

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розпізнавання та реєстрація штрих кодів DataMatrix
у потоковому відео

Виконав: студент IV курсу, групи СН-43
спеціальності 122 Комп'ютерні науки

(шифр і назва спеціальності)

Абрамов М.О.
(підпис) (прізвище та ініціали)

Никитюк В.В.
(підпис) (прізвище та ініціали)

Шимчук Г.В.
(підпис) (прізвище та ініціали)

Боднарчук І.О.
(підпис) (прізвище та ініціали)

Станько А.А.
(підпис) (прізвище та ініціали)

Тернопіль - 2025

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
Боднарчук І.О.
(підпис) (прізвище та ініціали)
«__» _____ 2025 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Абрамову Миколі Олеговичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розпізнавання та реєстрація штрих кодів DataMatrix
у потоковому відео

Керівник роботи Никитюк Вячеслав Вячеславович, к.т.н., доц.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «07» 05 2025 року № 4/7-444

2. Термін подання студентом завершеної роботи 21.06.2025 р.

3. Вихідні дані до роботи наукові літературні джерела

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ

1. Аналіз предметної області.

2. Теоретична частина

3. Тестування сервісу. Сценарії використання

4. Безпека життєдіяльності, основи охорони праці

Висновки

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

1. Титулка. 2. Мета, задачі дослідження. 3. Послідовність розпізнавання штрих-кодів

DataMatrix. 4. Використовувані технології. 5. Модулі сервісу. 6. Архітектура сервісу

7. Сценарій 1. Запит із зображенням на пошук та декодування штрих-коду.

8. Результати декодування. 9. Сценарій 2. Виявлення та збереження штрих-кодів у відеопотоці. 10. Тестування безпечної передачі даних. Використання проксі-сервера Nginx.

11. Тестування безпечної передачі даних. Використання самопідписаного сертифіката.

12. Висновки. Основні результати проведеного дослідження

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	Сенчишин В.С., доц. каф. МТ		

7. Дата видачі завдання 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	08.05 – 09.05.25	Виконано
2.	Підбір джерел про розпізнавання та реєстрацію штрих кодів DataMatrix	10.05 – 13.05.25	Виконано
3.	Опрацювання джерел про розпізнавання та реєстрацію штрих кодів DataMatrix	14.05 – 17.05.25	Виконано
4.	Виконання дослідження щодо розробки програмного забезпечення для розпізнавання та реєстрації штрих кодів DataMatrix у потоковому відео	18.05 – 21.05.25	Виконано
5.	Розроблення програмного коду	22.05 – 24.05.25	Виконано
6.	Оформлення розділу «Аналіз предметної області»	22.05 – 23.05.25	Виконано
7.	Оформлення розділу «Теоретична частина»	25.05 – 27.05.25	Виконано
8.	Оформлення розділу «Тестування сервісу. Сценарії використання»	28.05 – 30.05.25	Виконано
9.	Виконання завдання до підрозділу «Безпека життєдіяльності, основи охорони праці»	31.05 – 03.06.25	Виконано
10.	Оформлення кваліфікаційної роботи	20.05 – 03.06.25	Виконано
11.	Нормоконтроль	01.06 – 05.06.25	Виконано
12.	Перевірка на плагіат	04.06 – 10.06.25	Виконано
13.	Попередній захист кваліфікаційної роботи	11.06 – 18.06.25	Виконано
14.	Захист кваліфікаційної роботи	28.06.25	

Студент

(підпис)

Абрамов М.О.

(прізвище та ініціали)

Керівник роботи

(підпис)

Никитюк В.В.

(прізвище та ініціали)

АНОТАЦІЯ

Розпізнавання та реєстрація штрих кодів DataMatrix у потоковому відео // Абрамов Микола Олегович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем та програмної інженерії, кафедра комп'ютерних наук, група СН-43 // Тернопіль, 2025 // С. – 50, рис. – 23, табл. – 1, слайдів – 12, бібліогр. – 13.

Ключові слова: DataMatrix, REST API, виявлення, відеопотік, розпізнавання, штрих код

Кваліфікаційна робота присвячена розробці сервісу розпізнавання та реєстрації штрих кодів DataMatrix у потоковому відео.

У першому розділі проаналізовано предметну область дослідження, описано особливості цифрового маркування як елементу захищення від нелегального товару. Наведено характеристики визначення двовимірного коду DataMatrix та його розпізнавання. Розглянуто методи розпізнавання штрих-кодів DataMatrix.

У другому розділі роботи описані програмні технології, котрі застосовуються при розробці сервісу. Спроектовано його архітектуру, котра містить чотири модулі (захоплення відео, розпізнавання, зберігання даних, HTTP-сервера). Докладно описано роботу кожного із них із наведенням фрагментів програмного коду скриптів на мові Python.

Третій розділ роботи присвячений тестуванню сервісу та двом сценаріям його використання. Протестовано безпечну передачу даних із застосуванням проксі-сервера Nginx та самопідписуваного сертифікату.

У четвертому розділі розглянуто важливі питання безпеки життєдіяльності та основ охорони праці.

ANNOTATION

Recognition and registration of DataMatrix barcodes in streaming video // Abramov Mykola // Ternopil Ivan Pul'uj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Science // Ternopil, 2025 // P. - 50, Fig. - 23, Table - 1, Slide - 12, References - 13.

Keywords: DataMatrix, REST API, detection, video stream, recognition, barcode

The qualification Thesis deals with the development of a service for recognizing and registering DataMatrix barcodes in streaming video.

The first section analyzes the subject area of the study, describes the features of digital marking as an element of protection against illegal goods. The characteristics of determining a two-dimensional DataMatrix code and its recognition are given. Methods for recognizing DataMatrix barcodes are considered.

The second section of the work describes the software technologies used in the development of the service. Its architecture is designed, which contains four modules (video capture, recognition, data storage, HTTP server). The operation of each of them is described in detail with fragments of the program code of scripts in the Python language.

The third section of the work is devoted to testing the service and two scenarios of its use. Secure data transmission using the Nginx proxy server and a self-signed certificate is tested.

The fourth section considers important issues of life safety and the basics of labor protection.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ СКОРОЧЕНЬ І ТЕРМІНІВ

API (англ. application programming interface) – інтерфейс програмування застосунків (прикладний програмний інтерфейс), є способом взаємодії комп'ютерних програм між собою.

DataMatrix – двовимірний матричний штрих-код, що має вигляд чорно-білих елементів або елементів декількох різних ступенів яскравості, зазвичай у формі квадрата, розміщені в прямокутній або квадратній групі. Матричний штрихкод призначений для кодування тексту або даних інших типів.

JSON (англ. JavaScript Object Notation) – текстовий формат обміну даними між комп'ютерами.

HTTP (англ. Hypertext Transfer Protocol) – протокол передачі гіпертекстових документів, протокол передачі даних, що використовується в комп'ютерних мережах.

OpenCV (англ. Open Source Computer Vision Library) – бібліотека комп'ютерного зору з відкритим кодом, містить функції та алгоритми комп'ютерного зору, обробки зображень і чисельних алгоритмів загального призначення з відкритим кодом.

REST API (англ. Representational State Transfer Application Programming Interface) – стандарт архітектури web -служб, що полегшує взаємодію між різними програмними компонентами.

SOAP (англ. Simple Object Access Protocol) — протокол обміну структурованими повідомленнями в розподілених обчислювальних системах, базується на форматі XML.

SSL (англ. Secure Sockets Layer) – рівень захищених сокетів, криптографічний протокол, який забезпечує встановлення безпечного з'єднання між клієнтом і сервером.

ПЗ – програмне забезпечення.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	9
1.1 Цифрове маркування товару як захист від нелегальної продукції.....	9
1.2 Робота зі штрих-кодами DataMatrix	10
1.2.1 Визначення двовимірного коду DataMatrix	10
1.2.2 Розпізнавання штрих-кодів DataMatrix	11
1.3 Дослідження методів розпізнавання штрих-кодів DataMatrix.....	14
РОЗДІЛ 2. ТЕОРЕТИЧНА ЧАСТИНА	18
2.1 Використовувані технології.....	18
2.2 Архітектура сервісу.....	20
2.2.1 Модуль розпізнавання штрих-кодів DataMatrix	21
2.2.2 Модуль відеозахоплення.....	23
2.2.3 Модуль HTTP -сервера. HTTP –протокол.....	24
2.3 REST API	26
2.3.1 Переваги REST API. Порівняння з SOAP API.....	26
2.3.2 REST API та зворотний проксі: ефективне зберігання та обмін інформацією про штрих-коди DataMatrix.....	27
РОЗДІЛ 3. ТЕСТУВАННЯ СЕРВІСУ. СЦЕНАРІЇ ВИКОРИСТАННЯ	29
3.1 Сценарій 1. Запит із зображенням на пошук та декодування штрих-коду.....	29
3.2 Сценарій 2. Виявлення та збереження штрих-кодів у відеопотоці	33
3.3 Тестування безпечної передачі даних	35
3.3.1 Використання проксі-сервера Nginx	36
3.3.2 Використання самопідписаного сертифіката.....	37
РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	41
4.1 Навчання працюючих і інструктажі з охорони праці.....	41
4.2 Санітарно-гігієнічні вимоги до умов праці.....	43
ВИСНОВКИ	46
ПЕРЕЛІК ДЖЕРЕЛ.....	47
ДОДАТКИ	

ВСТУП

Актуальність теми. У світі збір, обробка, форматування і передача інформації є цілісною складовою життя людини. Властиво інформаційні технології торкаються всіх сфер діяльності людства, особливо у виробництві та управлінні. У зв'язку з цим існує необхідність швидкого і точного введення інформації про будь-який об'єкт на електронному пристрої для її подальшої обробки.

У розпізнаванні цифрових маркувань на продукції також є необхідність швидкого та безпомилкового зчитування закодованої інформації про товар для її подальшої передачі [1]. У світі основним засобом ідентифікації товарів за обов'язкового маркування є двомірний штрих-код DataMatrix.

Проте, за великих обсягів продукції з виробництва виникає проблема збільшення часу виконання процесу розпізнавання штрих кодів для усіх товарів. Сервіс дозволить зчитувати та обробляти закодовану інформацію на продуктах із високою швидкістю, що значно прискорить процес ідентифікації товарів на виробництві. Крім того, даний сервіс матиме високу точність розпізнавання, що дозволить знизити ймовірність помилок при ідентифікації продуктів.

Розробка подібного сервісу є актуальним і важливим завданням, враховуючи популярність маркування продукції з використанням двійкових кодів в останні роки. Таке зростання пов'язане з вибором штрих -коду DataMatrix для обов'язкового маркування товару. У перспективі до переліку категорій, що підлягають даному маркуванню, увійдуть усі товари. Також буде реалізовано відстеження всієї продукції, яка була виготовлена або реалізована на території України. Отже, зацікавленість інформаційного співтовариства до розробки програмних продуктів для ефективного зчитування двійкових кодів DataMatrix буде виявлятися як в даний час, так і в майбутньому.

Мета роботи – створення сервісу для розпізнавання та реєстрації штрих-кодів DataMatrix у потоковому відео.

Для досягнення мети виділено ряд завдань:

- аналіз існуючих алгоритмів розпізнавання штрих-кодів DataMatrix, виявлення їх переваг та недоліків;
- вивчення та вибір інструментів, необхідних для реалізації сервісу;
- проектування архітектури сервісу;
- реалізація сервісу;
- розробка web-інтерфейсу для реєстрації штрих-кодів;
- тестування та налагодження сервісу для забезпечення коректної роботи.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Цифрове маркування товару як захист від нелегальної продукції

В останні роки проблема нелегальної продукції стала дедалі актуальнішою. Нелегально вироблені товари не лише створюють загрозу здоров'ю та безпеці споживачів, а й завдають шкоди легальним виробникам, які зазнали конкуренції з нечесними постачальниками.

