

Міністерство освіти і науки України  
Національний технічний університет  
«Дніпровська політехніка»

Інститут електроенергетики

(інститут)

факультет інформаційних технологій

(факультет)

Кафедра інформаційних технологій та комп'ютерної інженерії

(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА

кваліфікаційної роботи ступеня бакалавра  
(бакалавра, спеціаліста, магістра)

студента Лисунова Анатолія Олександровича

(ПІБ)

академічної групи 126-19-1

(шифр)

спеціальності 126 «Інформаційні системи та технології»  
(код і назва спеціальності)

за освітньо-професійною програмою  
(за наявності)

«Інформаційні системи та технології»

(офіційна назва)

на тему Розробка застосунку для розпізнавання номерів автотранспорту з використанням бібліотеки OpenCV

(назва за наказом ректора)

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	проф. Олевський В.І.			
розділів:				
Рецензент				
Нормоконтролер	проф. Коротенко Г.М.			

Дніпро  
2023

**ЗАТВЕРДЖЕНО:**

завідувач кафедри

інформаційних технологій

та комп'ютерної інженерії

(повна назва)

Гнатушенко В.В.

(підпис)

(прізвище, ініціали)

« \_\_\_\_\_ » \_\_\_\_\_ 2023 року

**ЗАВДАННЯ**  
**на кваліфікаційну роботу**  
**ступеня бакалавра**  
(бакалавра, спеціаліста, магістра)

студенту Лисунову А.О. академічної групи 126-19-1  
(прізвище та ініціали) (шифр)

спеціальності 126 «Інформаційні системи та технології»

за освітньою-професійною програмою \_\_\_\_\_

(за наявності)

«Інформаційні системи та технології»

на тему Розробка застосунку для розпізнавання номерів автотранспорту з використанням бібліотеки OpenCV

затверджену наказом ректора НТУ «Дніпровська політехніка» від 16.05.2023 р. № 350-с

Розділ	Зміст	Термін виконання
Розділ 1	Аналіз теми та постановка задачі	01.02.2023 – 28.02.2023
Розділ 2	Огляд методів автоматизованного управління транспортними потоками	01.03.2023 – 30.04.2023
Розділ 3	Експериментальна частина. Підготовка матеріалів для захисту роботи	01.05.2023 – 04.06.2023

Завдання видано \_\_\_\_\_

(підпис керівника)

В.І. Олевський

(прізвище, ініціали)

Дата видачі 16.05.2023 р.

Дата подання до екзаменаційної комісії 08.06.2023 р.

Прийнято до виконання \_\_\_\_\_

(підпис студента)

Лисунов А.О.

(прізвище, ініціали)

## РЕФЕРАТ

**Пояснювальна записка:** 64 стор., 24 рис., 1 додатки, 12 джерел.

**Об'єкт розробки:** застосунок для розпізнавання номерів автотранспорту з використанням бібліотеки OpenCV.

**Мета кваліфікаційної роботи:** розробка застосунку для розпізнавання номерів автотранспорту з використанням бібліотеки OpenCV.

У вступі наведено стан проблеми та обґрунтована її актуальність.

В першому розділі кваліфікаційної роботи були розглянуті принципи роботи інформаційних систем розпізнавання та ідентифікації зображень номерів транспортних засобів. Ці системи базуються на використанні спеціалізованих алгоритмів, які дозволяють автоматично виявляти та розпізнавати номерні знаки на зображеннях.

У другому розділі був проведений аналіз і вибір інструментів розпізнавання автомобільних номерів.

У третьому розділі була проведена розробка та перевірка ефективності системи пошуку автомобільних.

Практичне значення роботи полягає у розробці застосунку для розпізнавання номерів автотранспорту з використанням бібліотеки OpenCV, який дозволив би ефективно розпізнавати і ідентифікувати номерні знаки транспортних засобів.

**Список ключових слів:** НОМЕР АВТОТРАНСПОРТУ, РОЗПІЗНАВАННЯ, OPENCV, ІНФОРМАЦІЙНА СИТЕМА, HTML, CSS, MYSQL, PHP, UML

## ABSTRACT

**Explanatory note:** 64 pages, 24 figures, 1 appendices, 12 sources.

**The object of development:** an application for recognizing license plates using the OpenCV library.

**The purpose of the thesis:** is to develop an application for recognizing license plates using the OpenCV library.

The introduction describes the state of the problem and justifies its relevance.

In the first chapter of the qualification work, the principles of operation of information systems for recognizing and identifying images of vehicle license plates were considered. These systems are based on the use of specialized algorithms that allow to automatically detect and recognize license plates in images.

The second section analyzed and selected license plate recognition tools.

