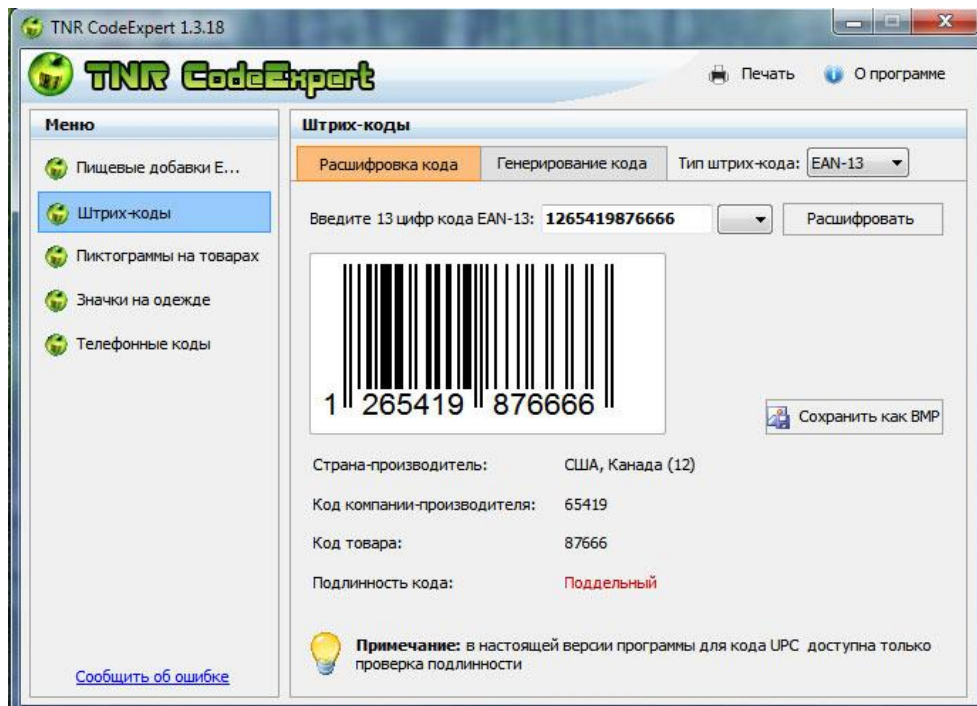


Рисунок 1.3 – Програмне забезпечення TNR CodeExpert [11], [16]

Основна функція програми TNR CodeExpert полягає у розпізнаванні штрих-кодів, які містяться на зображеннях. Це програмне забезпечення може впізнавати штрих-коди за допомогою різних методів: від захоплення області екрану до використання веб-камери, буфера обміну або завантаження файлу. Після обробки цих даних користувач отримує розшифрований текст, який був закодований у вказаному штрих-коді [11], [16].

Програмне забезпечення TNR CodeExpert має можливість роботи як дистанційна платформа для відстеження активів. Ця платформа може перетворити звичайні мобільні пристрої або планшети, на сканери штрихових кодів, замінюючи потребу у стандартних сканерах [11], [16].

Інтерфейс користувача програмного забезпечення TNR CodeExpert дозволяє налаштувати програму для відстеження конкретних активів та оптимізації робочих процесів. Крім цього, програмне забезпечення TNR CodeExpert пропонує різноманітні функції, такі як інтерактивні інформаційні

панелі, відстеження амортизації, вбудовану технологію оброблення та створення штрихових кодів [11], [16].

Це програмне забезпечення дозволяє працювати великій кількості працівників компанії. Програмне забезпечення TNR CodeExpert дає можливість призначати аудити окремим користувачам або надавати високому керівництву доступ до інформаційних панелей. Всі користувачі отримують доступ до потрібної інформації на основі наданих дозволів [11], [16].

Серед особливостей програмного забезпечення TNR CodeExpert для розпізнавання штрихових кодів можна виділити такі [11], [16]:

- висока швидкість зчитування: програмне забезпечення TNR CodeExpert забезпечує досить швидке зчитування та розпізнавання штрихових кодів [11], [16];

- можливість створення власних зображень штрихових кодів: TNR CodeExpert дає змогу користувачам створювати власні штрихові коди, вводячи текст у відповідне поле, і отримувати готовий штриховий код для використання [11], [16];

- простота у використанні: інтерфейс програмного забезпечення TNR CodeExpert досить інтуїтивно зрозумілий, що робить процес створення та розпізнавання штрих-кодів більш зручним для користувача [11], [16];

- доступність різних методів сканування: програма TNR CodeExpert пропонує кілька способів сканування штрих-кодів з різних джерел, що дозволяє користувачам вибрати найзручніший метод для них [11], [16];

- здатність до розпізнавання штрих-кодів на будь-якій поверхні: здатність розпізнавати штрихові коди з різних джерел, включаючи захоплення частини екрану, веб-камеру, буфер обміну або завантаження файлу [11], [16];

- розшифрування інформації: після сканування програмне забезпечення TNR CodeExpert розшифровує текст, закодований у штрих-коді, і представляє його для користувача [11], [16];

– формати збереження і копіювання: після створення штрих-коду користувач може зберегти його у різних форматах або скопіювати в буфер обміну [11], [16].

Серед недоліків програмного забезпечення TNR CodeExpert можна відзначити те, що для певних користувачів вона може виявитися складною у використанні. Крім того, програмне забезпечення TNR CodeExpert не має підтримки української мови [11], [16].

Порівняльну характеристику програмного забезпечення для розпізнавання штрихових кодів наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння програмного забезпечення для розпізнавання штрихових кодів

Критерій порівняння	QR Code Desktop Reader [14]	BarCode Descriptor [15]	TNR CodeExpert [16]
Зручність роботи для розпізнавання штрихових кодів	+-	+-	+
Підтримка великої кількості форматів файлів з зображеннями	+	+-	-
Безкоштовність	+	+	-
Підтримка багатьох мов	-	-	+
Інтеграція з іншими програмними засобами обробки зображень	-	-	+

За результатами проведеного аналізу можна зробити висновок, що у наш час існує досить багато програмних засобів для розпізнавання штрихових кодів. Проте деякі програми для розпізнавання штрихових кодів можуть бути обмеженими у підтримці різних форматів штрихових кодів, що може викликати проблеми при розпізнаванні певних типів кодів. При використанні

інших програмних засобів, особливо при обробці великих обсягів даних або на повільних пристроях, час, необхідний для розпізнавання, може бути значним. Крім того, багато програмних засобів для розпізнавання штрихових кодів вимагають чіткості та яскравості штрихового коду, внаслідок чого низька якість друку, пошкодження або розмивання штрихового коду можуть призвести до неправильного або некоректного розпізнавання. Також освітлення та кути зору можуть впливати на точність розпізнавання, відповідно, недостатнє або неправильне освітлення може призвести до помилок у розпізнаванні. Деякі програмні засоби для розпізнавання штрихових кодів можуть вимагати дорогого обладнання або платних програмних рішень, що в деяких випадках обмежує їх застосування на практиці, особливо в невеликих організаціях. Тому актуальною є розробка програмного забезпечення для розпізнавання штрихових кодів.

При розробці програмного забезпечення для розпізнавання штрихових кодів необхідно забезпечити такі функціональні вимоги:

- можливість розпізнавання штрихових кодів;
- можливість вибору файлу з графічним зображенням;
- можливість пошуку товару за розпізнаним штриховим кодом;
- можливість змінити зображення штрихового коду;
- можливість декодування штрихових кодів різних форматів.

1.3 Висновки за розділом 1

Визначено, що штрихові коди є числовими та лінійними символами, які знаходяться на товарах та допомагають в їх ідентифікації. Показано, що програмні системи оброблення штрих-кодів допомагають підприємствам в відстеженні товарів, управлінні цінами та запасами. Проаналізовано різні види штрихових кодів. Відзначено, що штрихові коди легко можна впровадити в бізнес, використовуючи відповідне програмне забезпечення. Проаналізовано переваги використання програмного забезпечення оброблення штрихових

кодів для бізнесу. Відзначено, що використання штрихових кодів у поєднанні з програмними системами дозволяє компаніям завжди мати точну та повну інформацію про поточні запаси та їх рух. Відзначено, що розпізнавання штрихових кодів відноситься до завдання розпізнавання образів. Проаналізовано компоненти програмних засобів розпізнавання штрихових кодів, а також інтелектуальні методи розпізнавання штрихових кодів.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для розпізнавання штрихових кодів. Проте деякі програми для розпізнавання штрихових кодів можуть бути обмеженими у підтримці різних форматів штрихових кодів, що може викликати проблеми при розпізнаванні певних типів кодів. При використанні інших програмних засобів, особливо при обробці великих обсягів даних або на повільних пристроях, час, необхідний для розпізнавання, може бути значним. Крім того, багато програмних засобів для розпізнавання штрихових кодів вимагають чіткості та яскравості штрихового коду, внаслідок чого низька якість друку, пошкодження або розмивання штрихового коду можуть призвести до неправильного або некоректного розпізнавання. Також освітлення та кути зору можуть впливати на точність розпізнавання, відповідно, недостатнє або неправильне освітлення може призвести до помилок у розпізнаванні. Деякі програмні засоби для розпізнавання штрихових кодів можуть вимагати дорогого обладнання або платних програмних рішень, що в деяких випадках обмежує їх застосування на практиці, особливо в невеликих організаціях. Тому актуальною є розробка програмного забезпечення для розпізнавання штрихових кодів.

Сформульовано функціональні вимоги до програмного забезпечення для розпізнавання штрихових кодів.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування

При розробці програмного забезпечення для розпізнавання штрихових кодів було використано мову програмування C# [17], [18].

C# є мовою програмування, спрямованою на об'єктно-орієнтований підхід. Вона була створена групою інженерів Microsoft для розробки програм під платформу Microsoft .NET Framework [17], [18].

Мова C# входить до сімейства мов типу C, має синтаксис, близький до C++ та Java. У неї впроваджена статична типізація, поліморфізм, перевантаження операторів (включаючи явні та неявні приведення), делегати, атрибути, події, загальні типи та методи, анонімні функції з підтримкою закриття, винятки та коментарі [17], [18].

Переваги мови програмування C# включають наступні. Висока сумісність з мовою C, що дозволяє використовувати існуючий код C з незначними змінами [17], [18].

Мова програмування C# підтримує різні стилі і технології програмування, включаючи загальне програмування та об'єктно-орієнтоване програмування [17], [18].

Мова програмування C# надає можливість розробки простих контейнерів та алгоритмів для типів даних та їх обчислення на етапі трансляції за допомогою моделей [17], [18].

C# є об'єктно-орієнтованою мовою програмування, розробленою Microsoft для платформи .NET Framework. Ця мова програмування підтримує об'єктно-орієнтоване програмування, що дозволяє створювати програми, орієнтовані на об'єкти, що спрощує розробку складних систем. Також мова програмування C# має статичну типізацію, що дозволяє виявляти помилки на етапі компіляції програмного забезпечення, що робить програми надійнішими та ефективнішими [17], [18].

Однією з особливостей C# є підтримка платформи .NET Framework, що дозволяє використовувати багато вбудованих бібліотек та інструментів для розробки різноманітних програмних продуктів, від мобільних програм до настільного програмного забезпечення [17], [18].

Також C# підтримує асинхронне програмування, яке дозволяє створювати додатки, що працюють швидше та ефективніше, що спрощують роботу з послідовностями даних [17], [18].

Унікальність C# полягає в його здатності поєднувати в собі потужність, простоту та високий рівень безпеки, роблячи його однією з переважних мов програмування для різних типів проєктів [17], [18].

Обґрунтування вибору мови програмування для розробки програмного забезпечення для розпізнавання штрихових кодів наведено у таблиці 2.1.

Таблиця 2.1 – Обґрунтування вибору мови програмування для розробки програмного забезпечення для розпізнавання штрихових кодів

Критерій порівняння мов програмування	Мова програмування		
	C#	Python	C++
Зручність розробки програмного забезпечення для розпізнавання штрихових кодів	+	—	—
Робота з графічною інформацією	+	+	+—
Засоби спадкування	+	+—	+
Засоби створення інтерфейсу користувача	+	+—	+—
Наявність бібліотек для розпізнавання графічних образів	+	+	+—

Отже, для реалізації програмного забезпечення для розпізнавання штрихових кодів обрано мову програмування C#, яка є мовою програмування,

що підтримує статичну типізацію, поліморфізм, перевантаження операторів (включаючи явні та неявні приведення), делегати, атрибути, події, загальні типи та методи, анонімні функції, винятки та коментарі, що є досить зручним при розробці відповідного програмного забезпечення.

2.2 Вибір середовища розробки для розпізнавання штрихових кодів

В якості середовища розробки для створення програмного забезпечення для розпізнавання штрихових кодів було обрано Microsoft Visual Studio [19], [20].

Microsoft Visual Studio є платформою для розробки програм на мові C#, створеною компанією Microsoft. Це інтегроване середовище розробки, яке вважається одним із найзручніших та найефективніших інструментів для програмування на C# [19], [20].

Основні можливості середовища розробки Microsoft Visual Studio IDE включають:

- аналіз коду, підсвічування помилок та відображення синтаксичних підказок;
- зручна навігація по коду та проекту: швидкий перехід між різними файлами, класами та методами;
- функціонал для рефакторингу: можливість перейменування, видалення методів, змінних, констант тощо;
- вбудований відгадчик для мови C#;
- підтримка різних систем контролю версій [19], [20].

Microsoft Visual Studio є комплексним середовищем розробки спеціально для мови програмування C#, обладнане багатьма корисними функціями. Воно надає можливості для кодування аналізу, візуального налагодження, модульного тестування [19], [20].

Для програмістів, які працюють з C#, Microsoft Visual Studio є зручним інструментом. Воно не просто постачається з широким спектром корисних можливостей, але також допомагає аналізувати та оптимізувати код, підвищуючи ефективність розробки програмного забезпечення [19], [20].