Для боротьби з нелегальною продукцією було створено багато методів та технологій. До 2015 року питання створення системи відстеження переміщення такої продукції в Україні залишалося відкритим. Проте на той момент вже було виявлено понад 1 мільйон предметів сірої продукції на ринку. У відповідь цю проблему в Україні почалося використання системи обов'язкового маркування продукції, основним компонентом якої став Національний знак відповідності [2].

Система маркування товарів дозволяє формувати унікальний матричний штрих-код DataMatrix для кожного товару. Штрих-код DataMatrix має бути просканий на всіх етапах перевезення продукції, щоб можна було простежити весь шлях.

У випадку із зарубіжною продукцією, імпортери несуть відповідальність за маркування товару. Вони повинні перевірити наявність маркування товару перед його відправкою, і якщо його немає, вона має бути нанесена складі консолідації чи процесі виробництва.

На останньому етапі продажу касир сканує штрих-код DataMatrix, який потім передається через фіскальний накопичувач для видалення з товарообігу.

Введення системи Національного знаку відповідності та штрих-коду DataMatrix має велике значення для всіх учасників торговельних відносин. Контрафактна або низькоякісна продукція може призвести до збитків виробнику, споживачеві та спричинити великі фінансові втрати для обох сторін. Тому система маркування є необхідним заходом для забезпечення справжності та якості продукції.

1.2 Робота зі штрих-кодами DataMatrix

1.2.1 Визначення двовимірного коду DataMatrix

Для створення сервісу в першу чергу необхідно розуміти, що є кодом DataMatrix, як він працює і які завдання може вирішувати.

Штрих-код - це унікальний код, що складається з послідовності смуг білого та чорного кольору, що використовується для зберігання та передачі інформації про товар у зашифрованому вигляді, що зчитується спеціальними електронними пристроями.

Є два типи штрих-кодів: лінійні та двовимірні. Перші працюють за принципом зчитування інформації у горизонтальному напрямку, що дозволяє використовувати відносно дешеве зчитування. Проте їх недолік полягає в обмеженій ємності - такий код може містити не більше 30 символів.

На відміну від лінійних, двовимірні штрих-коди можуть зберігати набагато більший обсяг інформації. З них найбільш широко використовується код DataMatrix [3].

DataMatrix - це матричний штрих-код із двома вимірами, що складається з чорних та білих елементів різної яскравості. Всі ці елементи укладені в групи, які можуть мати прямокутну або квадратну форму. Цей штрих-код може містити до 1556 байт інформації, що дозволяє його використання у широкому спектрі додатків - від відстеження товарів до ідентифікації з медичною метою. Технологія DataMatrix дозволяє кодувати дані в різних режимах, що підвищує його гнучкість та застосовність.

Приклад DataMatrix коду представлений на рис. 1.1.

Ключовою перевагою штрих-коду DataMatrix є його компактний розмір. Навіть на квадратній поверхні розміром 2 на 2 міліметри можна закодувати до 50 символів, що недоступно для лінійних штрих-кодів. Крім того, DataMatrix легко зчитується електронними пристроями, так як вони можуть ідентифікувати інформацію незалежно від орієнтації приладу, що зчитує [4].



Рисунок 1.1 – Приклад DataMatrix коду рядка "test code"

На сьогоднішній день є п'ять способів нанесення кодів DataMatrix. Перший метод передбачає прямий друк штрих-коду на виріб з використанням струминного друку або гравіювання за допомогою лазера або голок. Другий спосіб полягає в друкуванні на паперових та поліпропіленових матеріалах за допомогою термотрансферних принтерів. Три методи, що залишилися, призначені для друку DataMatrix на принтерах, які працюють на основі лазерної, струменевої та матричної технологій.

1.2.2 Розпізнавання штрих-кодів DataMatrix

Можна виділити 6 етапів алгоритму розпізнавання коду DataMatrix:

1) бінаризація – на вхід подається повнокольорове зображення, після чого виділяються межі темних груп точок [4]. Приклад зображення штрих-коду DataMatrix до обробки та після наведені нижче (рис. 1.2 та 1.3);

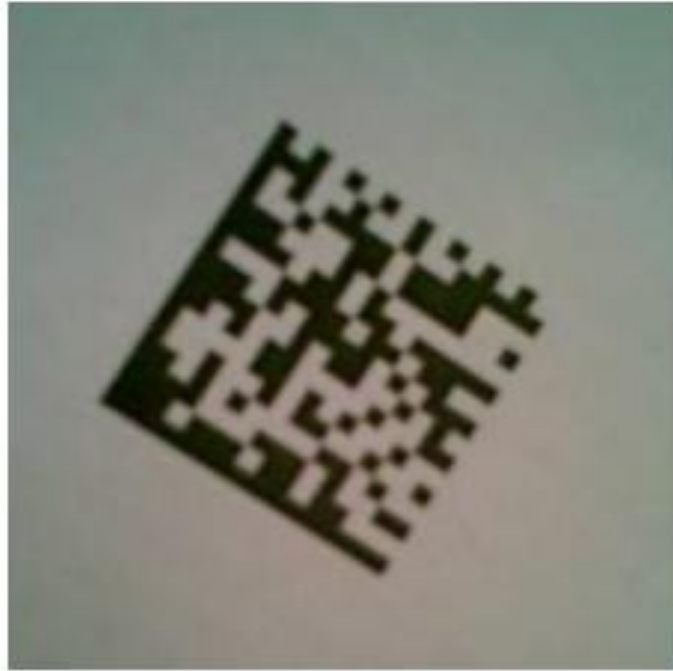


Рисунок 1.2 – Отримане камерою зображення

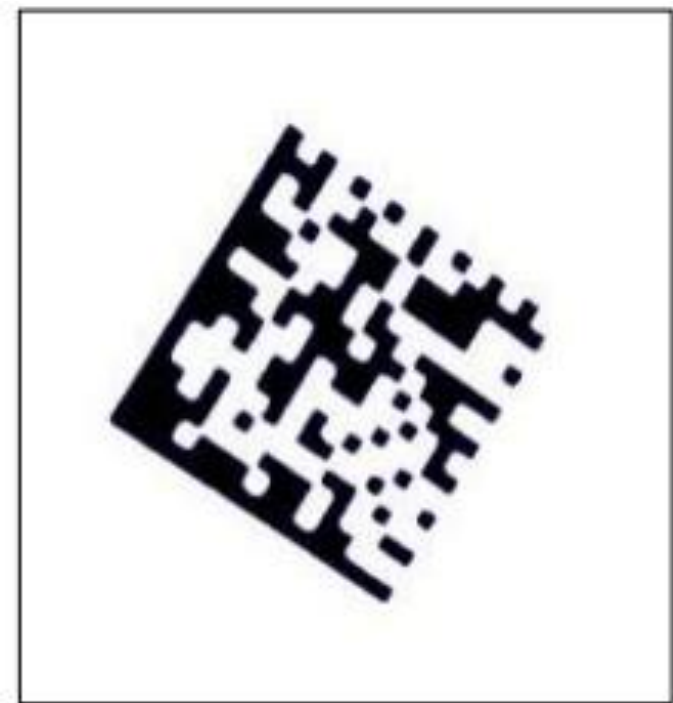


Рисунок 1.3 – Оброблене зображення

2) пошук групи точок - у кожному виділеному ділянці визначаються точки, які стосуються коду, і видаляються ті, які мають мале значення. Формуються темні блоки, обмежені однією лінією. Цей процес представлений на рис. 1.4;

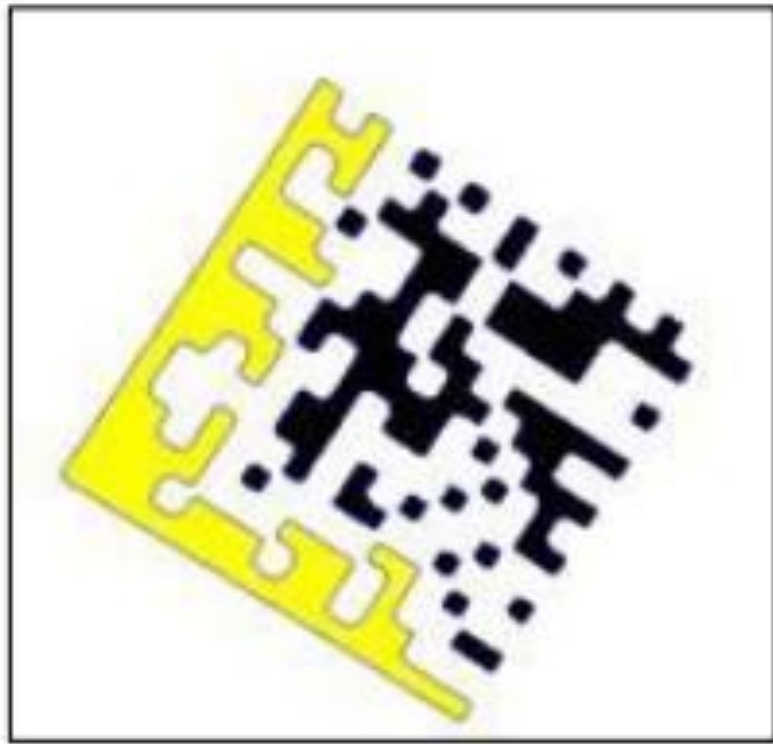


Рисунок 1.4 – Виділення чорної області

3) визначення меж коду - виділяється мінімальна обмежуюча прямокутна область, після чого визначаються 4-8 точок, найближче розташованих до кожної сторони прямокутника. Потім за цими точками будуються лінії, і оптимальним кордоном вважається той, був утворений максимальною кількістю точок;

4) визначення нахилу коду - фіксуються зміни розмірів точок коду, викликані різним кутом нахилу камери. В результаті знімка коду може вийти зображення, яке не є правильним чотирикутником. Щоб виявити зміни розмірів крапок, необхідно аналізувати межі коду з найменшою концентрацією крапок. Кожна зміна кута нахилу камери фіксується, а потім використовується для знаходження лінійних трендів зміни розмірів кожної сторони. Для обчислення цих трендів можна використати таку формулу

$$C = (L_1, L_2),$$

де L_1 та L_2 - формули для ліній.

Вони, так само, можуть бути обчислені як:

$$s_i = a * i + b,$$

де s – розмір пари точок, i – номер пари, a та b – параметри лінії;

5) побудова матриці - визначаються значення меж матриці коду та пропорції зміни межі розмірів точок та їх кількість.

Оптимальні розміри пар точок можна обчислити за такою формулою:

$$\sum_{i=1}^m s_i = len,$$

де m – кількість пар точок, len – довжина сторони.

Потім визначаються точки на кожній стороні прямокутника, і сітка для визначення матриці формується з ліній, побудованих по парах точок, що знаходяться на протилежних сторонах коду. Матриця буде сформована коректно, якщо на етапі бінаризації оброблене зображення було виділено правильно. В іншому випадку, матриця буде визначена некоректно;

6) декодування - викликається функція, яка повертає інформацію у разі правильної побудови матриці та помилку у разі пошкодження коду.

1.3 Дослідження методів розпізнавання штрих-кодів DataMatrix

Існують різні методи, якими можна нанести маркування на товари, такі як лазерний друк, термотрансферний перенесення, гравіювання, але найбільш зручним і швидким для більшості виробництв є наклейка етикеток з 2D-кодами (QR або DataMatrix [5]), які друкуються друкарським способом. Проте виробники

продукції обов'язково стикаються із завданнями контролю та цілісності 2D -коду, а також його розпізнавання та зчитування з подальшим внесенням інформації до інформаційної системи підприємства. У зв'язку з впровадженням кодів DataMatrix у різні галузі діяльності постало питання про те, за допомогою яких алгоритмів відбуватиметься розпізнавання штрих-кодів на різних матеріалах та поверхнях.

Стаття [6] описує метод визначення орієнтації коду DataMatrix з допомогою алгоритму автоматичного зсуву. Метод заснований на оцінці координат об'єктів на зображенні і може працювати на різних видах кодів, що тестуються.

Однак слід зазначити, що даний метод має деякі недоліки. Одним із обмежень алгоритму середнього зсуву є те, що він може не справлятися з деякими видами шуму, такими як шум або градієнтний шум від освітлення. Крім того, цей метод також не враховує можливі спотворення зображення, такі як спотворення перспективи або спотворення, пов'язані з розрізом об'єктива камери.

Ще одним недоліком алгоритму середнього зсуву є те, що він спирається на припущення про те, що модулі коду DataMatrix мають прямокутну форму і розташовані в регулярній сітці. Це обмежує застосування алгоритму на випадки, коли модулі мають нестандартну форму або розташовані у довільному порядку.

Проте, цей метод є ефективним і застосовним для визначення орієнтації коду DataMatrix у більшості випадків, що робить його корисним інструментом для роботи з такими кодами. В цілому, стаття представляє інтерес і може бути корисною для фахівців, які займаються обробкою зображень і розпізнаванням кодів.

У роботі [7] автори представляють новий метод сканування кодів DataMatrix, нанесених на криволінійні поверхні, використовуючи уявну сітку точок, що має ту ж орієнтацію, що й поверхня. Автори описують метод локалізації модулів коду DataMatrix, що дозволяє сканувати модулі незалежно від виду камери чи орієнтації.

Однак, варто зазначити, що цей метод має свої недоліки. Автори не врахували багато факторів, таких як погане освітлення, різний кут нахилу камери і т.д., що може суттєво вплинути на точність сканування. Крім того, для сканування необхідне точне калібрування камери, що може бути проблематично

в реальних умовах виробництва.

Тим не менш, цей метод може бути корисним у випадках, коли потрібно сканувати коди DataMatrix на складних поверхнях, таких як сферичні поверхні. Він може застосовуватися у різних сферах, як то виробництво, логістика або медицина.

В цілому описаний метод має обмежене застосування в реальних умовах і вимагає додаткового доопрацювання для підвищення точності та стійкості до умов навколишнього середовища.

У науковій статті [8] сформульовано недоліки вищезазначених алгоритмів розпізнавання DataMatrix, а саме, велику чутливість до перешкод і нестійкість поведінки при повороті та нахилі камери. Для вирішення цієї проблеми автор розробив власний алгоритм, що враховує поворот камери на довільний кут.

Оскільки для штрих-кодів DataMatrix, нанесених на різні поверхні властивий широкий розкид умов розпізнавання, автором статті було проведено експеримент, до якого увійшли зображення з різними умовами освітлення, якістю нанесення штрих-кодів та кольором поверхні. При створенні даного алгоритму були застосовані різні методи бінаризації та налаштування обробки зображень для досягнення кращого результату.

Так, в ході експериментів було використано п'ять режимів бінаризації та чотири бінаризації порівняння з еталоном кола різних радіусів, що зафарбовується. Також для аналізу перепадів яскравості на зображенні було застосовано метод виявлення бінарних об'єктів, створений тим самим автором [9]. Даний метод заснований на блоковій сегментації зображень з використанням гістограмних ознак і спрямований на підвищення точності та достовірності розпізнавання об'єктів на зображенні в умовах різного освітлення та зміни метеоумов зйомки.

У ньому представлені розподіл зображення на зони та виділення структурних елементів, при цьому розподіл зображення на зони здійснюють шляхом фільтрації об'єктів на бінарному контурному зображенні.

Оскільки в системах реального часу розпізнавання штрих-кодів відбувається з різних поверхонь і в складних умовах, прийняття рішення

відводиться обмежений час.

Алгоритм, представлений у [9], допоможе вирішити проблему в ідентифікації баркодів DataMatrix в умовах з підвищеним шумовим фоном. Саме тому він був обраний для реалізації модуля розпізнавання штрих-кодів DataMatrix.

РОЗДІЛ 2. ТЕОРЕТИЧНА ЧАСТИНА

2.1 Використовувані технології

З огляду на те, що алгоритм для розшифрування кодів DataMatrix є досить ресурсомістким, доцільно використовувати мову програмування, котра компілюється, а не інтерпретується. При виборі мови програмування також потрібно враховувати можливість перенесення програми на інші платформи. Тому необхідно, щоб засоби організації багатопроцесорної обробки були частиною обраної мови програмування, а не API використовуваної операційної системи.

Мова програмування C++ повністю відповідає цим вимогам. Логотип ПЗ наведено на рис. 2.1.



Рисунок 2.1 – Лого мови C++

Її вибір для вирішення цього завдання обумовлений кількома основними причинами:

- швидкість, оскільки програми компілюються в машинний код, а у стандартній бібліотеці використовуються оптимізовані алгоритми і структури даних;
- універсальність, так як компілятори цієї мови існують у кожній операційній системі, абсолютна більшість застосунків успішно переноситься із однієї платформи на іншу;
- активна підтримка, оскільки постійно вдосконалюється стандарт мови

та впроваджуються поліпшення;

- підтримка об'єктно-орієнтованого програмування.

Qt було обрано як середовище розробки завдяки його простоті використання та багатофункціональності. Його логотип показано на рис. 2.2.



Рисунок 2.2 – Лого середовища Qt

Qt надає багатий набір інструментів та бібліотек, які спрощують процес розробки та прискорюють його. Крім того, дане середовище розробки дозволяє створювати кросплатформні програми, що було важливим фактором при виборі середовища розробки.

Окрім мови програмування C++ та середовища розробки Qt були використані й інші технології. Наприклад, для обробки та аналізу зображень була використана бібліотека OpenCV. Це дозволило ефективно обробляти зображення, виділяти з них штрих-коди та розпізнавати їх. Лого OpenCV показано на рис. 2.3.



Рисунок 2.3 – Лого бібліотеки OpenCV

2.2 Архітектура сервісу

У цьому підрозділі буде розглянуто архітектуру сервісу для розпізнавання та реєстрації штрих-кодів DataMatrix у потоковому відео. До складу сервісу входить набір модулів, кожний з котрий виконує визначені завдання:

- модуль захоплення відео відповідає за отримання відеопотоку з камери чи іншого джерела;
- модуль розпізнавання виконує завдання аналізу відео та пошуку штрих-кодів DataMatrix на кожному кадрі;
- модуль зберігання даних відповідає за збереження інформації про розпізнані штрих-коди у базі даних;
- модуль HTTP -сервера надає REST API для взаємодії з сервісом за допомогою протоколу HTTP.

Основні компоненти сервісу представлені на рис. 2.4.

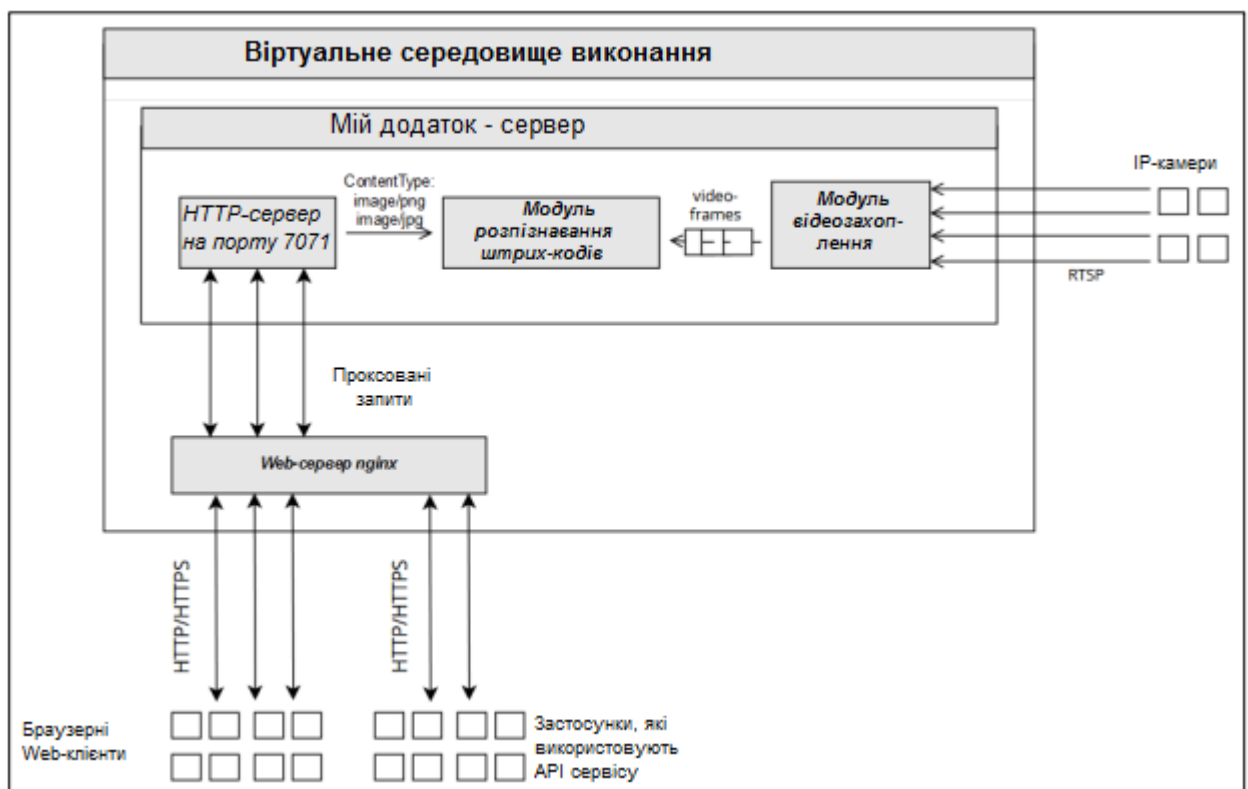


Рисунок 2.4 – Архітектура сервісу

Архітектура сервісу є важливим аспектом його розробки, оскільки визначає

спосіб взаємодії компонентів між собою. Надійність та ефективність сервісу залежать від правильного проектування архітектури та взаємодії його компонентів. У наступних підрозділах кожен із модулів та описувані ним функції будуть розглянуті докладніше.

2.2.1 Модуль розпізнавання штрих-кодів DataMatrix

Цей модуль відповідає за роботу з відеопотоком і виділення зображення, що містить штрих-коди. Він повинен включати алгоритми і технології, які дозволяють ефективно і точно розпізнавати DataMatrix коди в потоковому відео. Крім того, модуль повинен бути гнучким і налаштованим, щоб забезпечувати високий рівень точності розпізнавання, а також підтримувати різні налаштування камер, з яких вони захоплюються. Важливо також, щоб модуль був економічним за ресурсами, щоб обробка відеопотоку не займала занадто багато часу і не навантажувала сервер.

У модулі розпізнавання штрих-кодів важливу роль відіграє бібліотека OpenCV. За допомогою OpenCV можна налаштувати параметри фільтрації, налаштування кольору, параметри контрастності та яскравості, щоб оптимізувати зображення для розпізнавання штрих-коду. Також у OpenCV є інструменти детектування контурів та ліній на зображенні, що дозволяє виділити межі штрих-коду та вирізати його із зображення [8].