Однією з ключових переваг Microsoft Visual Studio є його вбудована можливість візуального налагодження, яка полегшує виявлення та виправлення помилок у коді. Платформа також має інструменти для ефективного модульного тестування, що забезпечує високу якість програмного продукту [19], [20].

Серед основних можливостей Microsoft Visual Studio варто виділити його здатність підтримувати розробку застосунків у різних фреймворках.

Обґрунтування вибору середовища розробки для створення програмного забезпечення для розпізнавання штрихових кодів наведено у таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору середовища розробки для створення програмного забезпечення для розпізнавання штрихових кодів

Критерій порівняння середовищ розробки	Середовище розробки		
	Visual Studio	MonoDevelop	SharpDevelop
Зручність розробки на мові C#	+	+—	+—
Засоби для аналізу коду	+	+	+—
Велика спільнота розробників	+	+	—
Підтримка різних платформ	+	+	—
Розширені можливості для відладки	+	—	—

Таким чином, для створення програмного забезпечення для розпізнавання штрихових кодів обрано середовище розробки Microsoft Visual Studio, що є досить ефективним та зручним інструментом для програмування на C#, а також для розробки програмного забезпечення різного призначення.

2.3 Методи та алгоритми аналізу даних для розпізнавання штрихових кодів

Програмне забезпечення розпізнавання штрихових кодів включає в себе кілька пов'язаних завдань: визначення місця розташування штрих-коду на об'єкті, виділення складових частин штрих-коду, розшифрування інформації, закодованої у штриховому коді [21]–[23].

Алгоритм виявлення штрих-кодів має фазу, що визначає область зображення, яка містить штрихову символіку. Для досягнення цієї мети використовувався метод фільтрації з використанням багатьох каналів, при цьому для кожного обробленого зображення застосовувався випадковий ієрархічний пошук горизонтальних сегментів однакової довжини. Далі для цих сегментів визначалася частотна характеристика. Якщо вона перевищувала задане значення, вважалося, що цей сегмент представляє лінію з високою частотою. Подальше порівняння високочастотних сегментів здійснювалося за допомогою алгоритму найбільшої спільної підпоследовності. Для обмеження рамки вибиралися лінії з високою частотою та подібністю. Гістограма обчислювалася вздовж осі X та Y для усіх парних вибраних ліній, з орієнтацією на те, що гістограма зображення штрих-коду матиме один пік з великою амплітудою. Розташування штрих-коду визначалося через аналіз початкових і кінцевих координат вершин піків, а його орієнтація - як кут повороту зображення [21]–[23].

Алгоритм виділення структурних елементів штрих-коду дозволяє визначити ширину найменшого елемента в штрих-коді, відомого як модуль.

Використовуючи цю ширину та місце розташування елементів на зображенні, він визначає товщину самого штриху і проміжку між ними [21]–[23].

Кожній точці вихідного сірого зображення (яку ми називаємо материнською) відповідає три точки на отриманому зображенні (які ми називаємо нащадками). Ці нащадки містять набори значень яскравості (як одиниць, так і нулів), отриманих після порівняння значень з порогоми материнської точки. При цьому однакові значення групуються за місцем розташування [21]–[23].

Наступний крок у розпізнаванні штрих-коду полягає у зборі статистики щодо ширини самих штрихів та інтервалів між ними. Ця операція включає створення гістограми, яка відображає кількість штрихів та проміжків в залежності від їх ширини [21]–[23].

Отримані діапазони значень ширини можуть бути переведені в конкретні величини, які вимірюються в цілих кратних значеннях ширини модуля (мінімального елемента штрих-коду). Цей вектор значень потім передається для використання в процесі декодування штрих-коду [21]–[23].

Алгоритм розшифрування штрих-коду працює із значеннями ширини елементів, які складають штрихи та проміжки, вимірюваних у модулях ширини [21]–[23].

Цей алгоритм виконує розшифрування символів, які закодовані через ширину штрихів та проміжків [21]–[23].

Знак-роздільник є важливим символом, який служить для відділення інформаційних знаків і додаткового символу [21]–[23]. Парний паритет – характеристика кодування знаку символу, що вказує, чи міститься парна кількість темних стрічок (сума модулів штрихів для знаку з парним паритетом є парним числом). Непарний паритет – це характеристика кодування знаку символу, яка показує, чи міститься непарна кількість темних штрихових кодів (сума модулів штрихів для знаку з непарним паритетом є непарним числом) [21]–[23].

Знак-обмежувач – це певна послідовність штрихів та проміжків, що може відповідати відповідним знакам в інших системах кодування, або слугувати для розділення основного символу на дві половини. Набір знаків – це набір комбінацій штрихів та проміжків з парним або непарним паритетом, які кодують цифри від нуля до дев'яти [21]–[23].

Програмні системи читання штрих-кодів, що ґрунтуються на відповідних алгоритмах декодування, здатні розпізнавати символи у різних системах кодування. Для цього використовується алгоритм, що включає наступні кроки: підтвердження наявності початкової вільної зони, далі для кожного знака символу виконується вимірювання загальної ширини п'яти штрихів і чотирьох проміжків, обчислення граничного значення та порівняння ширини кожного елемента знака з відповідним значенням. Якщо елемент ширший, його вважають широким, в іншому випадку – вузьким. Після цього виконується визначення, чи утворена комбінація широких і вузьких елементів дійсно відповідає одному з певного набору знаків. Перший прочитаний знак символу має бути знаком старт або стоп, що допомагає визначити напрямок зчитування. Далі продовжується зчитування знаків до тих пір, поки не буде виявлено дійсний знак старт або стоп. Після цього виконується підтвердження наявності кінцевої вільної зони.

Отже, запропонуємо метод розпізнавання штрихових кодів на основі інтелектуальних обчислень (рис. 2.1), що складається з етапів зчитування зображення, визначення місця розташування штрих-коду на об'єкті, виділення складових частин штрих-коду, розшифрування інформації, закодованої у штриховому коді, виведення результату.

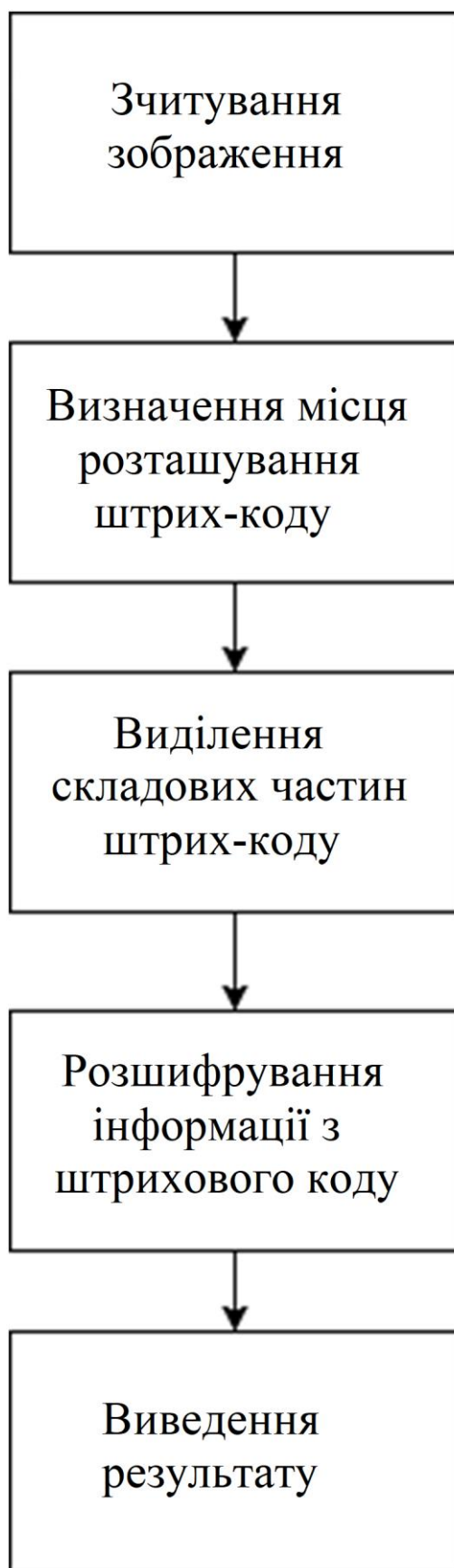


Рисунок 2.1 – Схема методу розпізнавання штрихових кодів на основі інтелектуальних обчислень

2.4 Висновки за розділом 2

Для реалізації програмного забезпечення для розпізнавання штрихових кодів обрано мову програмування C#, яка є мовою програмування, що підтримує статичну типізацію, поліморфізм, перевантаження операторів (включаючи явні та неявні приведення), делегати, атрибути, події, загальні типи та методи, анонімні функції, винятки та коментарі, що є досить зручним при розробці відповідного програмного забезпечення.

Для створення програмного забезпечення для розпізнавання штрихових кодів обрано середовище розробки Microsoft Visual Studio, що є досить ефективним та зручним інструментом для програмування на C#, а також для розробки програмного забезпечення різного призначення.

Запропоновано метод розпізнавання штрихових кодів на основі інтелектуальних обчислень, що складається з етапів зчитування зображення, визначення місця розташування штрих-коду на об'єкті, виділення складових частин штрих-коду, розшифрування інформації, закодованої у штриховому коді, виведення результату.

3 ОПИС ПРОГРАМИ

3.1 Структура програмного забезпечення для розпізнавання штрихових кодів

Загальна діаграма класів програмного забезпечення для розпізнавання штрихових кодів зображена на рис. 3.1.

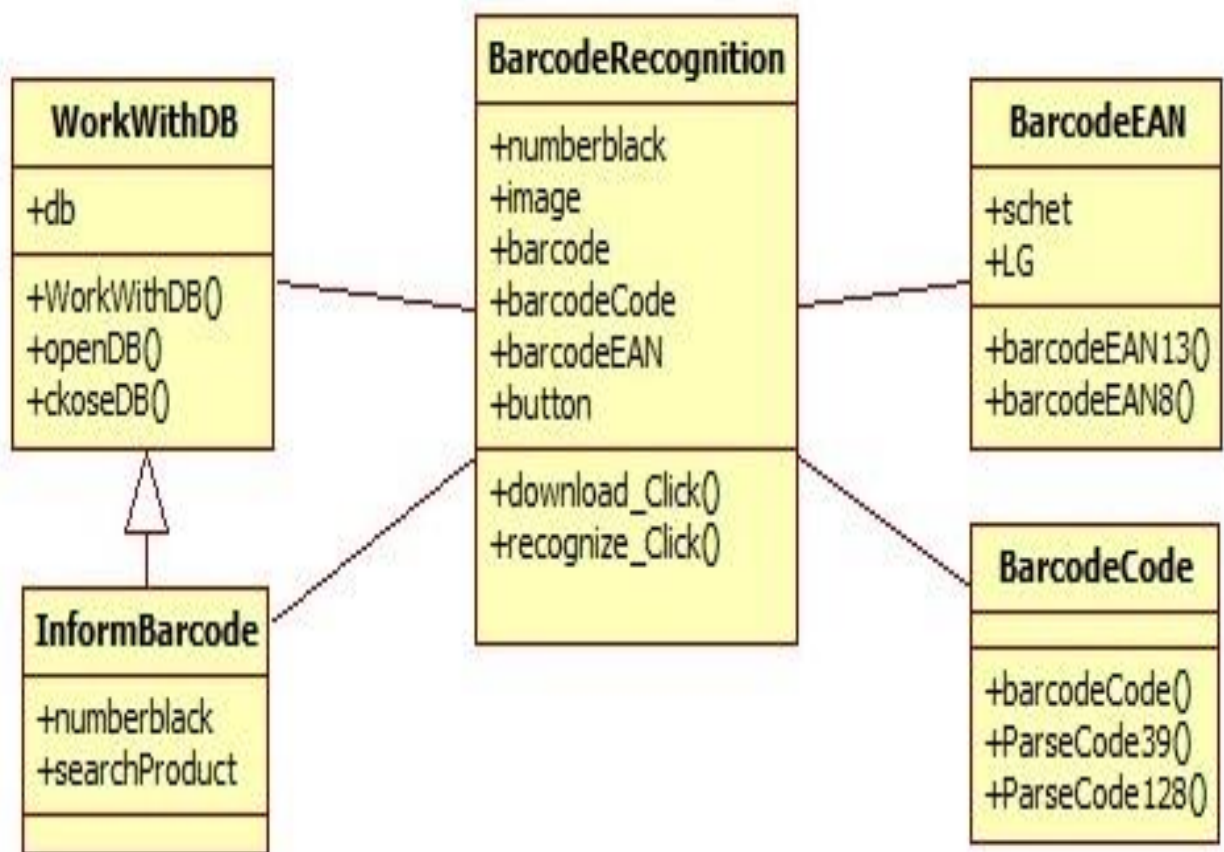


Рисунок 3.1 – Діаграма класів програмного забезпечення для розпізнавання штрихових кодів

Як видно з діаграми, основними класами програмного забезпечення для розпізнавання штрихових кодів є: штрих-код (**BarcodeCode**), штриховий код у системі EAN (**BarcodeCodeEAN**), розпізнавання коду (**BarcodeCodeRecognition**), інформація про штриховий код (**InformBarcode**) та інші.

Діаграма компонентів програмного забезпечення для розпізнавання штрихових кодів зображена на рис. 3.2.