Для розробки модуля розпізнавання штрих-кодів DataMatrix був реалізований клас QmDecoder, який містить два статичні методи декодування QR-кодів з зображень QImage або cv::Mat (матриця OpenCV). Метод decode приймає вхідне зображення і декодує в ньому QR-коди, зберігаючи результати QVector об'єктів QmDecoderResult (лістинг 2.1).

Лістинг 2.1 – Фрагмент коду класів QmDecoder та QmDecoderResult

```
class QmDecoderResult;
class QmDecoder
{
public:
    static int decode(const QImage& img, QVector<QmDecoderResult>
&results, int stop_after_n = 1, long stop_after_timeout = 200);
    static int decode(const cv::Mat cvframe, QVector<QmDecoderResult>&
results, int stop_after_n = 1, long stop_after_timeout = 200);
};

class QmDecoderResult
{
public:
    QmDecoderResult(const QString& data, const QPolygon& polygon =
QPolygon())
        : mMessage(data), mPolygon(polygon)
    {
    }
    ~QmDecoderResult() {
    }
    const QString& message() const { return mMessage; }
    const QPolygon& polygon() const { return mPolygon; }

    QJsonObject toJson() const {
        QJsonObject jobj;
        jobj.insert("message", mMessage);
        return jobj;
    }
};
```

Повністю програмний код наведено у Додатку А.

Функція `dmtxDecodeImage` відповідає за декодування QR -кодів у зображенні, представленому як об'єкт `DmtxImage`. Вона використовує функцію `dmtxDecodeCreate` для створення об'єкта `DmtxDecode`, який використовується для пошуку областей зображення, які можуть містити QR- коди. Функція `dmtxRegionFindNext` використовується для пошуку наступної області, і функція `dmtxDecodeMatrixRegion` використовується для декодування QR -коду в цій області. Розшифроване повідомлення та багатокутник, представляє положення QR- коду на зображенні, зберігаються в об'єкті `QmDecoderResult` і додаються до результатів вектора.

Клас `QmDecoderResult` є результатом декодування одного QR- коду. Він зберігає розшифроване повідомлення у вигляді `QString` та багатокутник у вигляді

QPolygon.

Псевдокод для алгоритму, описаного вище, має вигляд, наведений у лістингу 2.2.

Лістинг 2.2 –Псевдокод алгоритму

```
Клас QmDecoder:
Статичний метод decode(qImage) :
Результат = QVector<QmDecoderResult>()
Розмір = qImage.size()
Якщо Розмір < 21, повернути Результат
qrКоди = ЗнайтиQrКоди(qImage)
Для кожного qrКоду в qrКодах:
Декодувати QRКод)
ЗберегтиРезультат QRКод, Результат)
Повернути Результат
Статичний метод decode (cvMat):
Результат = QVector<QmDecoderResult>() qImage =
Перетворити BQImage(cvMat)
Результат = decode(qImage)
Повернути Результат
Функція Знайти QrКоди (qImage):
qrКоди = []
Пошук усіх можливих областей, які можуть містити qr -коди Для кожної знайденої
області:
Якщо це qr -код:
qrКоди. append (область) повернути qrКоди
Функція Декодувати QRКод):
Структура QR = розшифрувати QRКод)
Зберегти QR .повідомлення та QR.полігон
Функція ЗберегтиРезультат(qrКод, Результат)
QR = структура з повідомленням та полігоном Результат. append(QR)
```

2.2.2 Модуль відеозахоплення

Даний модуль повинен включати компоненти для отримання відеопотоку від IP камер або інших пристроїв відеоспостереження, а також для обробки і передачі цього потоку далі для розпізнавання штрих-кодів.

Для цього модуль повинен мати можливість підключення до IP -камер через протоколи HTTP, RTSP та інші. Також він повинен мати засоби для обробки відеопотоку та його трансформації у зручний формат для подальшої обробки.

Модуль повинен володіти гнучкими налаштуваннями, щоб можна було

налаштувати якість і дозвіл запису відеопотоку, швидкість передачі і т.д. Важливо, щоб він мав можливість отримання відеопотоку від множини камер одночасно, а також була можливість реєстрації подій у потоці, пов'язаних з виявленням об'єктів або змінами станів.

Відеозахоплення реалізовано у класі QMCamera. Він успадкований від класу QThread з бібліотеки Qt, який є окремим потоком управління в програмі. Для забезпечення синхронізації доступу до загальних змінних-членів класу використовується QMutex mLock.

У класі є кілька методів для керування параметрами камери та її потоком, а також два сигнали, що відправляються при отриманні кадру та після його декодування. Також у класі є метод fromJson, який встановлює URL- камери з переданого JSON -об'єкта.

При запуску потоку в методі run відбувається відкриття відеопотоку з вказаною URL-адресою і зчитування кадрів за допомогою функції vcap.read(cvframe). Якщо кадр був успішно зчитаний, то він відправляється за допомогою сигналу cvFrameReady і якщо mDecodingTaskCounter < 3, відбувається асинхронний запуск методу decode для декодування штрих-кодів на кадрі. Потім потік засипає на вказану кількість мілісекунд і відбувається новий цикл зчитування кадрів.

Метод decode декодує штрих-коди на переданому кадрі за допомогою класу QmDecoder і надсилає результат за допомогою сигналу frameDecoded.

2.2.3 Модуль HTTP -сервера. HTTP -протокол

Модуль HTTP -сервера відіграє важливу роль у розробці сервісу для розпізнавання штрих-кодів DataMatrix. Він відповідає за обробку вхідних HTTP -запитів та надсилання відповідей клієнту. HTTP -протокол є стандартним протоколом передачі даних в Інтернеті і дозволяє забезпечити безпеку, цілісність і конфіденційність даних, що передаються.

Щоб реалізувати HTTP -сервер, можна використовувати стандартні бібліотеки мови програмування C++, такі як Boost.Asio або Qt Network. Для цього

проекту було обрано останній варіант. Він надає набір класів та функцій, що дозволяють легко реалізувати серверну частину програми.

HTTP -сервер повинен приймати запити від клієнта та повертати результат декодування штрих-коду у форматі JSON. Це можна реалізувати за допомогою класу QTcpServer та QTcpSocket, який надає засоби для встановлення з'єднання з клієнтом та передачі даних між сервером та клієнтом.

HTTP -протокол має свій формат запиту та відповіді. Запит складається з кількох частин, таких як метод, URI, заголовки та тіло запиту. Відповідь так само містить код стану, заголовки і тіло відповіді. Коректна обробка запитів та формування відповідей є важливим аспектом під час реалізації HTTP -сервера.

Для забезпечення безпеки передачі даних HTTP -сервер повинен використовувати протокол HTTPS замість HTTP. Для цього необхідно використовувати сертифікат SSL/TLS, який дозволить зашифрувати передані дані та захистити їх від несанкціонованого доступу.

З метою забезпечення безпеки та захисту від шкідливих атак було реалізовано механізм зворотного проксування з використанням сервера Nginx. Він забезпечує функції балансування навантаження, кешування та маршрутизації запитів, а також може діяти як SSL - термінація.

Nginx використовується як зворотний проксі для того, щоб приховати інформацію про HTTP -сервер, що працює на сервері додатків. Це забезпечує покращення безпеки, тому що зловмисник не зможе отримати інформацію про технології, що використовуються, що ускладнить проведення атак на сервер.

Крім того, Nginx дозволяє налаштувати доступ до ресурсів за IP -адресою, доменним ім'ям, портом та іншими параметрами запиту. Це дозволяє ефективно контролювати доступ до програми та захистити її від можливих загроз.

З використанням Nginx можна налаштувати захист від DDoS -атак, блокування запитів з підозрілою активністю, а також обмеження частоти запитів з однієї IP -адреси. Це дає змогу підвищити рівень безпеки і уникнути можливих проблем з доступністю програми.

Таким чином, модуль HTTP -сервера має визначну роль для гарантування працездатності і безпечності сервісу для розпізнавання штрих-кодів DataMatrix.

Застосування стандартних бібліотек мови програмування C++ та використання протоколу HTTPS забезпечує високу продуктивність та захист переданих даних.

2.3 REST API

2.3.1 Переваги REST API. Порівняння з SOAP API

REST API використовує протоколи HTTP та HTTPS для обміну даними між клієнтом та сервером. Володіє низкою переваг перед іншими типами API. Однією з основних є його простота та легковаговість. REST API ґрунтується на протоколі HTTP, який широко використовується в Інтернеті. Це означає, що API REST легко інтегрується з іншими додатками і має простий і зрозумілий інтерфейс для взаємодії. Також REST API дозволяє використовувати кешування, що може знизити навантаження на сервер і прискорити відповіді на запити [10].

REST API дозволяє використовувати множину різних форматів даних, включаючи JSON, XML, CSV, YAML та багато інших. Це дає можливість вибрати найбільш відповідний формат для конкретної програми та покращити продуктивність (рис. 2.5).

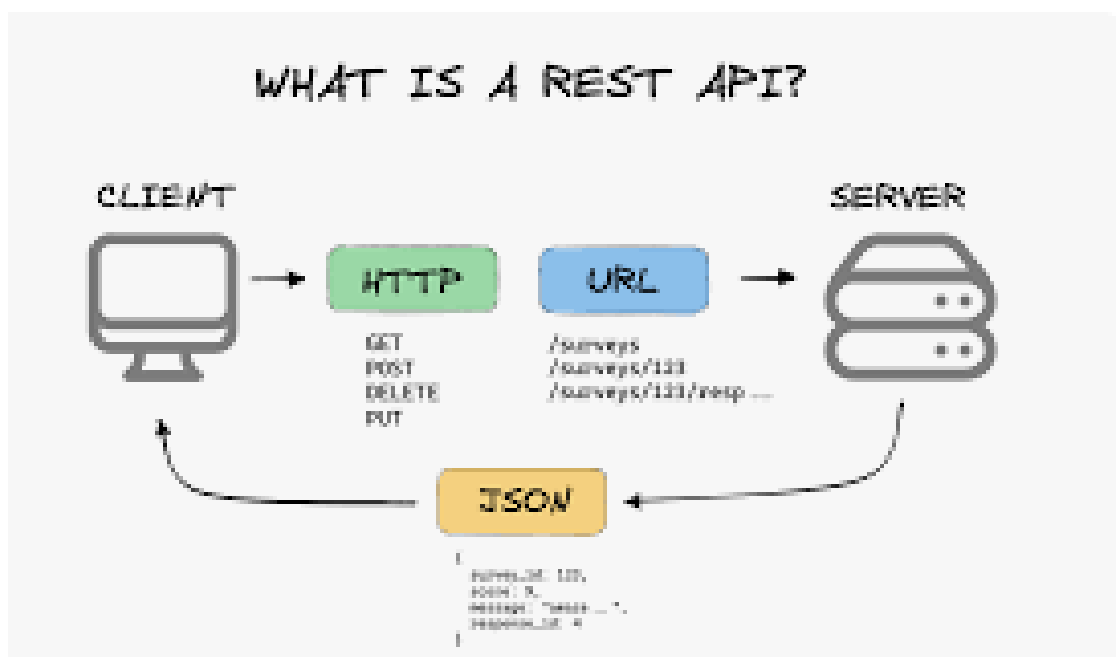


Рисунок 2.5 – Структура REST API

Ще однією важливою перевагою REST API є можливість гнучкої масштабованості. Цей тип API дозволяє легко додавати нові ресурси і методи, що робить його адаптивним до потреб, що змінюються. Крім того, REST API не зберігає стан між запитами, що спрощує масштабування та гарантує надійність та безпеку системи.