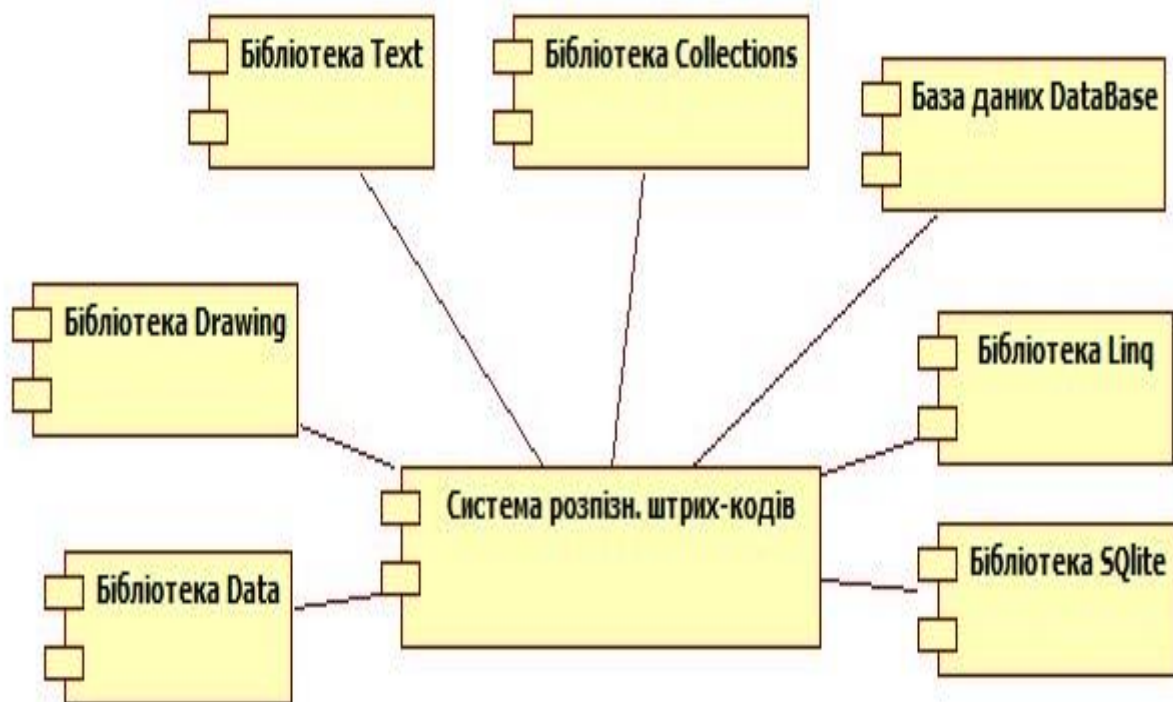


Рисунок 3.2 – Діаграма компонентів програмного забезпечення для розпізнавання штрихових кодів

3.2 Функціонування програмного забезпечення для розпізнавання штрихових кодів

Схема функціонування програмного забезпечення для розпізнавання штрихових кодів наведена на рис. 3.3.

Як видно з рис. 3.3 на першому етапі відбувається виведення головної форми програми. Після цього вхідні дані надходять у вигляді посилання на файл із зображенням штрихового коду. Потім відбувається перевірка коду на допустимість, декодування знаків та передача декодованих знаків. Далі відбувається виведення інформації про штриховий код, та, за необхідності, припинення роботи програми.

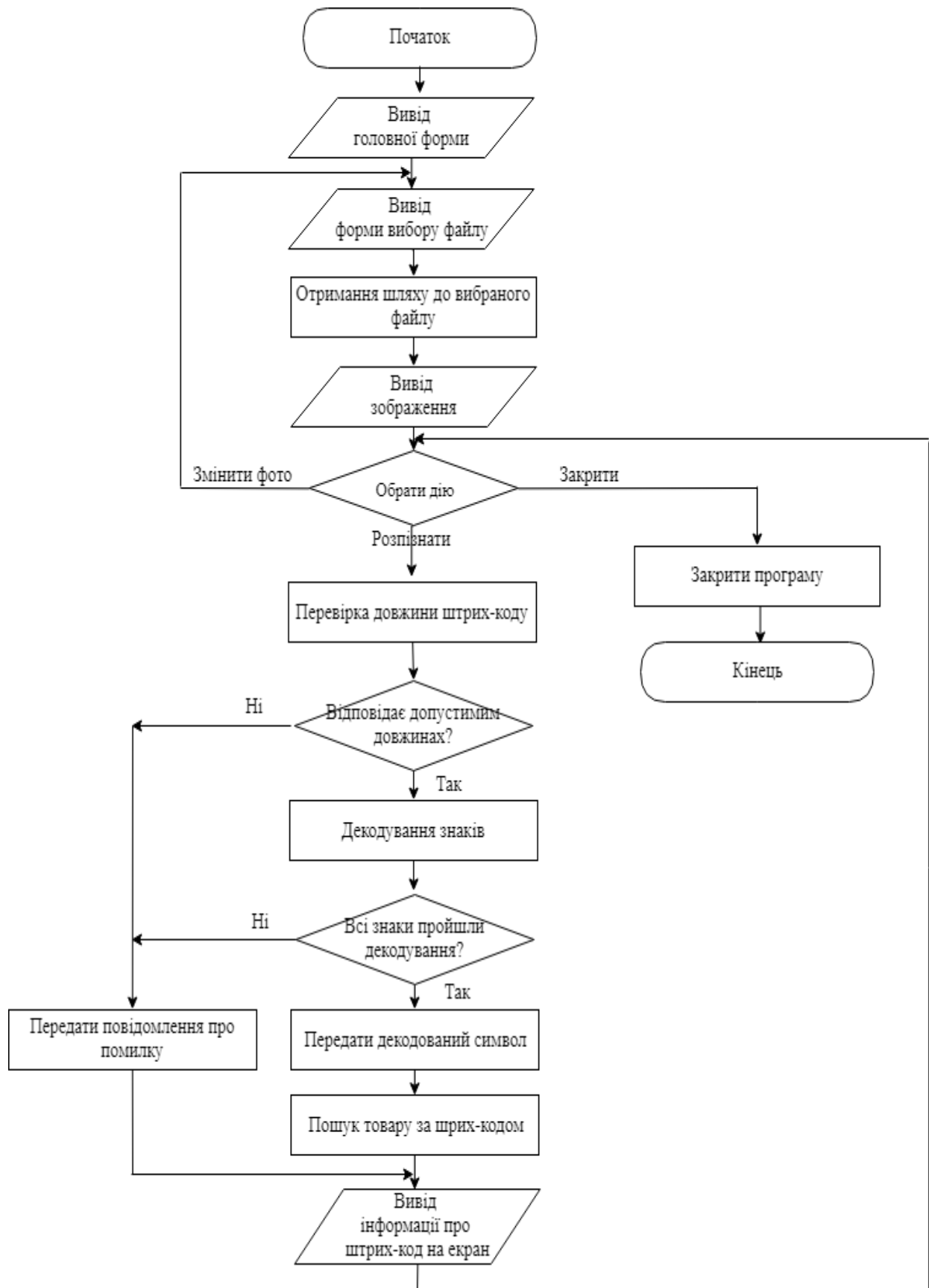


Рисунок 3.3 – Схема функціонування програмного забезпечення для розпізнавання штрихових кодів

Наведемо також схему функціонування програмного забезпечення для розпізнавання штрихових кодів за допомогою діаграми кооперацій, що відображена на рис. 3.4.

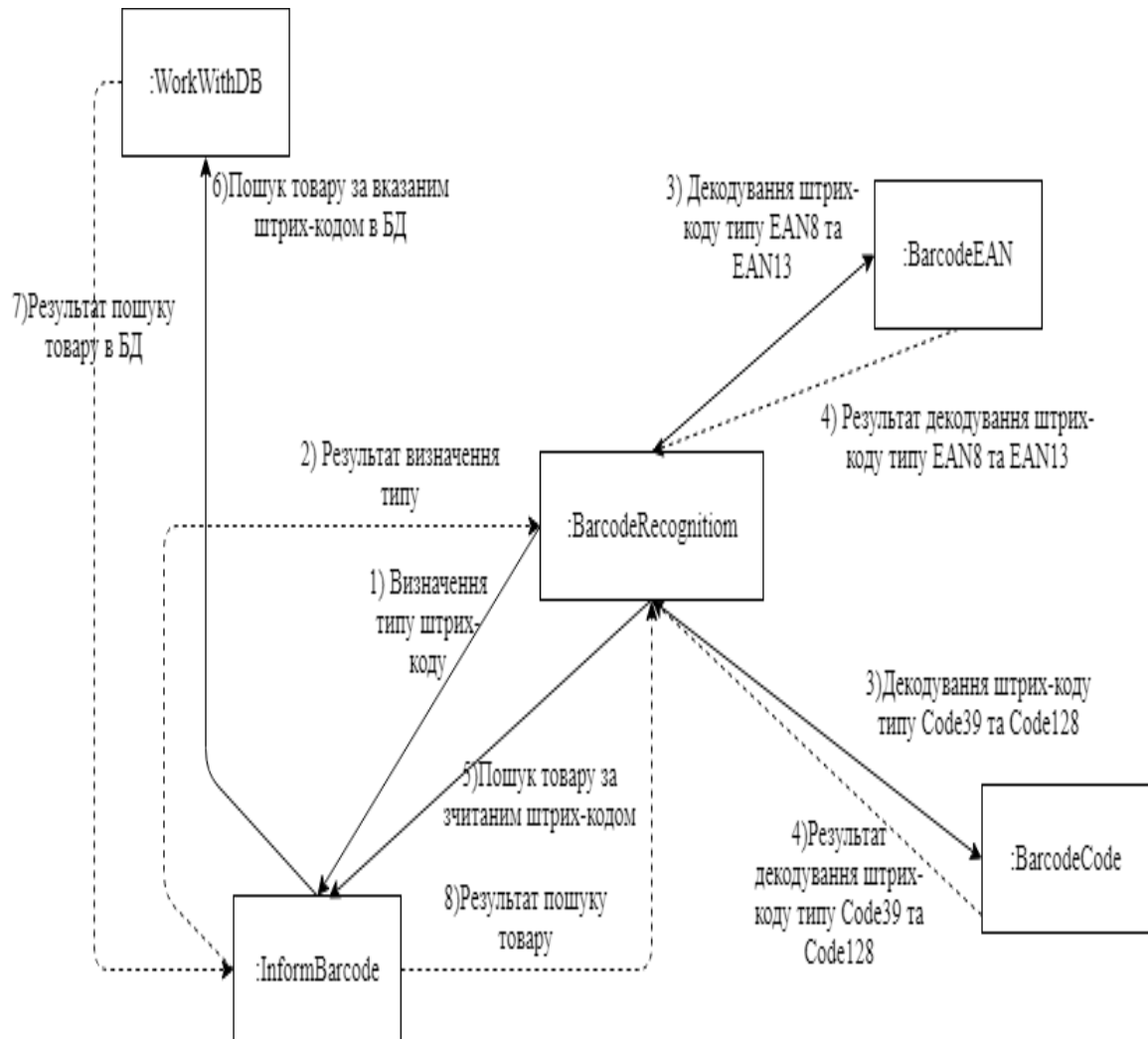


Рисунок 3.4 – Діаграма кооперацій програмного забезпечення для розпізнавання штрихових кодів

Для відображення роботи програмного забезпечення для розпізнавання штрихових кодів з точки зору його варіантів використання наведемо діаграму прецедентів (рис. 3.5).

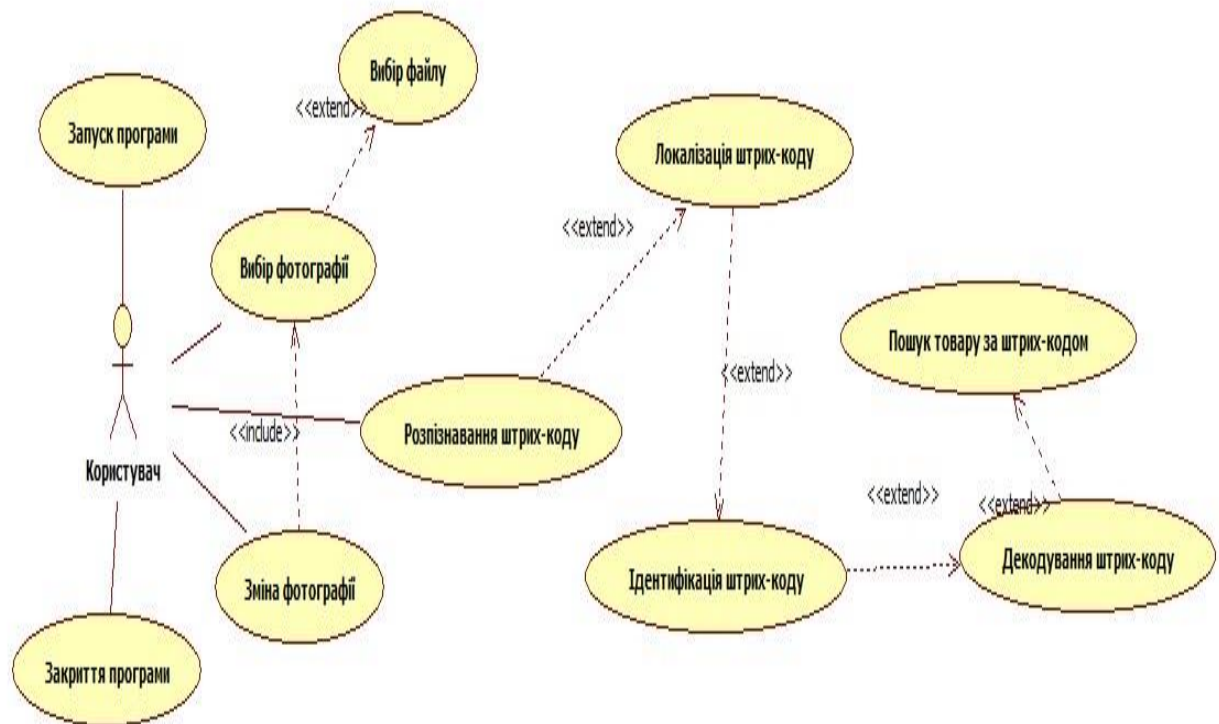


Рисунок 3.5 – Діаграма прецедентів програмного забезпечення для розпізнавання штрихових кодів

3.3 Проєктування інтерфейсу програмного забезпечення для розпізнавання штрихових кодів

Проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для розпізнавання штрихових кодів виконано з урахуванням вимог технічного завдання.