Даний тип API також забезпечує універсальність, що означає, що він може бути використаний на різних платформах та різних мовах програмування. Це робить REST API дуже гнучким та зручним для розробки розподілених систем.

Існують інші типи API, наприклад, SOAP, проте, порівняно з REST API, він має ряд недоліків. SOAP-API має складніший і великоваговіший протокол, що ускладнює його використання в розподілених додатках. Крім того, SOAP API не забезпечує універсальність та гнучкість, які є у REST API. З іншого боку, цей протокол пропонує ширший функціонал, включаючи обробку помилок та підтримку транзакцій. Однак, цей функціонал може бути надлишковим для деяких програм і SOAP API вимагає більш складного налаштування та керування.

Підсумовуючи, можна сказати, що порівняння REST API з іншими типами API показує, що кожен з них має свої переваги та недоліки, і вибір конкретного типу API залежить від вимог та потреб конкретної програми. Такий архітектурний стиль як REST API краще підходить для створення розподілених програм, особливо у випадках, коли потрібно обробляти потокове відео.

2.3.2 REST API та зворотний проксі: ефективне зберігання та обмін інформацією про штрих-коди DataMatrix

Сервіс використовує REST API для надання інформації про розпізнані штрих-коди DataMatrix стороннім програмам. Для запиту результатів роботи сервісу використовується метод HTTP-GET, а для надсилання зображень сервісу – HTTP-POST

REST API працює на основі HTTP -запитів, і повертає відповіді у вигляді JSON -об'єктів, що містять інформацію про запит DataMatrix. Таким чином, сервіс може легко інтегруватися з іншими додатками, дозволяючи їм швидко та

ефективно отримувати доступ до інформації про штрих-код DataMatrix, необхідної для виконання своїх завдань.

Для розробки API REST необхідні знання теоретичних концепцій, пов'язаних з архітектурою REST, такі як принципи обмеження інтерфейсу, однозначність запитів, кешування, і так далі.

Крім того, сервіс використовує зворотний проксі для керування запитами, що входять до REST API. Зворотний проксі дозволяє розподілити трафік між різними екземплярами сервісу, захистити його від атак та забезпечити високу доступність та продуктивність.

Також варто відзначити, що зворотний проксі може використовуватися для налаштування маршрутизації запитів та керування доступом до різних ресурсів на сервері. Наприклад, можна налаштувати зворотний проксі таким чином, щоб певні запити перенаправлялися на різні сервіси або мікросервіси на сервері залежно від їхнього типу або адреси.

Загалом використання зворотного проксі в даній роботі може забезпечити безпеку REST API, приховати реальну адресу сервера від клієнтів та захистити його від деяких видів атак. Крім того, він може покращити продуктивність REST API, кешуючи відповіді на запити та балансуючи навантаження між кількома серверами, якщо це необхідно.

Таким чином, розробка сервісу заснована на принципах REST API та зворотного проксі, які дозволяють забезпечити ефективне зберігання та обмін інформацією про розпізнані коди DataMatrix між сервісом та сторонніми програмами.

РОЗДІЛ 3. ТЕСТУВАННЯ СЕРВІСУ. СЦЕНАРІЇ ВИКОРИСТАННЯ

Тестування є невід'ємною частиною розробки будь-якого ПЗ, включаючи створений сервіс розпізнавання штрих-кодів DataMatrix в потоковому відео. Воно дозволяє переконатися у працездатності та відповідності сервісу вимогам, а також виявити та виправити можливі проблеми. У цьому розділі будуть розглянуті результати тестування та оцінено його ефективність у різних сценаріях використання [11].

3.1 Сценарій 1. Запит із зображенням на пошук та декодування штрих-коду

В межах першого сценарію використання сервісу, програма-клієнт надсилає запит із картинкою штрих-коду на розпізнавання та декодування. Результати повертаються у форматі JSON об'єкта. Цей сценарій дозволяє користувачам отримувати інформацію про вміст штрих-коду, використовуючи власні клієнтські програми.

Для демонстрації роботи сервісу було створено 3 файли формату PNG, що містять коди DataMatrix (рис. 3.1 – 3.3).

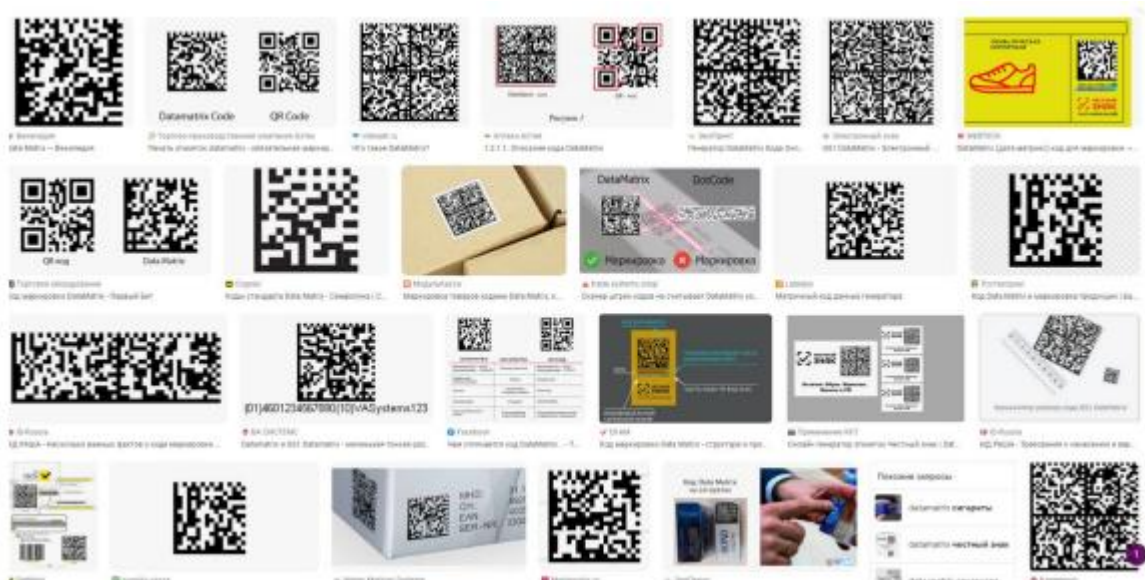


Рисунок 3.1 – Файл test-1.png



Рисунок 3.2 – Файл test-2.png



Рисунок 3.3 – Файл test-3.png

Важливо переконатися, що зображення містять правильні коди. Далі було запущено HTTP-сервер на порту 7071 (рис. 3.4):

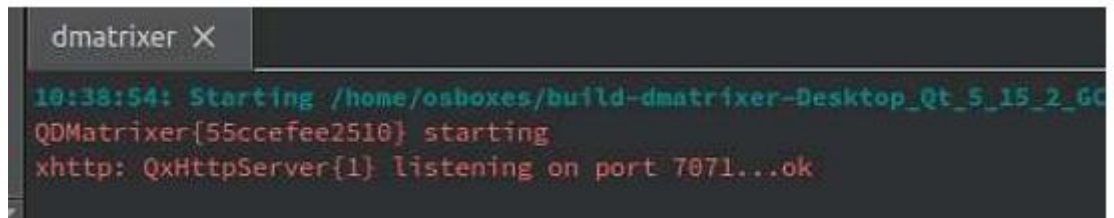


Рисунок 3.4 – Запуск сервера

За допомогою HTTP-POST запиту клієнтська програма надсилає зображення до сервісу для подальшого розпізнавання. У запиті вказана адреса сервера та дані, в даному випадку - саме зображення. Сервіс повинен знайти коди DataMatrix та повернути результат.

Для цього було створено скрипт мовою Python (лістинг 3.1). Як параметри запиту була вказана URL -адреса кінцевої точки endpoint, яка є посиланням на API сервісу декодування. У скрипті вказується ім'я файлу (fname), який буде надіслано на сервер, а також максимальну кількість штрих-кодів для розшифровки (Barcode_max_count).

Лістинг 3.1 – Фрагменту коду скрипта

```
endpoint = 'http://localhost:7071/api/v1/decoder'
try:
    fname = './test-1.png'
    fmime_type = mimetypes.guess_type(fname)[0]
    barcode_max_count = 5
    print('send file to decoding...')
    f = open(fname, 'rb')
    r = requests.post(endpoint, headers={
        'Content-Type': fmime_type,
        'BarcodeMaxCount': str(barcode_max_count)},
        data=f.read())

    r.raise_for_status()
    print('Result:\n', r.text)

except requests.exceptions.HTTPError as errh:
    print ("Http Error:", errh)
except requests.exceptions.ConnectionError as errc:
    print ("Error Connecting:", errc)
except requests.exceptions.Timeout as errt:
    print ("Timeout Error:", errt)
except requests.exceptions.RequestException as err:
    print (err)
```


Повністю код наведений у Додатку А.

Далі застосовується модуль `requests` для надсилання запиту POST до сервісу. Файл зчитується в двійковому режимі, а потім надсилається сервісу за допомогою методу `requests.post()`. У заголовках запиту вказується тип вмісту файлу, максимальна кількість штрих-кодів для розшифровки та інші необхідні параметри.

Після надсилання запиту скрипт отримує відповідь від сервісу. У даній відповіді очікується JSON -об'єкт із результатами розшифровки, який буде виведений на консоль.

Результат повертається у вигляді JSON -об'єкта, що містить дані, знайдені на зображеннях (рис. 3.5 – 3.7).

```
osboxes@osboxes:~/dmatrixer/examples$ ./client.py
send file to decoding...
Result:
{"results":[{"message":"https://marking.com.ua/"},"message":"ABCDabcd"},"message":"http://www.labeljoy.com"},"message":"01048103470032221omczdcsv8vz8\u001d918023\u001d92YInk1wZffMEBZ7nNGQGccH/mMyjxqb6SQH3ebMXVZBw+HK+uHdACsiK6pmlv+2gtXQ+aXpw9y03iimVeDutgaw=="},"message":"18800000571920119916120010109364801000000128003896"}],"status":"succes"}
osboxes@osboxes:~/dmatrixer/examples$
```

Рисунок 3.5 – Результат декодування для файлу test-1.png

```
osboxes@osboxes:~/dmatrixer/examples$ ./client.py
send file to decoding...
Result:
{"results":[{"message":"0102900000224193219VCX8C*Ggkx!f 91003A 92k6IjG7kuq442ol8whtz3Fw2eMMYieWhHnyLt57EfIO+9ba67G2/uh1NDIqxL/muAHeuYdmLLnTJgkSIbMw0RgQ=="}],"status":"succes"}
osboxes@osboxes:~/dmatrixer/examples$
```

Рисунок 3.6 – Результат декодування для файлу test-2.png

```
osboxes@osboxes:~/dmatrixer/examples$ ./client.py
send file to decoding...
Result:
{"results":[{"message":"UBS United Barcode Systems"},"message":"UBS United Barcode Systems"}],"status":"succes"}
osboxes@osboxes:~/dmatrixer/examples$
```

Рисунок 3.7 – Результат декодування для файлу test-3.png

У процесі тестування даного сценарію була перевірена коректність декодування штрих-коду, а також точність і повнота результатів, що повертаються. Були використані різні тестові зображення з різними типами та розмірами штрих-кодів, щоб оцінити здатність сервісу працювати з різноманітними вхідними даними.

Результати тестування підтвердили, що сервіс успішно розпізнає та декодує штрих-коди, а формат JSON- об'єкта дозволяє зручно передавати результати клієнтським застосункам.

3.2 Сценарій 2. Виявлення та збереження штрих-кодів у відеопотоці

У другому сценарії використання сервісу було розроблено web -інтерфейс, який надає можливість реєструвати знайдені та дешифровані штрих-коди. Користувач може переглядати список знайдених штрих-кодів за певний час, отримуючи його за запитом HTTP .

Для тестування сервісу було проведено наступний процес. Спершу запускається сервер, на якому розгорнутий сервіс. Потім використовується програма на базі операційної системи Android - Astra Streaming Studio. Ця програма дозволяє використовувати смартфон як мережеву камеру або IP камеру. IP камера передає відеопотік у цифровому форматі через мережу Ethernet і має унікальну IP адресу для підключення до мережі.

Для під'єднання до камери використовується протокол RSTP, і програмі передається IP адреса та порт локальної мережі, до якої під'єднана мережева камера та комп'ютер із сервісом. Потім камера налаштовується таким чином, щоб вона знімала зображення зі штрих-кодами.

Програма, працюючи в автоматичному режимі, виявляє та відзначає знайдені штрих-коди на зображенні (рис. 3.8).



Рисунок 3.8 – Тестовий запуск сервісу з кількома штрих-кодами у кадрі

Потім результат декодування виводиться на веб-сторінку (рис. 3.9):

Info	Date time	Message
CAMERAS	① 2025-04-26 10:23:00	This is Data Matrix code example
📷 camera01	① 2025-04-26 10:23:02	Account Number: 123456896534567899
📷 camera02	① 2025-04-26 10:23:03	Account Number:34567987654321234567890098-6.dfrgth
	① 2025-04-26 10:23:05	Account Number:34908987345698-6dfghn_#ydfj
	① 2025-04-26 10:23:07	This is Data Matrix code example
	① 2025-04-26 10:23:10	Account Number: 123456896534567899
	① 2025-04-26 10:23:15	Account Number:34567987654321234567890098-6.dfrgth
	① 2025-04-26 10:23:17	Account Number:34908987345698-6dfghn_#ydfj

Рисунок 3.9 – Результат декодування штрих-кодів на web -сторінці

Таким чином, у процесі тестування сервісу було перевірено його

працездатність у спільній роботі з IP камерою та програмою Astra Streaming Studio. Результати декодування штрих-кодів були успішно відображені на web - сторінці, що підтверджує ефективність та точність сервісу. Були використані різні відеопотоки з різними умовами освітленості, кутами огляду та різними типами штрих-кодів. В процесі тестування були враховані такі фактори, як швидкість обробки відеопотоку, точність виявлення штрих-кодів та збереження результатів у сховищі SQLite. Результати тестування підтвердили ефективність роботи сервісу в автоматичному режимі, а формат об'єкта JSON дозволяє зручно передавати список знайдених штрих-кодів.

Тестування створеного сервісу з розпізнавання штрих-кодів DataMatrix у потоковому відео показало його працездатність та відповідність вимогам. Обидва сценарії використання успішно пройшли тестування, підтверджуючи функціональність та надійність сервісу. Перший сценарій демонструє можливість запити декодування штрих-коду за допомогою клієнтської програми, а другий сценарій показує здатність сервісу автоматично виявляти штрих-коди у відеопотоці та зберігати їх у базі даних SQLite. Розроблений сервіс надає зручний та ефективний спосіб розпізнавання та використання штрих-кодів DataMatrix, розширюючи можливості клієнтських програм та автоматизуючи процеси обробки даних.

Подальша робота може бути спрямована на оптимізацію та розширення функціональності сервісу, а також на ширше покриття тестами для забезпечення стабільності та надійності у різних сценаріях використання.

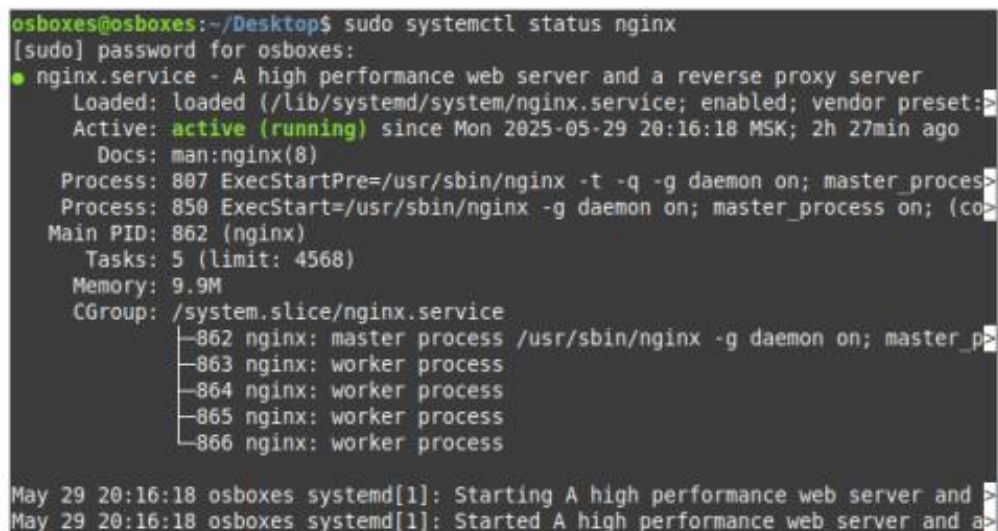
3.3 Тестування безпечної передачі даних

В рамках тестування сервісу для розпізнавання штрих-кодів DataMatrix було проведено додаткове тестування, яке спрямоване на забезпечення безпечної передачі даних. Для цього був використаний проксі - сервер Nginx та самопідписаний сертифікат.

3.3.1 Використання проксі-сервера Nginx

Nginx являє собою високопродуктивний web -сервер та проксі-сервер, створений для обробки HTTP -запитів та управління передачею даних між клієнтами та серверами. Він відрізняється легковагістю, масштабованістю і значною продуктивністю, саме це робить його прекрасним вибором для оптимізації та захисту створеного сервісу. Також він може перенаправляти запити web -сервера на інший web- сервер або програму, що дозволяє вам керувати трафіком і балансувати навантаження на web -сервери. Таким чином, застосування проксі-сервера Nginx дозволяє створити захищений канал зв'язку між клієнтами (користувачами) та сервером сервісу.

Для перевірки працездатності проксі-сервера Nginx було виконано такі дії. У консолі була виконана команда «`sudo systemctl status nginx`», і в результаті отримано повідомлення `active (running)`, що свідчить про те, що проксі-сервер активний і працює без помилок (рис. 3.10):



```
osboxes@osboxes:~/Desktop$ sudo systemctl status nginx
[sudo] password for osboxes:
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: ena
   Active: active (running) since Mon 2025-05-29 20:16:18 MSK; 2h 27min ago
     Docs: man:nginx(8)
   Process: 807 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process
   Process: 850 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (co
   Main PID: 862 (nginx)
    Tasks: 5 (limit: 4568)
   Memory: 9.9M
    CGroup: /system.slice/nginx.service
            └─862 nginx: master process /usr/sbin/nginx -g daemon on; master_p
              └─863 nginx: worker process
                └─864 nginx: worker process
                  └─865 nginx: worker process
                    └─866 nginx: worker process

May 29 20:16:18 osboxes systemd[1]: Starting A high performance web server and
May 29 20:16:18 osboxes systemd[1]: Started A high performance web server and a
```

Рисунок 3.10 – Поточний статус проксі-сервера

Далі в браузері було введено доменне ім'я для посилання на локальний комп'ютер `https://localhost`, що дозволило успішно відкрити web-інтерфейс сервісу (рис. 3.11):

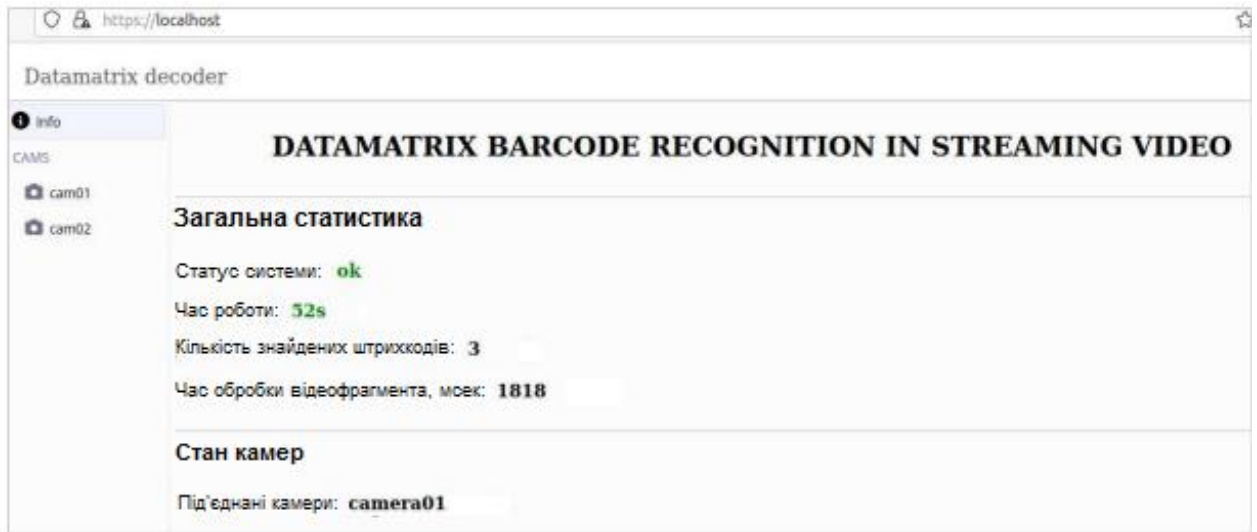


Рисунок 3.11 – Демонстрація роботи проксі-сервера

Це підтвердило, що проксі-сервер правильно налаштований та здатний обробляти запити клієнтів.

Таким чином, проведені дії підтверджують успішну установку та налаштування проксі-сервера Nginx, а також його здатність забезпечувати захищений канал зв'язку для сервісу.

3.3.2 Використання самопідписаного сертифіката

У криптографії самопідписаний SSL сертифікат є сертифікатом відкритого ключа, який був створений та підписаний тією самою особою, яку він ідентифікує. Він забезпечує безпечну передачу інформації в мережі як в Інтернеті, так і всередині мережі організації. Самопідписані SSL сертифікати часто використовуються для внутрішнього використання та ідеально підходять для розробки на локальному хостингу. Використання такого сертифіката забезпечує шифрування даних та захист від можливості заміни інформації третіми особами.

Самопідписаний сертифікат було створено за допомогою OpenSSL та встановлено на HTTPS-сервер. Далі було оновлено файл конфігурації Nginx для завантаження цього сертифіката.