Згідно технічного завдання програмного забезпечення для розпізнавання штрихових кодів забезпечує можливість розпізнавання штрихових кодів, вибору файлу з графічним зображенням, пошуку товару за розпізнаним штриховим кодом, можливість декодування штрихових кодів різних форматів та ін.

Приклад головної форми програми для розпізнавання штрихових кодів наведено на рис. 3.6.

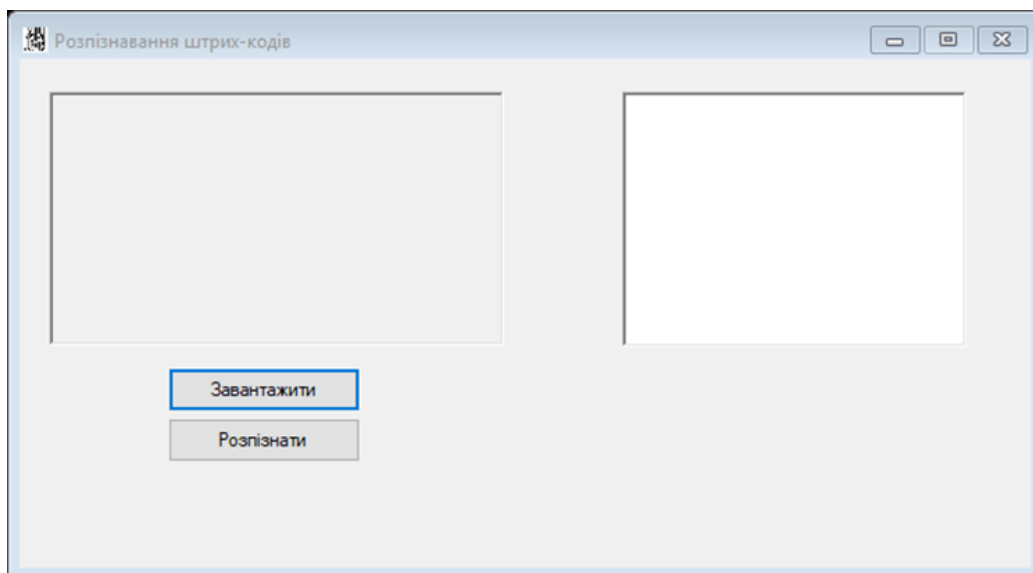


Рисунок 3.6 – Головна форма програми для розпізнавання штрихових кодів

Приклад діалогового вікна для вибору файлу для розпізнавання штрихового коду наведено на рис. 3.7.

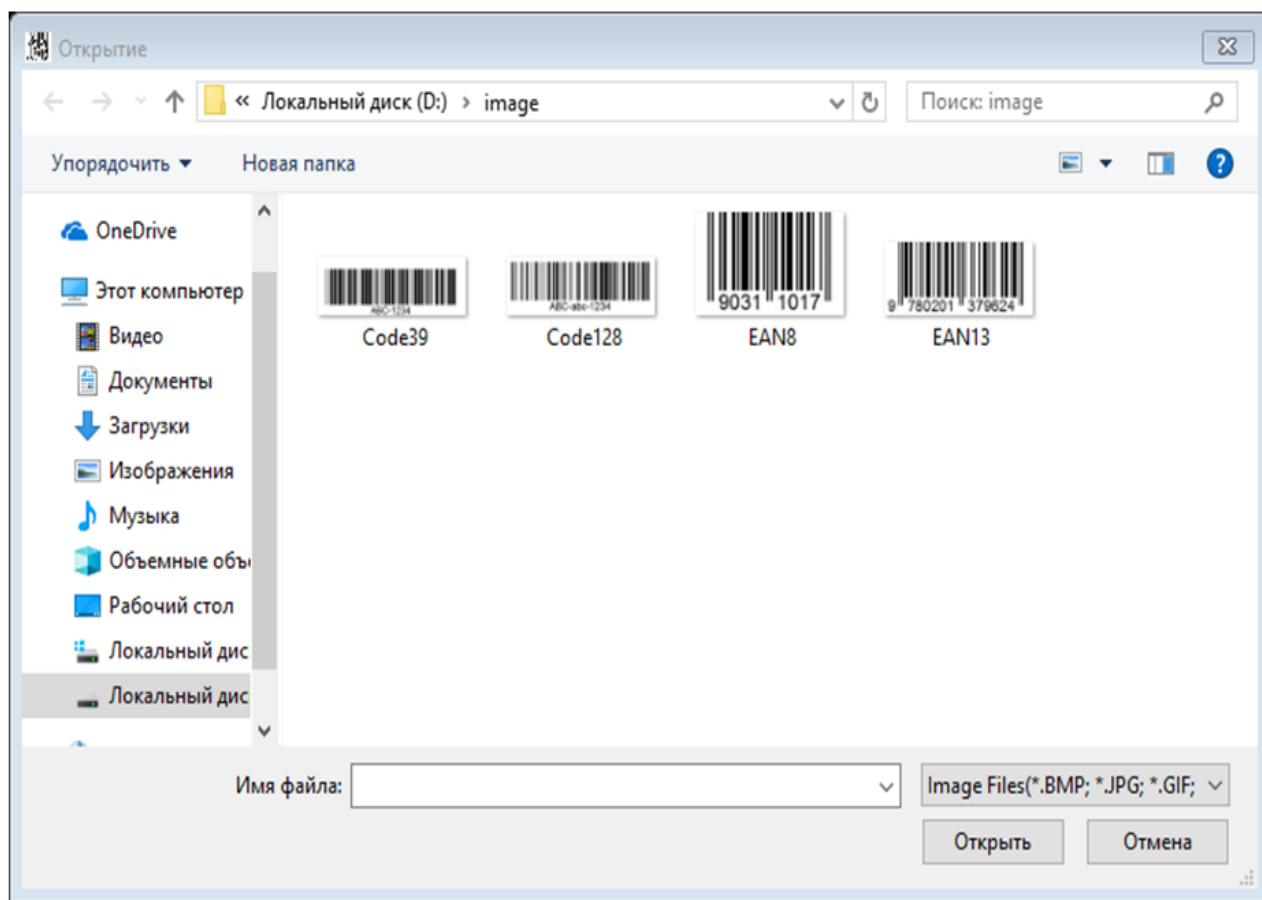


Рисунок 3.7 – Діалогове вікно для вибору файлу з штриховим кодом

Приклад результату розпізнавання штрихового коду за допомогою програми наведено на рис. 3.8.

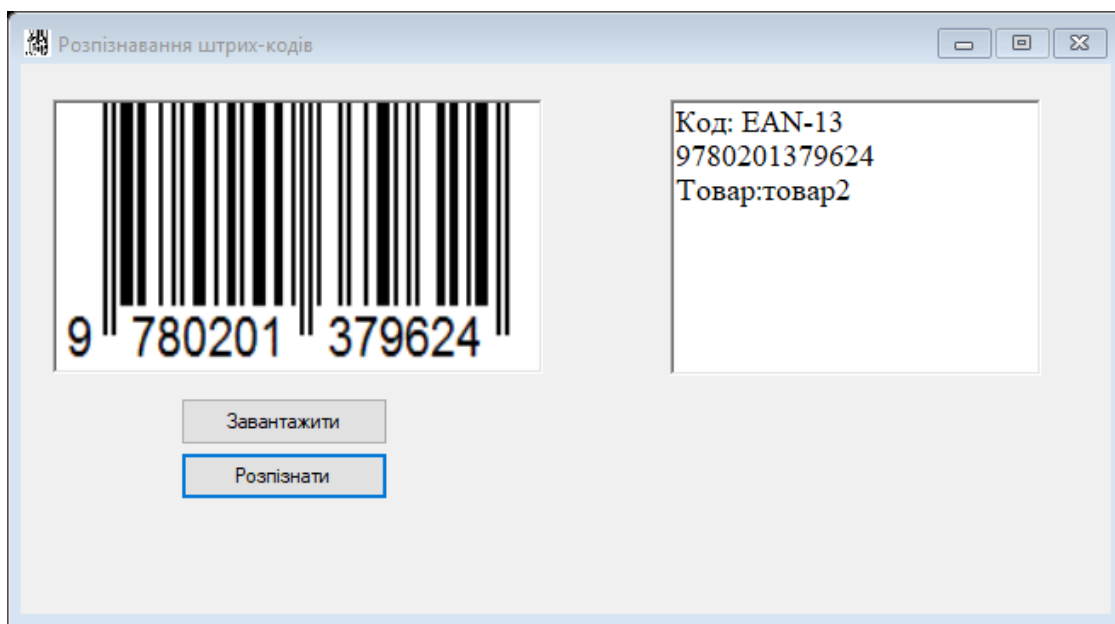


Рисунок 3.8 – Приклад результату розпізнавання штрихового коду

3.4 Висновки за розділом 3

Розроблено програмне забезпечення для розпізнавання штрихових кодів. Запропоновано структуру та описано основні модулі програмного забезпечення для розпізнавання штрихових кодів.

Описано функціонування програмного забезпечення для розпізнавання штрихових кодів. Описано особливості реалізації програмного забезпечення для розпізнавання штрихових кодів. Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для розпізнавання штрихових кодів.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Програма призначена для розв'язання задачі розпізнавання штрихових кодів. Програма написана на мові С#, яка є мовою програмування, що підтримує статичну типізацію, поліморфізм, перевантаження операторів (включаючи явні та неявні приведення), делегати, атрибути, події, загальні типи та методи, анонімні функції, винятки та коментарі, що є досить зручним при розробці відповідного програмного забезпечення. В якості середовища розробки для створення програми обрано Microsoft Visual Studio, що є досить ефективним та зручним інструментом для програмування на С#, а також для розробки програмного забезпечення різного призначення.

Програмне забезпечення для розв'язання задачі розпізнавання штрихових кодів забезпечує виконання таких функцій:

- можливість розпізнавання штрихових кодів;
- можливість вибору файлу з графічним зображенням;
- можливість пошуку товару за розпізнаним штриховим кодом;
- можливість змінити зображення штрихового коду;
- можливість декодування штрихових кодів різних форматів.

Для експлуатації програмного забезпечення для розпізнавання штрихових кодів необхідно таке апаратне та програмне забезпечення:

- оперативна пам'ять об'ємом 3Гб і більше;
- процесор із тактовою частотою 1,4 GHz і більше;
- вільне місце на диску 100 Мб або більше;
- наявність .NET Framework;
- встановлена операційна система Windows.

Функціональні характеристики розробленого програмного забезпечення для розпізнавання штрихових кодів наведено у технічному

завданні (додаток А). Фрагмент тексту програмного забезпечення для розпізнавання штрихових кодів наведено у додатку Б.

4.2 Характеристики програми для розпізнавання штрихових кодів

Програмне забезпечення для розпізнавання штрихових кодів може працювати під керуванням операційної системи Windows. Для коректного функціонування програми для розв'язання задачі розпізнавання штрихових кодів необхідно мати комп'ютер з забезпеченням надійного енергоживлення.

4.3 Інструкція по експлуатації програми

4.3.1 Звернення до програми

Для запуску програмного забезпечення для розпізнавання штрихових кодів необхідно запустити головний файл, що знаходиться в кореневій директорії програми.

4.3.2 Вхідні й вихідні дані

Вхідними даними до програмного забезпечення для розпізнавання штрихових кодів є файл з зображенням штрихового коду.

Вихідними даними програмного забезпечення для розпізнавання штрихових кодів є інформація про розпізнаний штриховий код, зокрема, його тип, номер та відповідний товар.

4.3.3 Повідомлення

Користувач програмного забезпечення для розпізнавання штрихових кодів може отримати такі повідомлення: повідомлення про те, що файл не

містить штриховий код, повідомлення про успішне розпізнавання штрихового коду, повідомлення про неможливість розпізнати штриховий код.

4.4 Виконання програмного забезпечення для розпізнавання штрихових кодів

Після запуску програмного забезпечення для розпізнавання штрихових кодів користувачу виводиться головна форма програми (рис. 4.1).

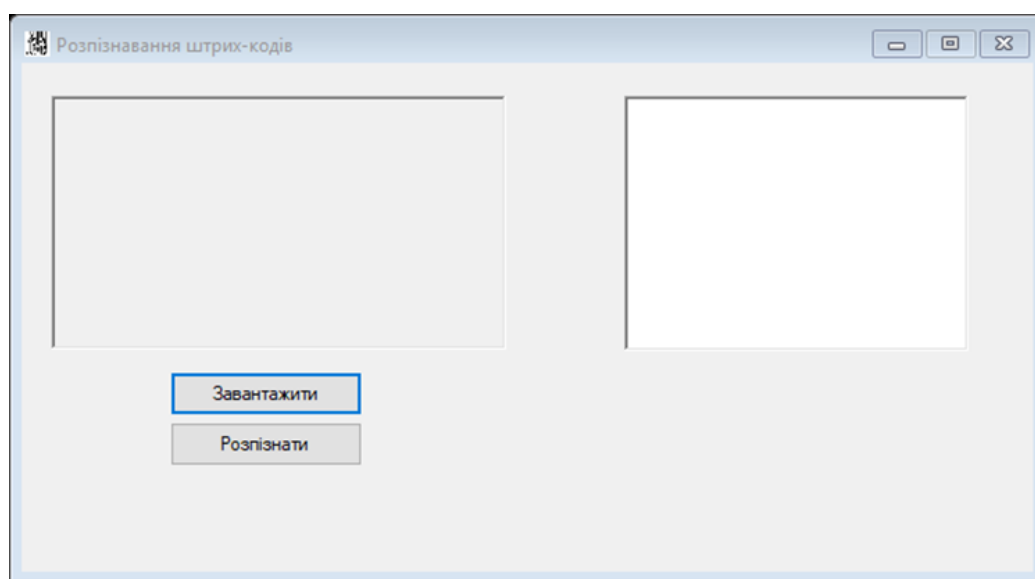


Рисунок 4.1 – Головна форма програми для розпізнавання штрихових кодів

Тут користувач бачить область для відображення штрихового коду та область для відображення результатів розпізнавання. Також користувачу відображаються дві кнопки для завантаження та розпізнавання штрихового коду.