При запуску сервісу та відкритті web-інтерфейсу в браузері користувачів з'являється попередження про те, що з'єднання не є захищеним, оскільки

самопідписаний сертифікат було згенеровано самостійно, без завірення довіреним центром сертифікації (рис. 3.12):

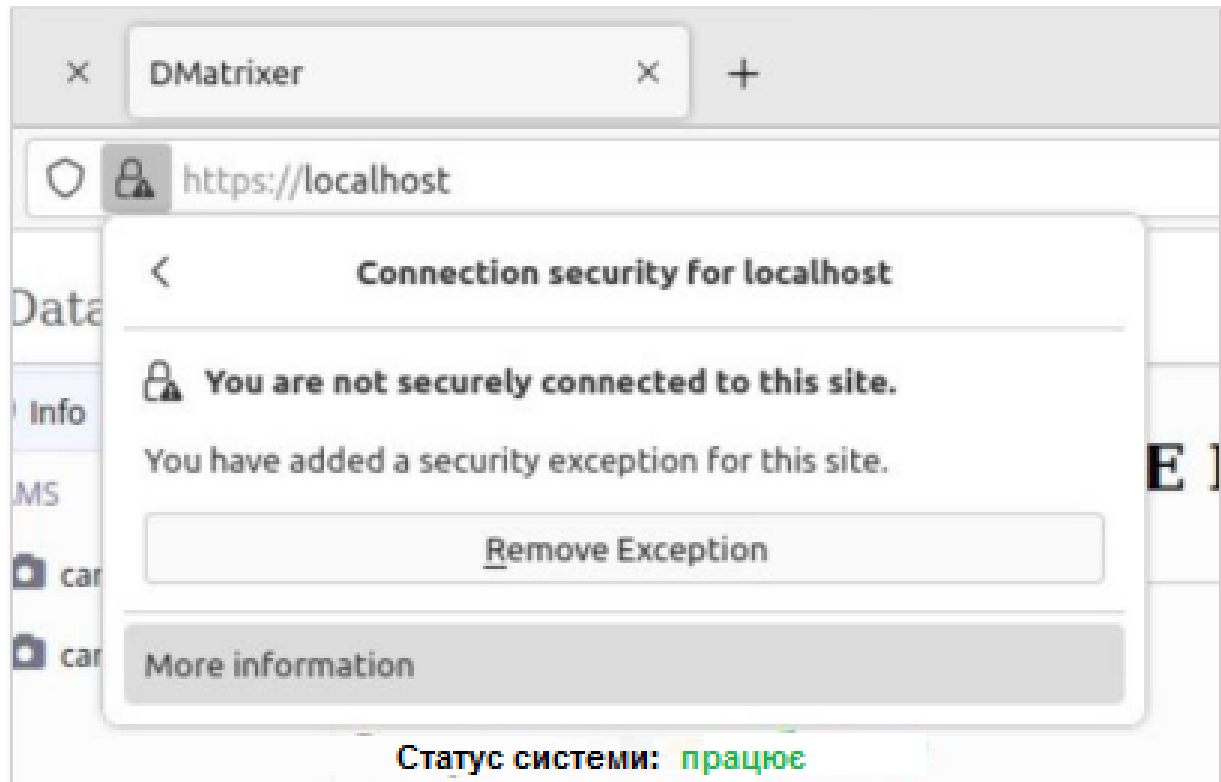


Рисунок 3.12 – Попередження про незахищеність з'єднання

Це пов'язано з тим, що цей сертифікат не забезпечує аутентифікацію сайту, тобто не підтверджує автентифікацію сервера.

Однак, незважаючи на відсутність запевнення, самопідписаний сертифікат все ж таки забезпечує шифрування трафіку між клієнтом і сервером за протоколом HTTPS. Це дає змогу захистити передані дані від НСД і підтримувати конфіденційність даних.

На рис. 3.13, 3.14 представлена інформація про створений сертифікат.

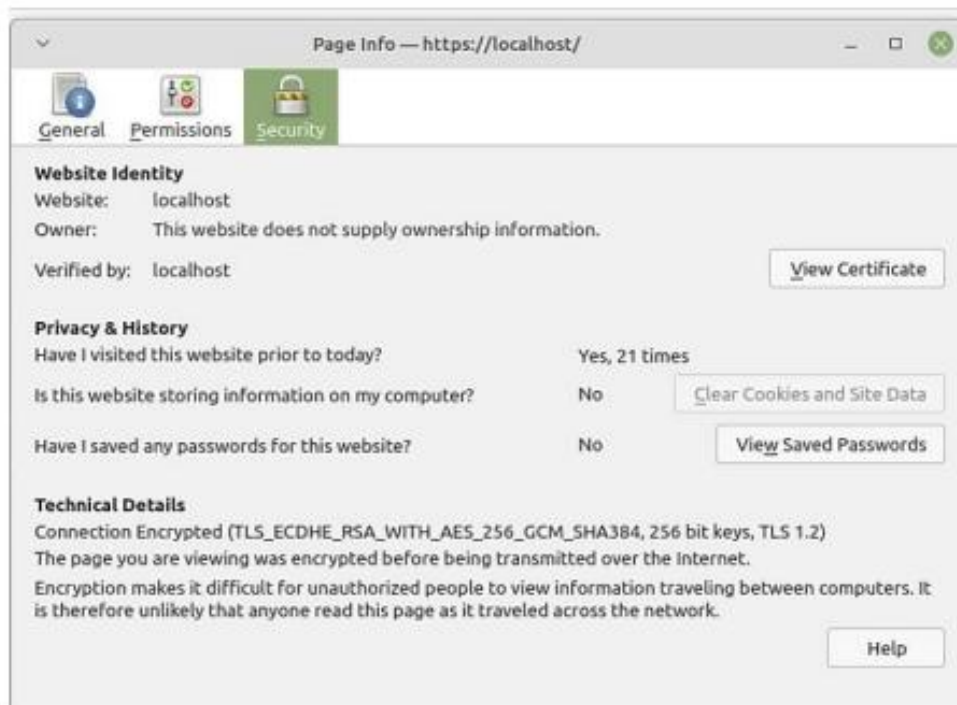


Рисунок 3.13 – Розділ «Security»



Рисунок 3.14 – Перегляд сертифікату

У контексті розробленого сервісу, який надає можливість розпізнавання та обробки штрих-кодів, захищене з'єднання є важливим аспектом.

Незважаючи на те, що самопідписаний сертифікат не забезпечує повну автентифікацію сервера, він гарантує шифрування даних, що є суттєвим для забезпечення безпеки взаємодії між клієнтом та сервером.

РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Навчання працюючих і інструктажі з охорони праці

Однією із складових ефективної роботи з профілактики виробничого травматизму є належна підготовка, навчання та підвищення кваліфікації працівників з питань охорони праці. Загальний порядок проведення навчання з питань охорони праці встановлений Законом України «Про охорону праці» (ст. 18. «Навчання з питань охорони праці»).

Виконання вимог Закону України «Про охорону праці» в частині проведення навчання та перевірки знань з питань охорони праці здійснюється відповідно до Типового положення про порядок проведення навчання і перевірки знань з питань охорони праці, затвердженого наказом Держкомітету України з нагляду за охороною праці 26 січня 2005 р. № 15 (далі — Типове положення).

Нагляд за дотриманням вимог Типового положення здійснюють органи державного нагляду за охороною праці, а координацію та методичний супровід — Головний навчально-методичний центр та навчальні підрозділи експертно-технічних центрів Держгірпромнагляду.

Вивчення предмета «Охорона праці» при підготовці, перепідготовці та підвищенні кваліфікації працівників, які залучаються до виконання робіт з підвищеною небезпекою, на підприємстві регламентується п. 2.3 Типового положення. На підприємствах згідно з п. 1.1 Додатку 3 Типового положення навчання та перевірку знань з питань охорони праці повинні проходити керівники, заступники керівників, головні спеціалісти, керівники основних виробничих та технічних служб, безпосередньо пов'язані з організацією безпечного ведення робіт. Крім цього, згідно з п. 5 Додатку 3, навчання та перевірку знань з питань охорони праці мають проходити керівники, спеціалісти служб охорони праці, члени комісій з перевірки знань з питань охорони праці, особи, відповідальні за технічний стан і безпечну експлуатацію об'єктів підвищеної небезпеки підприємств.

Типове положення встановлює порядок та місце проведення навчання та перевірки знань з питань охорони праці посадових осіб (п. 5.2 та п. 5.3). Посадові особи, перелік яких наведено в п. 5.2, проходять навчання у Головному навчально-методичному центрі Держнаглядохоронпраці. Перевірка знань цієї категорії посадових осіб проводиться комісією, створеною наказом Держгірпромнагляду.

Організацію навчання та перевірки знань з питань охорони праці працівників на підприємстві здійснюють працівники служби кадрів або інші спеціалісти, яким роботодавець доручив організацію цієї роботи. Навчання та перевірка знань з питань охорони праці працівників (виконавців і посадових осіб), які не залучаються до виконання робіт підвищеної небезпеки, проводиться не рідше ніж один раз на три роки. Посадові особи та працівники, які виконують роботи підвищеної небезпеки, проходять спеціальне навчання та перевірку знань відповідних нормативно-правових актів з охорони праці не рідше одного разу на рік.

Посадові особи малих підприємств (п. 5.4), які не мають можливості створити власні комісії з перевірки знань з питань охорони праці та провести навчання з питань охорони праці, проходять навчання та перевірку знань в навчальних закладах, які мають відповідний дозвіл.

Спеціальне навчання з питань охорони праці може проводитись безпосередньо на підприємстві або навчальним закладом, який має відповідний дозвіл. При проведенні такого навчання на підприємстві навчальні плани та програми розробляються з урахуванням конкретних видів робіт, виробничих умов і функціональних обов'язків працівників і затверджуються наказом керівника підприємства.

Періодичність інструктажів, навчання та перевірки знань з питань охорони праці залежить від видів виконуваних робіт та встановлюється Типовим положенням. Перевірка знань з питань охорони праці після проведення спеціального навчання проводиться комісією підприємства.

Якщо на підприємстві неможливо створити комісію з перевірки знань з питань охорони праці (п. 4.4 Типового положення), перевірка знань проводиться комісією спорідненого підприємства або Теруправління Держгірпромнагляду.

Всі працівники та посадові особи підприємства, включаючи посадових осіб, відповідальних за виконання робіт підвищеної небезпеки (крім зазначених в п. 5.2 та п. 5.3 Типового положення), проходять навчання та перевірку знань з питань охорони праці на підприємстві. Типове положення не зобов'язує, але й не забороняє проводити навчання всіх виконавців та посадових осіб (особливо тих, що виконують роботи підвищеної небезпеки) в навчальних закладах. У нашій країні є багато підприємств, де таке навчання проводиться, і це має позитивні наслідки. Ті витрати, які при цьому несуть підприємства, перекриваються створенням більш безпечних умов праці і в результаті збереженням життя та здоров'я працівників.

Також в навчальних закладах проходять навчання та перевірку знань із загальних питань охорони праці всі посадові особи та фахівці, які проводять інструктажі або навчання підлеглих працівників з питань охорони праці, виконують роботи з проектування об'єктів, а також інші працівники, незалежно від того, передбачено таке навчання Типовим положенням чи ні.

4.2 Санітарно-гігієнічні вимоги до умов праці

На робочих місцях працівників, які відповідальні за експлуатацію сервісу управління механізмом авторських прав на мультимедійні файли, необхідно забезпечити дотримання вимог, затверджених Наказом Мінсоцполітики від 14.02.2018 за № 207 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями».

Приміщення для роботи з ЕОМ мають бути обладнані системами опалення, кондиціонування повітря, або припливно-витяжною вентиляцією. У приміщеннях на робочих місцях мають забезпечуватись оптимальні значення параметрів мікроклімату: температури, відносної вологості та рухливості повітря відповідно до норм та правил, а також ДБН В.2.5-67:2013 «Опалення, вентиляція та кондиціонування», затверджених наказом Мінрегіону від 25.01.2013 р. № 24. Відповідно до санітарних норм мікроклімату виробничих приміщень ДСН 3.3.6.042-99 в офісних приміщеннях, обладнаних ЕОМ, температура повітря

повинна становити 22-25°C, відносна вологість повітря – 40-60 %, швидкість руху повітря – не більше 0,1 м/с [16].