При натисканні на кнопку Завантажити відбувається відкриття діалогового вікна, в якому користувачу надається можливість вибору файлу для розпізнавання штрихового коду (рис. 4.2).

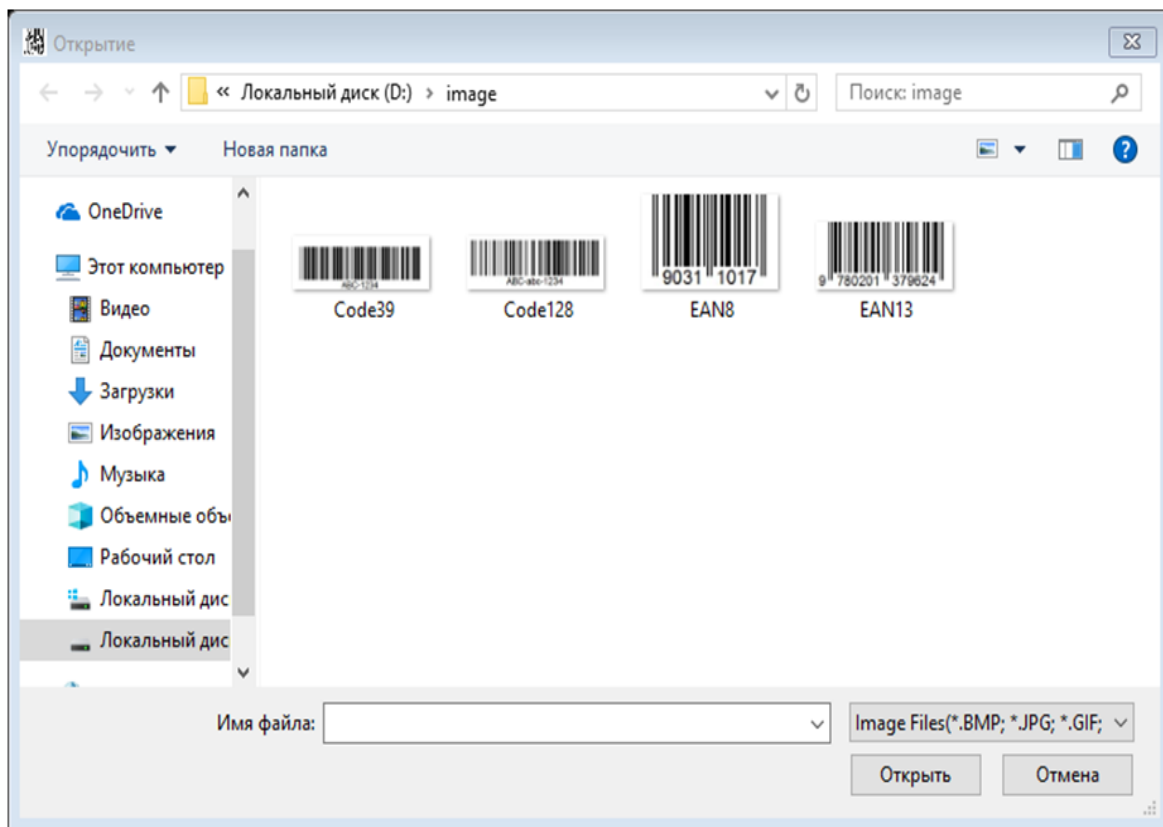


Рисунок 4.2 – Діалогове вікно для вибору файлу з штриховим кодом

Після вибору файлу та натискання на кнопку Розпізнати у головній формі програми у відповідній області (рис. 4.3) виводиться результат розпізнавання (тип штрихового коду, його номер та відповідний товар).

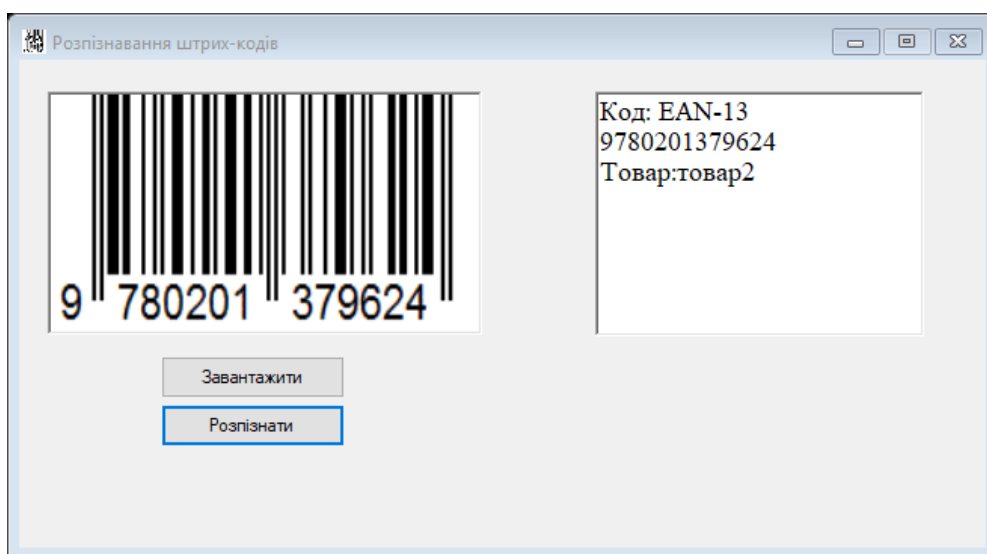


Рисунок 4.3 – Результат розпізнавання штрихового коду

4.5 Тестування програмного забезпечення для розпізнавання штрихових кодів

Виконано тестування програмного забезпечення для розпізнавання штрихових кодів.

У процесі тестування програмного забезпечення для розпізнавання штрихових кодів було перевірено коректність роботи програми. Із запропонованих системі ста фотографій із зображеннями штрихових кодів, всі були правильно розпізнані. Приклад розпізнавання наведено на рис. 4.4.

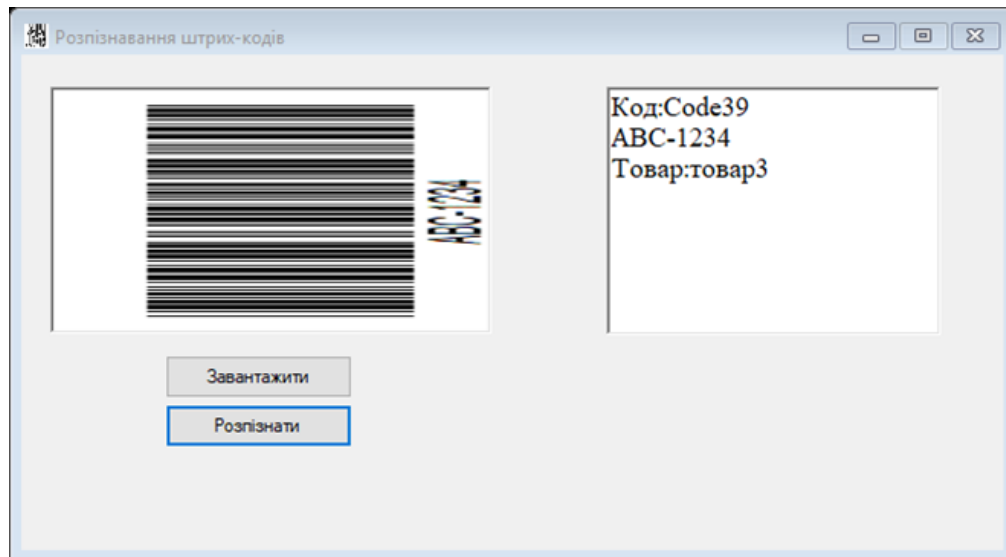


Рисунок 4.4 – Приклад розпізнавання штрихового коду у процесі тестування програми

На роботу програми для розпізнавання штрихових кодів не вплинуло навіть вертикальне розміщення штрих-коду. Наприклад, на рис.4.5 зображено штрих-код без цифр, але програма усе одно правильно розпізнала поданий штриховий код.

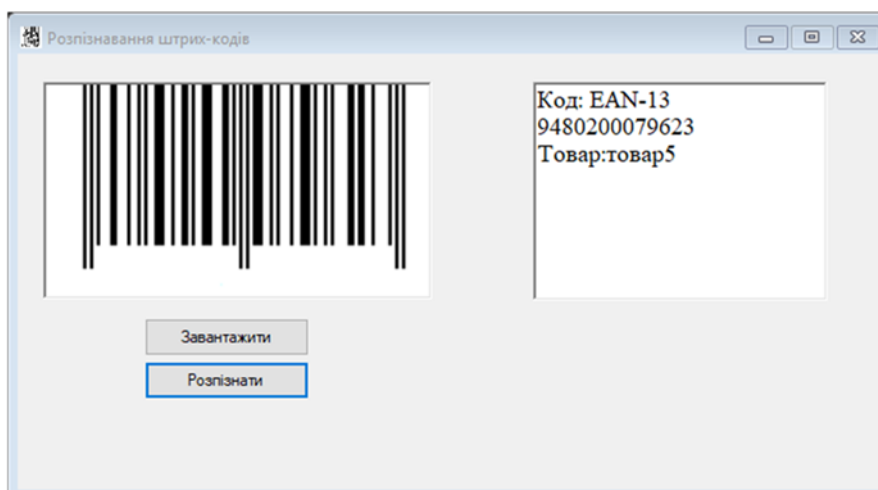


Рисунок 4.5 – Розпізнавання штрихового коду без числових символів

Також були протестовані основні функції системи шляхом створення модульних тестів.

Результати проведеного тестування зображено на рис.4.6.

Обозреватель тестов		
Поиск		
Выполнить все Выполнить... Список воспроизведения: Все тесты		
Barcode recognition (тесты: 6)		
✔ UnitTestBarcodeRecognition (6)		740 мс
✔ Barcode_recognition.Tests (5)		733 мс
✔ BarcodeCodeTests (1)		331 мс
✔ barcodeCodeTest		331 мс
✔ BarcodeEANTests (2)		54 мс
✔ barcodeEAN13Test		37 мс
✔ barcodeEAN8Test		17 мс
✔ InformBarcodeTests (1)		135 мс
✔ numberblackTest		135 мс
✔ InformProductTests (1)		210 мс
✔ searchProductTest		210 мс

Рисунок 4.6 – Результати модульного тестування

У процесі тестування було перевірено точність роботи функції для розпізнавання штрих-кодів типу EAN. Із запропонованих системі 20 фотографій, 18 було правильно розпізнано, а дві було неправильно розпізнано. Причиною неправильного розпізнавання штрих-кодів виявилось неякісне зображення, втрата різкості, каламутність зображення – особливо в дрібних

деталях, невеликий розмір зображення. Отже відносна похибка становить 0,1(10%). Результати тестування роботи функції зображено на рис.4.7.

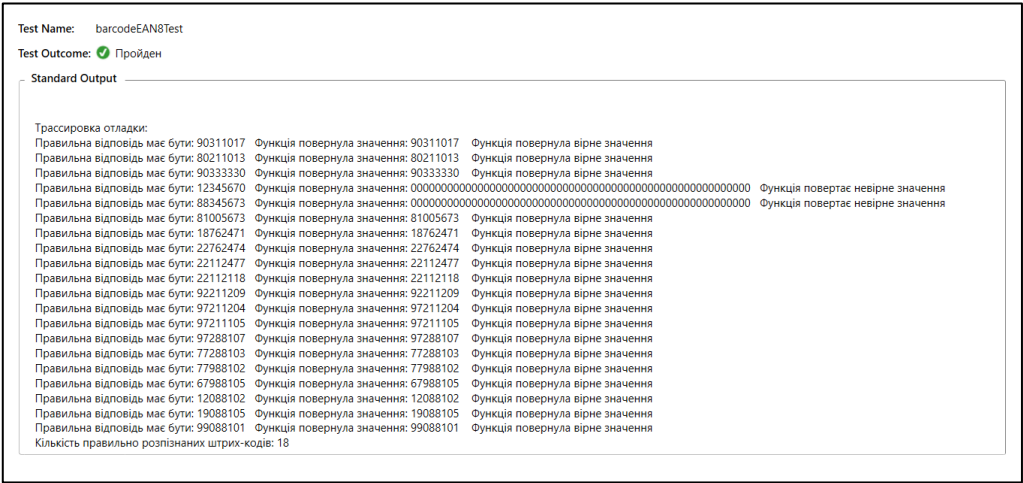


Рисунок 4.7 – Результати модульного тестування функції barcodeEAN

У процесі тестування було перевірено точність роботи функції barcodeEAN13 для розпізнавання штрих-кодів типу EAN13. Із запропонованих системі 20 фотографій, 16 було правильно розпізнано, а 4 було неправильно розпізнано. Отже відносна похибка становить 0,2(20%). Результати тестування роботи функції зображено на рис.4.8.

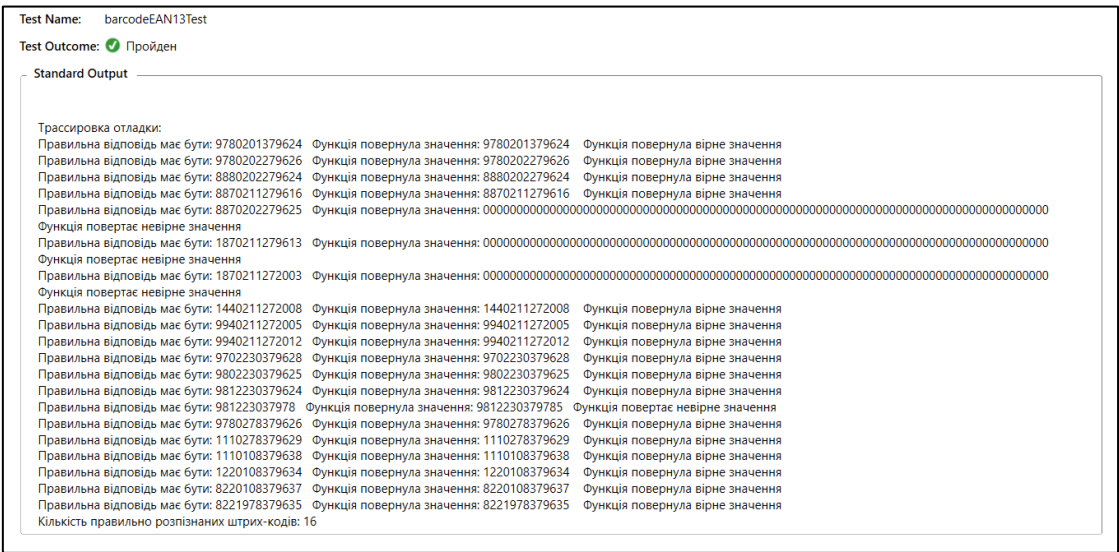


Рисунок 4.8 – Результати модульного тестування функції barcodeEAN13

У процесі тестування було перевірено точність роботи функції BarcodeCode для розпізнавання штрих-кодів типу Code39 і Code128. Із запропонованих системі 20 фотографій, 17 було правильно розпізнано, а 3 було неправильно розпізнано. Причиною неправильного розпізнавання штрих-кодів виявилось неякісне зображення, втрата різкості, каламутність зображення – особливо в дрібних деталях, невеликий розмір зображення. Отже відносна похибка становить 0,15(15%). Результати тестування роботи функції зображено на рис.4.9.

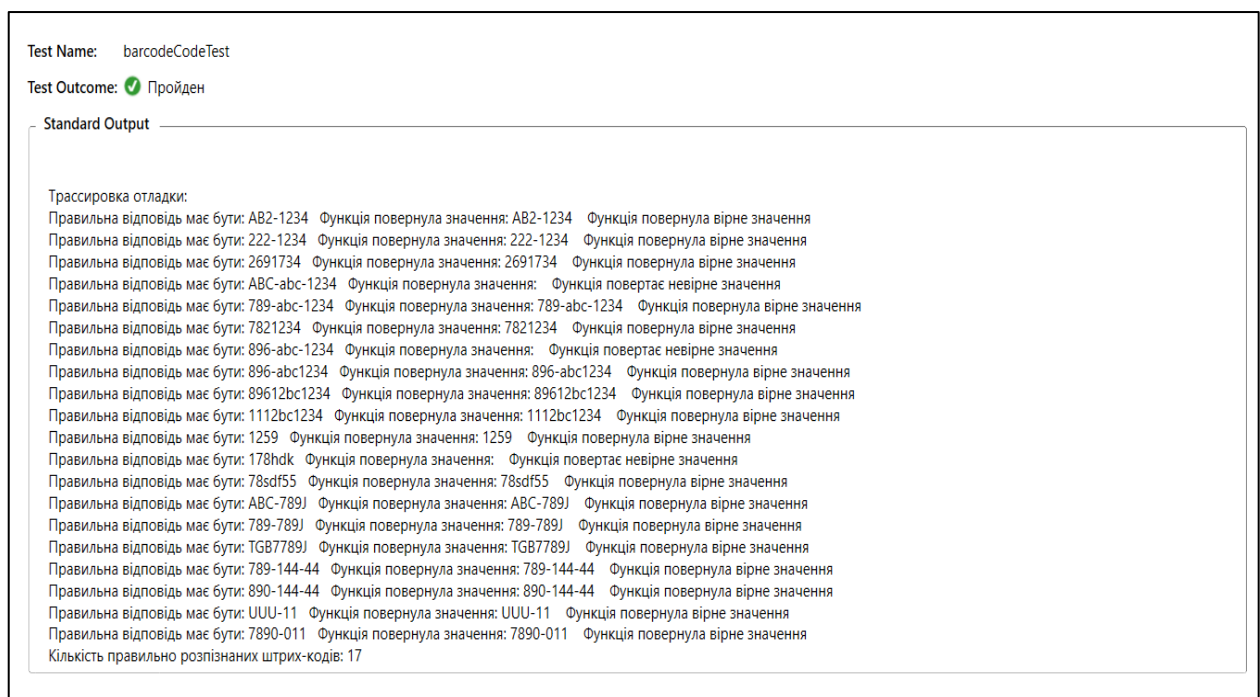


Рисунок 4.9 – Результати модульного тестування функції BarcodeCode

Таким чином, за результатами тестування виявлено, що програма для розпізнавання штрихових кодів функціонує коректно та відповідає технічному завданню. Програма для розпізнавання штрихових кодів реалізує всі функціональні вимоги та успішно виконує свою основну задачу – розв’язання задачі розпізнавання штрихових кодів.

Розроблене програмне забезпечення дозволяє розв’язувати завдання розпізнавання штрихових кодів.

4.6 Висновки за розділом 4

Описано програмне забезпечення для розпізнавання штрихових кодів. Виконано тестування розробленого програмного забезпечення для розпізнавання штрихових кодів. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє розв'язувати задачу розпізнавання штрихових кодів.

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи магістра було проаналізовано та досліджено процес розробки програмного забезпечення для розпізнавання образів.

Визначено, що штрихові коди є числовими та лінійними символами, які знаходяться на товарах та допомагають в їх ідентифікації. Показано, що програмні системи оброблення штрих-кодів допомагають підприємствам в відстеженні товарів, управлінні цінами та запасами. Проаналізовано різні види штрихових кодів. Відзначено, що штрихові коди легко можна впровадити в бізнес, використовуючи відповідне програмне забезпечення. Проаналізовано переваги використання програмного забезпечення оброблення штрихових кодів для бізнесу. Відзначено, що використання штрихових кодів у поєднанні з програмними системами дозволяє компаніям завжди мати точну та повну інформацію про поточні запаси та їх рух. Відзначено, що розпізнавання штрихових кодів відноситься до завдання розпізнавання образів. Проаналізовано компоненти програмних засобів розпізнавання штрихових кодів, а також інтелектуальні методи розпізнавання штрихових кодів.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для розпізнавання штрихових кодів. Проте деякі програми для розпізнавання штрихових кодів можуть бути обмеженими у підтримці різних форматів штрихових кодів, що може викликати проблеми при розпізнаванні певних типів кодів. При використанні інших програмних засобів, особливо при обробці великих обсягів даних або на повільних пристроях, час, необхідний для розпізнавання, може бути значним. Крім того, багато програмних засобів для розпізнавання штрихових кодів вимагають чіткості та яскравості штрихового коду, внаслідок чого низька якість друку, пошкодження або розмивання штрихового коду можуть призвести до неправильного або некоректного розпізнавання. Також освітлення та кути зору можуть впливати на точність розпізнавання,

відповідно, недостатнє або неправильне освітлення може призвести до помилок у розпізнаванні. Деякі програмні засоби для розпізнавання штрихових кодів можуть вимагати дорогого обладнання або платних програмних рішень, що в деяких випадках обмежує їх застосування на практиці, особливо в невеликих організаціях. Тому актуальною є розробка програмного забезпечення для розпізнавання штрихових кодів.

Сформульовано функціональні вимоги до програмного забезпечення для розпізнавання штрихових кодів.

Для реалізації програмного забезпечення для розпізнавання штрихових кодів обрано мову програмування C#, яка є мовою програмування, що підтримує статичну типізацію, поліморфізм, перевантаження операторів (включаючи явні та неявні приведення), делегати, атрибути, події, загальні типи та методи, анонімні функції, винятки та коментарі, що є досить зручним при розробці відповідного програмного забезпечення.

Для створення програмного забезпечення для розпізнавання штрихових кодів обрано середовище розробки Microsoft Visual Studio, що є досить ефективним та зручним інструментом для програмування на C#, а також для розробки програмного забезпечення різного призначення.

Новизна роботи полягає в тому, що створено метод розпізнавання штрихових кодів на основі інтелектуальних обчислень, що складається з етапів зчитування зображення, визначення місця розташування штрих-коду на об'єкті, виділення складових частин штрих-коду, розшифрування інформації, закодованої у штриховому коді, виведення результату.

Практичне значення роботи полягає в тому, що розроблено програмне забезпечення для розпізнавання штрихових кодів.

Запропоновано структуру та описано основні модулі програмного забезпечення для розпізнавання штрихових кодів.

Описано функціонування програмного забезпечення для розпізнавання штрихових кодів. Описано особливості реалізації програмного забезпечення для розпізнавання штрихових кодів. Виконано проєктування інтерфейсу

взаємодії користувача з програмним забезпеченням для розпізнавання штрихових кодів.

Виконано тестування розробленого програмного забезпечення для розпізнавання штрихових кодів. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє розв'язувати задачу розпізнавання штрихових кодів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Barcode: The Ultimate Guide to Barcodes [Electronic resource]. – Access mode : <https://www.wasppbarcode.com/buzz/barcode>.
2. GS1 barcodes [Electronic resource]. – Access mode : <https://www.gs1.org/standards/barcodes>.
3. Barcodes Defined. How They Work, Benefits & Uses [Electronic resource]. – Access mode : <https://www.netsuite.com/portal/resource/articles/inventory-management/barcode.shtml>.
4. What Is a Barcode? Definition and Guide [Electronic resource]. – Access mode : <https://www.shopify.com/blog/what-is-a-barcode>.
5. How do Barcodes Work [Electronic resource]. – Access mode : <https://www.lightspeedhq.com/blog/how-do-barcodes-work/>.
6. What is a barcode [Electronic resource]. – Access mode : <https://www.teklynx.com/en/blog/what-is-a-barcode>.
7. Barcode: What It Means, Benefits, History [Electronic resource]. – Access mode : <https://www.investopedia.com/terms/b/barcode.asp>.
8. How to Identify a Barcode [Electronic resource]. – Access mode : <https://www.barcodefaq.com/barcode-match/>.
9. Barcode Recognition [Electronic resource]. – Access mode: https://docs.opencv.org/4.x/d6/d25/tutorial_barcode_detect_and_decode.html.
10. Deep Learning in Barcode Recognition: A Systematic Literature Review [Electronic resource]. – Access mode : <https://ieeexplore.ieee.org/document/9681298>.
11. Best Barcode Software [Electronic resource]. – Access mode : <https://www.g2.com/categories/barcode>.
12. Barcoding Software [Electronic resource]. – Access mode : <https://www.capterra.com/barcoding-software/>.
13. Barcode Software [Electronic resource]. – Access mode : <https://www.trustradius.com/barcode>.

14. QR Code Desktop Reader [Electronic resource]. – Access mode : <https://www.codetwo.com/freeware/qr-code-desktop-reader/>.

15. BarCode Descriptor [Electronic resource]. – Access mode : <https://www.softpedia.com/get/System/System-Miscellaneous/BarCode-Descriptor.shtml>.

16. TNR CodeExpert [Electronic resource]. – Access mode : <https://wikiprograms.org/tnr-codeexpert/>.

17. C# The modern, innovative, open-source programming language for building all your apps [Electronic resource]. – Access mode : <https://dotnet.microsoft.com/en-us/languages/csharp>.

18. C Sharp [Electronic resource]. – Access mode : https://vi.wikipedia.org/wiki/C_Sharp.

19. GitHub Copilot and Visual Studio 2022 [Electronic resource]. – Access mode : <https://visualstudio.microsoft.com/>.

20. What is Visual Studio [Electronic resource]. – Access mode : <https://blog.hubspot.com/website/what-is-visual-studio>.

21. Chapter 3. Barcode scanning algorithm: localization and decode [Electronic resource]. – Access mode : <https://www.dynamsoft.com/blog/insights/best-practices-for-maximizing-barcode-reader-technology-3-6/>.

22. A Bayesian Algorithm for Reading 1D Barcodes [Electronic resource]. – Access mode : <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2859730/>.

23. Deep Learning in Barcode Recognition: A Systematic Literature Review [Electronic resource]. – Access mode : <https://ieeexplore.ieee.org/document/9681298>.

ДОДАТОК А
Технічне завдання

Вступ

Програмне забезпечення може використовуватися для розв’язання задачі розпізнавання штрихових кодів.

А.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Дослідження та програмна реалізація засобів розпізнавання штрихових кодів», затверджене наказом Національного університету «Запорізька політехніка» № 437 від 14 листопада 2023 р.

А.2 Призначення розробки

Програмний продукт призначений для розв’язання задачі розпізнавання штрихових кодів.

А.3 Основні вимоги до програми, що розробляється

А.3.1 Вимоги до функціональних характеристик

Програмне забезпечення для розв’язання задачі розпізнавання штрихових кодів забезпечує виконання таких функцій:

- можливість розпізнавання штрихових кодів;
- можливість вибору файлу з графічним зображенням;
- можливість пошуку товару за розпізнаним штриховим кодом;
- можливість змінити зображення штрихового коду;
- можливість декодування штрихових кодів різних форматів.

А.3.2 Вимоги до інтерфейсу програми

Інтерфейс програмного забезпечення для розпізнавання штрихових кодів повинен бути зручним для користувачів. Повинна бути забезпечена можливість роботи як у візуальному режимі.

А.3.3 Вимоги до надійності

Програмне забезпечення для розпізнавання штрихових кодів повинно забезпечити надійне функціонування.

А.3.4 Умови експлуатації

Для експлуатації програмного забезпечення для розв'язання задачі розпізнавання штрихових кодів необхідна наявність персонального комп'ютера у користувача.

А.3.5 Вимоги до складу та параметрів технічний засобів

Для експлуатації програмного забезпечення для розв'язання задачі розпізнавання штрихових кодів необхідно таке апаратне та програмне забезпечення:

- оперативна пам'ять об'ємом 3Гб і більше;
- процесор із тактовою частотою 1,4 GHz і більше;
- вільне місце на диску 100 Мб або більше;
- наявність .NET Framework;
- встановлена операційна система Windows.

A.3.6 Вимоги до маркування і пакування

Програма для розв’язання задачі розпізнавання штрихових кодів може бути записана на будь-якому носії інформації.