Приміщення, призначені для роботи з ЕОМ, повинні мати природне освітлення. У виробничих приміщеннях, обладнаних ЕОМ, необхідно створити належне освітлення. При експлуатації сервісу управління механізмом авторських прав на мультимедійні файли, важливим, з точки зору охорони праці, є забезпечення достатньої величини природного та штучного освітлення, які визначені у НПАОП 0.00-7.15-18 [17]. Природне світло повинно бути бічним, зорієнтованим, як правило, на північ чи північний схід, і забезпечувати коефіцієнт природної освітленості не нижче 1,5%. При виробничій потребі дозволяється експлуатувати ЕОМ у приміщеннях без природного освітлення за узгодженням з органами Держпромгірнагляду та органами й установами санітарно-епідеміологічної служби. Вікна приміщень повинні мати регулювальні пристрої для відчинення, а також жалюзі, штори тощо. Штучне освітлення приміщення з робочими місцями, обладнаними відеотерміналами ЕОМ загального та персонального користування, має бути всеосяжним і рівномірним. У випадку, коли переважають роботи з документами, допускається комбіноване освітлення (додатково до загального освітлення встановлюється світильники місцевого освітлення). Світильники розміщуються збоку від робочих місць (переважно ліворуч), або локально над робочим місцем (при розташуванні відеотерміналів ЕОМ за периметром приміщення). Як джерело світла при штучному освітленні застосовуються, як правило, люмінесцентні лампи. У світильниках місцевого освітлення допускається застосування ламп розжарювання. Рівень освітленості на робочому місці має становити 300-500 лк. При використанні комбінованого освітлення не допускається відблисків на поверхні екрана та збільшення освітлення екрана вище 300 лк.

Орієнтація вікон повинна бути на північ або північний схід, вікна повинні мати жалюзі, які можна регулювати, або штори; не дозволяється розмішувати кабінети обчислювальної техніки у підвальних приміщеннях будинків; кабінети, обладнані комп'ютерною технікою, в навчальних закладах повинні розмішуватись в окремих приміщеннях з природним освітленням та організованим обміном

повітря; стіни, стеля і підлога та обладнання кабінетів комп'ютерної техніки повинні мати покриття із матеріалів з матовою фактурою з коефіцієнтом відбиття: стін — 40- 50 %, стелі — 70 - 80 %, підлоги — 20-30 %, предметів обладнання — 40-60 % (робочого столу — 40-50 %, корпусу дисплею та клавіатури — 30-50 %, стелажів — 40-60 %); поверхня підлоги повинна мати антистатичне покриття та бути зручною для вологого прибирання; забороняється використовувати для оздоблення інтер'єру приміщень комп'ютерних кабінетів полімерні матеріали (дерев'яно-стружкові плити, шпалери, що придатні для миття, плівкові та рулонні синтетичні матеріали, шаровий паперовий пластик та ін.), що виділяють у повітря шкідливі хімічні речовини, які перевищують гранично допустимі концентрації; вміст шкідливих хімічних речовин в повітрі дошкільних та учбових приміщень з комп'ютерною технікою не повинен перевищувати середньодобові концентрації [17].

Організація робочого місця фахівця із експлуатації сервісу повинна забезпечувати відповідність усіх елементів робочого місця та їх розташування ергономічним вимогам ДСТУ 8604:2015 «Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги».

Відстань від екрана до ока фахівців, які працюють за комп'ютером визначається згідно з вимогами ДСанПіН 3.3.2.007-98.

Рівень шуму не повинний перевищувати: на місцях, де працюють програмісти та оператори ЕОМ, 55 дБА, у лабораторіях, де складаються алгоритми та ведеться робота з документацією – 60 дБА, у машинному залі – 65 дБА, у приміщеннях, де розміщені гучні агрегати обчислювальних машин – 75 дБА.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи було створено сервіс для розпізнавання та реєстрації штрих-кодів DataMatrix у потоковому відео. Було досягнуто наступних результатів:

- створено ефективний алгоритм, який здатний коректно розпізнавати штрих-коди DataMatrix у різних умовах, включаючи різні кути огляду, освітлення та якість відео;
- вивчено та проаналізовано різні засоби та інструменти, необхідні для розробки сервісу виявлення та розпізнавання штрих-кодів. Це включає бібліотеку комп'ютерного зору OpenCV, алгоритми обробки зображень і середовище розробки Qt;
- спроектовано архітектуру сервісу та розроблено модулі розпізнавання штрих-кодів та відеозахоплення;
- розроблено web-інтерфейс, що дозволяє користувачам зареєструвати штрих-коди та отримати доступ до відповідної інформації про товари або об'єкти;
- проведено тести та налагодження сервісу для забезпечення його коректної роботи. Це включає перевірку функціональності, стабільності та точності розпізнавання штрих-кодів. Також сервіс був протестований з використанням проксі-сервера та самопідписаного сертифікату для забезпечення безпечної передачі. Було перевірено, що проксі-сервер працює коректно та створює захищений канал зв'язку між клієнтом та сервером.

Сервіс забезпечує швидке та точне зчитування та обробку закодованої інформації на продуктах, що призводить до значного прискорення процесу ідентифікації товарів на виробництві. Завдяки високій швидкості роботи та точності розпізнавання, даний сервіс скорочує ймовірність помилок під час ідентифікації продуктів, забезпечуючи надійність та точність у процесі зчитування штрих-кодів.

Після аналізу кожного етапу виконання кваліфікаційної роботи та розгляду поставлених завдань можна зробити висновок про досягнення поставленої мети.

ПЕРЕЛІК ДЖЕРЕЛ

1. ЗЧИТУВАННЯ КОДІВ [Електронний ресурс] – Режим доступу: https://machinevisionukraine.com/code_reading_application/ (Дата звернення: 18.03.2025).
2. Правила та види маркування продукції [Електронний ресурс] – Режим доступу: <https://ukrstandart.net/korysna-informatsiia/pravy-la-ta-vydy-markuvannia-produktsii> (Дата звернення: 28.03.2025).
3. Data Matrix [Електронний ресурс] – Режим доступу: <https://www.vostok.dp.ua/ukr/infa1/glossary/data-matrix/> (Дата звернення: 28.03.2025).
4. Chung, F. Spectral Graph Theory, AMS. Providence, RI. DeValois, R. and DeValois, K. Spatial Vision. Oxford University Press. New York, N.Y.
5. У чому різниця між кодами Data Matrix і QR-кодом/ [Електронний ресурс] – Режим доступу: <https://dou.ua/forums/topic/48845/> (Дата звернення: 02.04.2025).
6. Dita, I. Using mean shift algorithm in the recognition of industrial data matrix codes / I. Dita, V. Gui, M. Ottesteanu, F. Quint // SITE'12 Proceedings of the 11th international conference on Telecommunications and Informatics, Proceedings of the 11th international conference on Signal Processing. – Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS) Stevens Point, 2012. – P. 174-179.
7. Dita, I. A scanning method for industrial data matrix codes marked on spherical surfaces / I. Dita, V. Gui, M. Ottesteanu, F. Quint // SITE'12 Proceedings of the 11th international conference on Telecommunications and Informatics, Proceedings of the 11th international conference on Signal Processing. – Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS) Stevens Point, 2012. – P. 38-42.
8. L. Karrach, E. Pivarčiová. “Options to use data matrix codes in production engineering”. Management Systems in Production Engineering, vol. 26 (4), pp. 231-236, 2018.
9. Karrach Ladislav, Pivarčiová Elena. Recognition of Data Matrix Codes in

Images and their Applications in Production Processes. // Management Systems in Production Engineering, Vol. 28, no. 3. pp. 154 – 161. 2020. DOI <https://doi.org/10.2478/mspe-2020-0023>.

10. Gaitatzis, T. Learn REST APIs: Your guide to how to find, learn, and connect to the REST APIs that powers the Internet of Things revolution / T. Gaitatzis. – Ottawa: BackupBrain Press, 2019. 109 p. ISBN 1989775004.

11. Основні підходи до ефективного тестування ПЗ [Електронний ресурс] – Режим доступу: <https://dou.ua/forums/topic/48845/> (Дата звернення: 04.04.2025).

12. Bodnarchuk, I., Skorenkyu, Y., Kramar, T., Duda, O., & Nykytyuk, V. (2022). Use of Analytical Hierarchy Process in Scenarios Design for a Digital Museum with XR components. ITTAP, 414–425.

13. Dozorskyi, V., Dediv, I., Sverstiuk, S., Nykytyuk, V., & Karnaukhov, A. (2023). The Method of Commands Identification to Voice Control of the Electric Wheelchair.

14. Koroliuk, R., Nykytyuk, V., Tymoshchuk, V., Soyka, V., & Tymoshchuk, D. (2025). Automated monitoring of bee colony movement in the hive during winter season.

15. Zagorodna, N., Skorenkyu, Y., Kunanets, N., Baran, I., Stadnyk, M. Augmented Reality Enhanced Learning Tools Development for Cybersecurity Major. CEUR Workshop Proceedings., 2022, 3309, pp. 25–32

16. Заїкіна Д., Глива В. Основи охорони праці та безпека життєдіяльності. 2019. URL: <https://doi.org/10.31435/rsglobal/001> (дата звернення: 14.04.2025).

17. Безпека в надзвичайних ситуаціях. Методичний посібник для здобувачів освітнього ступеня «магістр» всіх спеціальностей денної та заочної (дистанційної) форм навчання / укл.: Стручок В. С. Тернопіль: ФОП Паляниця В. А., 2022. 156 с.

ДОДАТКИ

Додаток А – Вихідний код модуля розпізнавання

Файл qdmatrixer.h.

```
#ifndef QMDECODER_H
#define QMDECODER_H
#include <QtCore>
#include <QImage>
#include <opencv2/opencv.hpp>
class QmDecoderResult; class QmDecoder {
public:
static int decode(const QImage&img, QVector<QmDecoderResult>
    &results, int stop after n = 1, long stop after timeout =
    200);
static int decode(const cv::Mat cvframe,
    QVector<QmDecoderResult>& results, int stop after n = 1, long
    stop after timeout = 200);
};
class QmDecoderResult {
public:
QmDecoderResult(const QString&data, const QPolygon& polygon =
    QPolygon())
: mMessage(data), mPolygon(polygon)
{ }
~QmDecoderResult() { }
const QString& message() const { return mMessage; } const
    QPolygon& polygon() const { return mPolygon; }
QJsonObject toJson() const {
QJsonObject jobj;
jobj.insert("message", mMessage); return jobj;
}
private:
QString mMessage;
QPolygon mPolygon;
};
#endif
```

файл qdmatrixerhttp.h

```
#ifndef QDMATRIXERHTTP_H
#define QDMATRIXERHTTP_H
#include <QtCore>
#include "xhttpserver/qxhttpserver.h"
class QDMatrixerHttp : public QxHttpServer {
Q_OBJECT
//static QString makeWebappEndpointPattern(const
    QString&endpoint); public:
explicit QDMatrixerHttp(QObject* parent = nullptr);
~QDMatrixerHttp() ;
const QString& docRoot() const { return mDocRoot; } void
    setDocRoot(const QString&value) { mDocRoot = value; }
private:
QString makeDocumentFullPath(const QString&doc) const; void
    initEndpoints(); private:
QString mDocRoot{"./docroot"}; };
#endif
```

Файл main.cpp

```
//#include <QCoreApplication>
#include <QApplication>
#include "qdmatrixer.h"
int main(int argc, char *argv[])
{
    // qputenv("OPENCV_FFMPEG_CAPTURE_OPTIONS",
    "rtsp flags;prefer tcp");
    QThreadPool::globalInstance()->setMaxThreadCount(32); qDebug()
        << QThreadPool::globalInstance()->maxThreadCount();
    qRegisterMetaType<qintptr>("qintptr") ;
    qRegisterMetaType<qintptr>("qintptr");
    //QCoreApplication a(argc, argv);
    QApplication a(argc, argv);
    std::unique_ptr<QDMatrixer> dmatrixer(new QDMatrixer());
    dmatrixer->start() ;
    return a.exec(); }
```