На пакуванні повинна бути назва програми – «Програмне забезпечення для розпізнавання штрихових кодів».

A.4 Вхідні дані до роботи

Вхідними даними до програмного забезпечення для розпізнавання штрихових кодів є файл з зображенням штрихового коду.

Вихідними даними програмного забезпечення для розпізнавання штрихових кодів є інформація про розпізнаний штриховий код, зокрема, його тип, номер та відповідний товар.

ДОДАТОК Б
Фрагмент тексту програми

```

using System;
namespace Brcd_rcgnt
{
    public partial class BrcdRcgnt : Form
    {
        Bitmap img;
        int nmbrbk = 0;
        BrcdEAN BrcdEAN = new BrcdEAN();
        BrcdKod BrcdKod = new BrcdKod();
        InformBrcd Brcd = new InformBrcd();
        bool button = false;
        public BrcdRcgnt()
        {
            InitializeComponent();
        }
        private void BrcdRcgnt_Load(object sender, EventArgs e)
        {
            pctrbx1.SizeMode = PictureBoxSizeMode.AutoSize;
            pctrbx1.BorderStyle = BorderStyle.Fixed3D;
        }
        private void rcgn_Click(object sender, EventArgs e)
        {
            nmbrbk = 0;
            rchtbx1.Text = "";
            if (button==true)
            {
                Color[,] Vsi = new Color[img.Width - 1, img.Height - 1];
                for (int i = 0; i < (img.Width - 1); i++)
                {
                    for (int j = 0; j < (img.Height - 1); j++)
                    {
                        Vsi[i, j] = img.GetPixel(i, j);
                    }
                }
                nmbrbk= Brcd.nmbrbk(img, Vsi);
                switch (nmbrbk)
                {
                    case 30:
                    {
                        rchtbx1.Text += "Код: EAN-13" + "\n";
                        string[] rslt = BrcdEAN.BrcdEAN13(img, Vsi);
                        if (rslt.Length != 0)
                        {
                            for (int i = 0; i < 13; i++)
                                rchtbx1.Text += rslt[i];
                            rchtbx1.Text += "\nТовар:" +
Brcd.srchprdct(String.Concat<string>(rslt));
                        }
                        else rchtbx1.Text = "Штрих-код не розпізнано!";
                    }
                    break;
                    case 22:
                    {
                        rchtbx1.Text += "Код: EAN-8" + "\n";
                        string[] rslt = BrcdEAN.BrcdEAN8(img, Vsi);

```

```

        if (rslt.Length != 0)
        {
            for (int i = 0; i < 8; i++)
                rchtbx1.Text += rslt[i];
            rchtbx1.Text += "\nТовар:" +
Brcd.srchprdct(String.Concat<string>(rslt));
        }
        else rchtbx1.Text = "Штрих-код не розпізнано!";
    }
    break;
default:
    {
        rchtbx1.Text += "Код:Kod39" + "\n";
        ArrayList Brcd = new ArrayList();
        string kod = "";
        BrcdKod.BrcdKod(ref Brcd, img);
        if (Brcd.Count != 0)
        {
            for (int i = 0; i < Brcd.Count; i++)
            {
                rchtbx1.Text += Brcd[i];
                kod += Brcd[i];
            }
            rchtbx1.Text += "\nТовар:" +
this.Brcd.srchprdct(kod);
        }
        else rchtbx1.Text = "Штрих-код не розпізнано!";
    }
    break;
}
}
else MessageBox.Show("Завантажте зображення!", "Помилка",
MessageBoxButtons.OK, MessageBoxIcon.Error);
}
private void download_Click(object sender, EventArgs e)
{
    rchtbx1.Text = "";
    OpenFileDialog opdg = new OpenFileDialog();
    opdg.Filter = "Img Files(*.BMP; *.JPG; *.GIF; *.PNG)| *.BMP;
*.JPG; *.GIF; *.PNG | Vsi files(*.*) | *.* ";
    if (opdg.ShowDialog() == DialogResult.OK)
    {
        try
        {
            img = new Bitmap(opdg.FileName);
            pctrbx1.SizeMode = PictureBoxSizeMode.StretchImg;
            pctrbx1.Image = img;
            pctrbx1.Invalidate();
            button = true;
        }
        catch
        {
            DialogResult rezult = MessageBox.Show("Не відкрився файл!",
"Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}

```

```

    }
    }
}

namespace Brcd_rcgnt
{
    public class WrkBD
    { public SQLiteConnection BD = new SQLiteConnection();
      protected WrkBD()
      {
          BD.ConnectionString = "Data Source=D:\\DataBase.BD";
      }
      public void openBD()
      {
          BD.Open();
      }
      protected void closeBD()
      {
          BD.Close();
      }
    }
    public class InformBrcd:WrkBD
    { public int nmbrbk (Bitmap img, Color[,] Vsi)
      {
          int nmbrbk = 0;
          for (int i = 0; i < (img.Width - 1); i++)
          {
              if (Vsi[i, 0].ToArgb() == Color.Black.ToArgb())
              {
                  for (int j = i; j < (img.Width - 1); j++)
                  {
                      if (Vsi[j, 0].ToArgb() == Color.Black.ToArgb())
                      {
                          {
                              nmbrbk++;
                              i = j;
                              break;
                          }
                      }
                  }
              }
          }
          return nmbrbk;
      }
      public string srchprdct(string Brcd)
      {
          string rslt="";
          openBD();
          SQLiteCommand cmand = new SQLiteCommand("SELECT nm FROM Prdct
WHERE Brcd='" + Brcd + "'", BD);

```

```

        SQLiteDataChytach chytach = command.ExecuteChytach();
        chytach.Read();
        if (chytach.HasRows)
        {
            rslt = chytach.GetString(0);
        }
        chytach.Close();
        closeBD();
        return rslt;
    }
}

public class BrcdEAN
{
    int rah = 0;
    string GL = "";
    public string[] BrcdEAN8(Bitmap img, Color[,] Vsi)
    {
        string[] BrcdEAN8 = new string[8];
        rah = 0;
        int[] wtlns = new int[22];
        GL = "";
        for (int j = 0; j < 22; j++)
        {
            for (int i = rah; i < (img.Width - 1); i++)
            {
                if (Vsi[i, 0].ToArgb() == Color.Black.ToArgb())
                {
                    wtlns[j]++;
                    if (Vsi[i + 1, 0].ToArgb() != Color.Black.ToArgb())
                    {
                        rah = i + 1;
                        break;
                    }
                }
            }
        }
        int mnln = wtlns.Min();
        int[] lnclr = new int[67];
        int nmbr = 0;
        for (int i = 0; i < (img.Width - 1); i++)
        {
            if (Vsi[i, 0].ToArgb() == Color.Black.ToArgb())
            {
                for (int j = i; j < i + 67 * mnln; j = j + mnln)
                {
                    if (Vsi[j, 0].ToArgb() == Color.Black.ToArgb())
                    {
                        lnclr[nmbr] = 1;
                    }
                    if (Vsi[j, 0].ToArgb() == Color.White.ToArgb())
                    {
                        lnclr[nmbr] = 0;
                    }
                }
            }
        }
    }
}

```

```

        }
        nmbr++;
    }
    break;
}
}
int nmbr_2 = 0;
for (int i = 3; i < 64; i = i + 7)
{
    if (i == 31)
    {
        i = i + 5;
    }
    for (int j = 0; j < 7; j++)
    {
        BrCdEAN8[nmbr_2] = BrCdEAN8[nmbr_2] +
Convert.ToString(lnclr[i]);
        i++;
    }
    i = i - 7;
    nmbr_2++;
}
for (int i = 0; i < 8; i++)
{
    switch (BrCdEAN8[i])
    {
        case "0001101":
        {
            BrCdEAN8[i] = "0";
            GL = GL + "L";
        }
        break;
        case "0011001":
        {
            BrCdEAN8[i] = "1";
            GL = GL + "L";
        }
        break;
        case "0010011":
        {
            BrCdEAN8[i] = "2";
            GL = GL + "L";
        }
        break;
        case "0111101":
        {
            BrCdEAN8[i] = "3";
            GL = GL + "L";
        }
        break;
        case "0100011":
        {

```

```

        BrcdEAN8[i] = "4";
        GL = GL + "L";
    }
    break;
case "0110001":
    {
        BrcdEAN8[i] = "5";
        GL = GL + "L";
    }
    break;
case "0101111":
    {
        BrcdEAN8[i] = "6";
        GL = GL + "L";
    }
    break;
case "0111011":
    {
        BrcdEAN8[i] = "7";
        GL = GL + "L";
    }
    break;
case "0110111":
    {
        BrcdEAN8[i] = "8";
        GL = GL + "L";
    }
    break;
case "0001011":
    {
        BrcdEAN8[i] = "9";
        GL = GL + "L";
    }
    break;
case "1110010":
    {
        BrcdEAN8[i] = "0";
    }
    break;
case "1100110":
    {
        BrcdEAN8[i] = "1";
    }
    break;
case "1101100":
    {
        BrcdEAN8[i] = "2";
    }
    break;
case "1000010":
    {
        BrcdEAN8[i] = "3";
    }

```

```

    }
    break;
case "1011100":
    {
        BrcdEAN8[i] = "4";
    }
    break;
case "1001110":
    {
        BrcdEAN8[i] = "5";
    }
    break;
case "1010000":
    {
        BrcdEAN8[i] = "6";
    }
    break;
case "1000100":
    {
        BrcdEAN8[i] = "7";
    }
    break;
case "1001000":
    {
        BrcdEAN8[i] = "8";
    }
    break;
case "1110100":
    {
        BrcdEAN8[i] = "9";
    }
    break;
case "0100111":
    {
        BrcdEAN8[i] = "0";
        GL += "G";
    }
    break;
case "0110011":
    {
        BrcdEAN8[i] = "1";
        GL += "G";
    }
    break;
case "0011011":
    {
        BrcdEAN8[i] = "2";
        GL += "G";
    }
    break;
case "0100001":
    {

```

```

        BrcdEAN8[i] = "3";
        GL += "G";
    }
    break;
case "0011101":
    {
        BrcdEAN8[i] = "4";
        GL += "G";
    }
    break;
case "0111001":
    {
        BrcdEAN8[i] = "5";
        GL += "G";
    }
    break;
case "0000101":
    {
        BrcdEAN8[i] = "6";
        GL += "G";
    }
    break;
case "0010001":
    {
        BrcdEAN8[i] = "7";
        GL += "G";
    }
    break;
case "0001001":
    {
        BrcdEAN8[i] = "8";
        GL += "G";
    }
    break;
case "0010111":
    {
        BrcdEAN8[i] = "9";
        GL += "G";
    }
    break;
    }
}
return BrcdEAN8;
}
public string[] BrcdEAN13(Bitmap img, Color[,] Vsi)
{
    string[] BrcdEAN13 = new string[12];

    GL = "";
    rah = 0;
    int[] wtlns = new int[30];
    for (int j = 0; j < 30; j++)

```

```

{
    for (int i = rah; i < (img.Width - 1); i++)
    {
        if (Vsi[i, 0].ToArgb() == Color.Black.ToArgb())
        {
            wtlns[j]++;
            if (Vsi[i + 1, 0].ToArgb() != Color.Black.ToArgb())
            {
                rah = i + 1;
                break;
            }
        }
    }
}
int mnln = wtlns.Min();
int[] lnclr = new int[95];
int nmbr = 0;
for (int i = 0; i < (img.Width - 1); i++)
{
    if (Vsi[i, 0].ToArgb() == Color.Black.ToArgb())
    {
        for (int j = i; j < i + 95 * mnln; j = j + mnln)
        {
            if (Vsi[j, 0].ToArgb() == Color.Black.ToArgb())
            {
                lnclr[nmbr] = 1;
            }
            if (Vsi[j, 0].ToArgb() == Color.White.ToArgb())
            {
                lnclr[nmbr] = 0;
            }
            nmbr++;
        }
        break;
    }
}
int nmbr_2 = 0;
for (int i = 3; i < 92; i = i + 7)
{
    if (i == 45)
    {
        i = i + 5;
    }
    for (int j = 0; j < 7; j++)
    {
        BrCdEAN13[nmbr_2] = BrCdEAN13[nmbr_2] +
Convert.ToString(lnclr[i]);
        i++;
    }
    i = i - 7;
    nmbr_2++;
}

```

```

for (int i = 0; i < 12; i++)
{
    switch (BrCdEAN13[i])
    {
        case "0001101":
        {
            BrCdEAN13[i] = "0";
            GL = GL + "L";
        }
        break;
        case "0011001":
        {
            BrCdEAN13[i] = "1";
            GL = GL + "L";
        }
        break;
        case "0010011":
        {
            BrCdEAN13[i] = "2";
            GL = GL + "L";
        }
        break;
        case "0111101":
        {
            BrCdEAN13[i] = "3";
            GL = GL + "L";
        }
        break;
        case "0100011":
        {
            BrCdEAN13[i] = "4";
            GL = GL + "L";
        }
        break;
        case "0110001":
        {
            BrCdEAN13[i] = "5";
            GL = GL + "L";
        }
        break;
        case "0101111":
        {
            BrCdEAN13[i] = "6";
            GL = GL + "L";
        }
        break;
        case "0111011":
        {
            BrCdEAN13[i] = "7";
            GL = GL + "L";
        }
        break;
    }
}

```

```

case "0110111":
{
    BrcdEAN13[i] = "8";
    GL = GL + "L";
}
break;
case "0001011":
{
    BrcdEAN13[i] = "9";
    GL = GL + "L";
}
break;
case "1110010":
{
    BrcdEAN13[i] = "0";
}
break;
case "1100110":
{
    BrcdEAN13[i] = "1";
}
break;
case "1101100":
{
    BrcdEAN13[i] = "2";
}
break;
case "1000010":
{
    BrcdEAN13[i] = "3";
}
break;
case "1011100":
{
    BrcdEAN13[i] = "4";
}
break;
case "1001110":
{
    BrcdEAN13[i] = "5";
}
break;
case "1010000":
{
    BrcdEAN13[i] = "6";
}
break;
case "1000100":
{
    BrcdEAN13[i] = "7";
}
break;

```

```

case "1001000":
{
    BrcdEAN13[i] = "8";
}
break;
case "1110100":
{
    BrcdEAN13[i] = "9";
}
break;
case "0100111":
{
    BrcdEAN13[i] = "0";
    GL += "G";
}
break;
case "0110011":
{
    BrcdEAN13[i] = "1";
    GL += "G";
}
break;
case "0011011":
{
    BrcdEAN13[i] = "2";
    GL += "G";
}
break;
case "0100001":
{
    BrcdEAN13[i] = "3";
    GL += "G";
}
break;
case "0011101":
{
    BrcdEAN13[i] = "4";
    GL += "G";
}
break;
case "0111001":
{
    BrcdEAN13[i] = "5";
    GL += "G";
}
break;
case "0000101":
{
    BrcdEAN13[i] = "6";
    GL += "G";
}
break;

```

```

        case "0010001":
        {
            BrCdEAN13[i] = "7";
            GL += "G";
        }
        break;
        case "0001001":
        {
            BrCdEAN13[i] = "8";
            GL += "G";
        }
        break;
        case "0010111":
        {
            BrCdEAN13[i] = "9";
            GL += "G";
        }
        break;
    }
}
int firstNmbr = 0;
if (GL == "LLLLLL")
{
    firstNmbr = 0;
}
if (GL == "LGLGLG")
{
    firstNmbr = 1;
}
if (GL == "LGLGGL")
{
    firstNmbr = 3;
}
if (GL == "GLLGLG")
{
    firstNmbr = 4;
}
if (GL == "GLGLGL")
{
    firstNmbr = 5;
}
if (GL == "GLGGLL")
{
    firstNmbr = 6;
}
if (GL == "GLGLGL")
{
    firstNmbr = 7;
}
if (GL == "GLGLGL")
{
    firstNmbr = 8;
}

```

```

    }
    if (GL == "GLGGLL")
    {
        firstNmbr = 9;
    }
    string[] rslt = new string[13];
    rslt[0]= Convert.ToString(firstNmbr); ;
    for (int j = 1; j < 13; j++)
    {
        rslt[j] += BrcdEAN13[j-1];
    }
    return rslt;
}
}

```

```

namespace Brcd_rcgnt.Tests
{
    [TestClass()]
    public class BrcdKodTests
    {
        [TestMethod()]
        public void BrcdKodTest()
        {
            Bitmap img = new Bitmap("D:\\img\\Kod39.gif");
            ArrayList Brcd = new ArrayList();
            string kod = "";
            string rslt = "ABC-1234";
            BrcdKod.BrcdKod(ref Brcd, img);
            for (int i = 0; i < Brcd.Count; i++)
            {
                kod += Brcd[i];
            }
            Assert.AreEqual(rslt, kod);
        }
    }
}

```

ДОДАТОК В
Слайди презентації

Міністерство освіти і науки України
Національний університет «Запорізька політехніка»
Кафедра програмних засобів

Дослідження та програмна реалізація засобів розпізнавання штрихових кодів

ст. групи КНТ-112м
д.т.н., професор

Коваль Максим Олександрович
Олійник Андрій Олександрович

Рисунок В.1 – Слайд 1

Об'єкт, предмет та мета роботи

Об'єкт дослідження – процес розробки програмного забезпечення для розпізнавання образів.

Предмет дослідження – методи та програмні засоби для розв'язання завдань розпізнавання штрихових кодів.

Метою роботи є дослідження та розробка програмного забезпечення для розпізнавання штрихових кодів.

2

Рисунок В.2 – Слайд 2

Завдання роботи

Для досягнення поставленої мети у кваліфікаційній роботі бакалавра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів для розпізнавання штрихових кодів;
- розробити метод розпізнавання штрихових кодів на основі інтелектуальних обчислень;
- здійснити проєктування програмного забезпечення для розпізнавання штрихових кодів;
- створити програмне забезпечення для розпізнавання штрихових кодів;
- виконати тестування розробленого програмного забезпечення для розпізнавання штрихових кодів.

3

Рисунок В.3 – Слайд 3

Порівняння існуючих аналогів

Критерій порівняння	QR Code Desktop Reader	BarCode Descriptor	TNR CodeExpert
Зручність роботи для розпізнавання штрихових кодів	+-	+-	+
Підтримка великої кількості форматів файлів з зображеннями	+	+-	-
Безкоштовність	+	+	-
Підтримка багатьох мов	-	-	+
Інтеграція з іншими програмними засобами обробки зображень	-	-	+

4

Рисунок В.4 – Слайд 4

Обґрунтування вибору мови програмування

Критерій порівняння мов програмування	Мова програмування		
	C#	Python	C++
Зручність розробки програмного забезпечення для розпізнавання штрихових кодів	+	–	–
Робота з графічною інформацією	+	+	+-
Засоби спадкування	+	+-	+
Засоби створення інтерфейсу користувача	+	+-	+-
Наявність бібліотек для розпізнавання графічних образів	+	+	+-

5

Рисунок В.5 – Слайд 5

Вибір середовища розробки

Критерій порівняння середовищ розробки	Середовище розробки		
	Visual Studio	MonoDevelop	SharpDevelop
Зручність розробки на мові C#	+	+-	+-
Засоби для аналізу коду	+	+	+-
Велика спільнота розробників	+	+	–
Підтримка різних платформ	+	+	–
Розширені можливості для відладки	+	–	–

6

Рисунок В.6 – Слайд 6

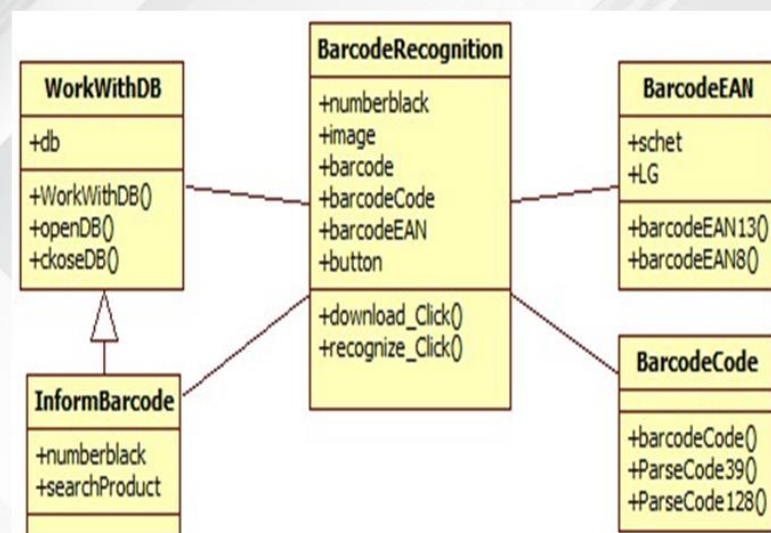
Метод розпізнавання штрихових кодів



7

Рисунок В.7 – Слайд 7

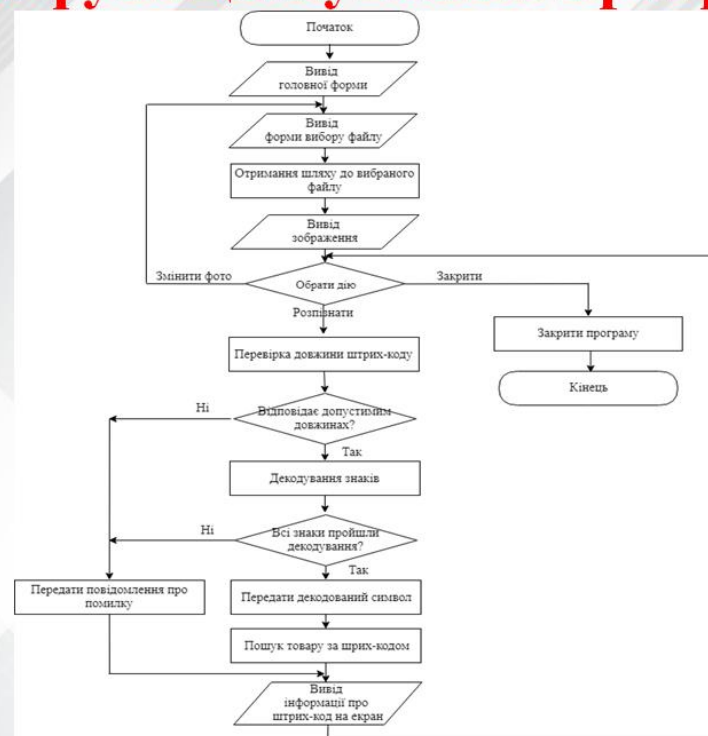
Діаграма класів програмного забезпечення



8

Рисунок В.8 – Слайд 8

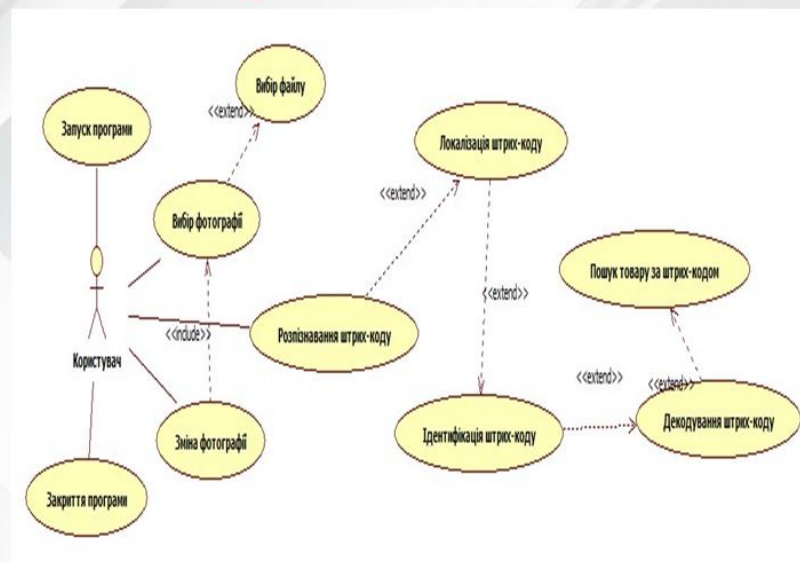
Схема функціонування програми



9

Рисунок В.9 – Слайд 9

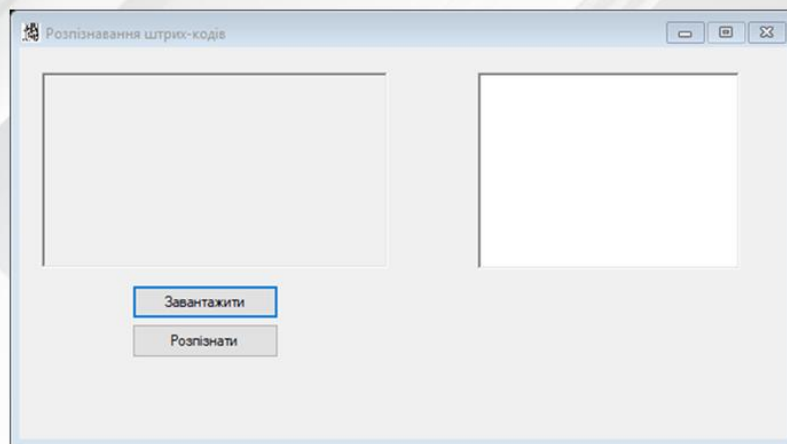
Логіка взаємодії – діаграма прецедентів



10

Рисунок В.10 – Слайд 10

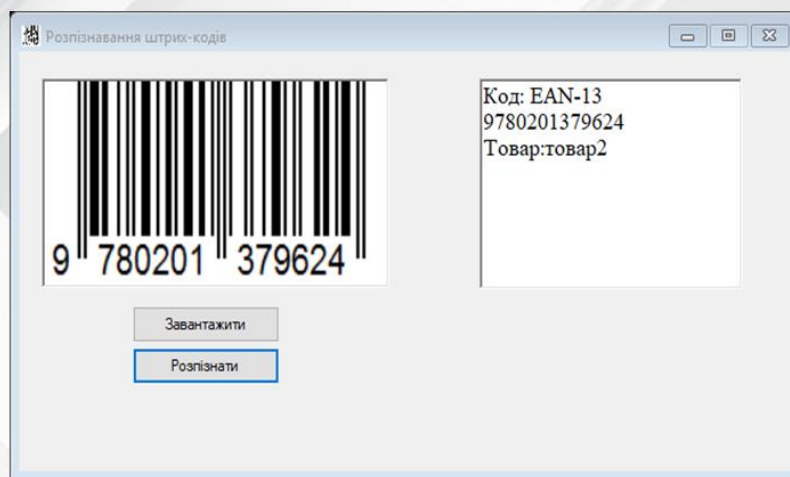
Головна форма програми



11

Рисунок В.11 – Слайд 11

Результати розпізнавання штрихового коду



12

Рисунок В.12 – Слайд 12

Висновки

В ході виконання даної дипломної кваліфікаційної роботи було проаналізовано та досліджено процес розробки програмного забезпечення для розпізнавання образів.

Сформульовано функціональні вимоги до програмного забезпечення для розпізнавання штрихових кодів. Для реалізації програмного забезпечення для розпізнавання штрихових кодів обрано мову програмування C# та середовище розробки Microsoft Visual Studio.

Новизна роботи полягає в тому, що створено метод розпізнавання штрихових кодів на основі інтелектуальних обчислень, що складається з етапів зчитування зображення, визначення місця розташування штрих-коду на об'єкті, виділення складових частин штрих-коду, розшифрування інформації, закодованої у штриховому коді, виведення результату.

Практичне значення роботи полягає в тому, що розроблено програмне забезпечення для розпізнавання штрихових кодів.

Усі завдання випускної дипломної кваліфікаційної роботи повністю виконано.

14

Рисунок В.14 – Слайд 14