In the third section, we developed and tested the effectiveness of the car license plate search system.

The practical significance of the work is to develop an application for license plate recognition using the OpenCV library, which would allow for effective recognition and identification of vehicle license plates.

**Keywords:** LICENSE PLATE, RECOGNITION, OPENCV, INFORMATION SYSTEM, HTML, CSS, MYSQL, PHP, UML

## ЗМІСТ

<b>РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ</b> .....	8
1.1 Ознайомлення із системами розпізнавання номерних знаків .....	8
1.2 Труднощі при ідентифікації номерних знаків автомобілів .....	12
1.3 Використання систем розпізнавання номерних знаків.....	15
1.4 Висновок до першого розділу.....	20
<b>РОЗДІЛ 2 АНАЛІЗ ТА ВИБІР ІНСТРУМЕНТІВ РОЗПІЗНАВАННЯ АВТОМОБІЛЬНИХ НОМЕРІВ</b> .....	22
2.1 Технологія розпізнавання автомобільних номерів OCR .....	22
2.2 Технологія обробки зображень та комп'ютерного зору .....	25
2.2.1 Пре-обробка зображення: .....	25
2.2.2 Сегментація зображення:.....	26
2.2.3 Виділення ознак номерного знаку:.....	26
2.2.4 Класифікація та розпізнавання: .....	26
2.3 Технології розпізнавання зображень номерних знаків в існуючих системах .....	28
2.3.1 HikVision .....	28
2.3.2 Axis .....	29
2.3.3 Трал-Паркінг.....	30
2.3.4 Номер.....	31
2.3.5 Модуль iPera EX-LPR .....	33
2.3.6 АТАРУ ANPR SDK.....	34
2.4 Вибір OpenCV інструменту для розпізнавання автомобільних номерів .....	35
<b>РОЗДІЛ 3 РОЗРОБКА ТА ПЕРЕВІРКА ЕФЕКТИВНОСТІ СИСТЕМИ ПОШУКУ АВТОМОБІЛЬНИХ ЗНАКІВ</b> .....	38
3.1 Інтеграція OpenCV бібліотеки комп'ютерного зору в систему.....	38
3.2 Розробка архітектури пошуку номерних знаків авто .....	40
3.3 Розробка баз даних.....	48
3.3.1 База даних "users.db" .....	49
3.3.1 База даних "results.db":.....	50
3.4 Розробка та дослідження програмного застосунку.....	52
<b>ВИСНОВКИ</b> .....	60
<b>ПЕРЕЛІК ПОСИЛАНЬ</b> .....	61
<b>Додаток А. Фрагмент лістингу програми</b> .....	62
<b>Додаток Б. Відгук керівника кваліфікаційної роботи</b> .....	Помилка! Закладку не визначено.
<b>Додаток В. Рецензія</b> .....	Помилка! Закладку не визначено.

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ**

OCR - Optical Character Recognition;

БД – база даних;

СУБД – система управління базами даних;

MSE - (англ. Mean Square Error) – середньоквадратична помилка.

HTML - (англ. Hyper Text Markup Language) – мова розмітки гіпертексту.

API - (англ. Application Programming Interface) – прикладний програмний інтерфейс.

HTTP - (англ. Hyper Text Transfer Protocol) – протокол передачі гіпертекстових документів.

## ВСТУП

**Актуальність роботи.** Зі зростанням кількості транспортних засобів на дорогах збільшується потреба в автоматичному регулюванні та контролі. Системи автоматичного розпізнавання та ідентифікації номерних знаків грають важливу роль у сучасних дорожніх системах, дозволяючи розпізнавати та ідентифікувати автотранспорт без необхідності прямої участі людини.

Однак, незважаючи на значний прогрес у цій галузі, все ще існує потреба в покращенні технічних та алгоритмічних аспектів систем автоматичного розпізнавання номерних знаків. Оскільки обробка зображень та розпізнавання варіативних номерних знаків можуть бути складними завданнями, важливо продовжувати вдосконалювати технології, зокрема застосовувати сучасні методи нейронних мереж та машинного навчання.

Розробка інформаційних систем автоматичного розпізнавання та ідентифікації номерів транспортних засобів з використанням таких передових технологій є актуальним напрямом наукових досліджень. Це дозволить поліпшити ефективність систем контролю та безпеки на дорогах, спростити процеси регулювання дорожнього руху і сприяти загальному зростанню якості транспортної інфраструктури.

**Об'єктом дослідження** є застосунок для розпізнавання номерів автотранспорту з використанням бібліотеки OpenCV.

**Метою роботи** є Розробка застосунку для розпізнавання номерів автотранспорту з використанням бібліотеки OpenCV, для полегшення прийняття рішення потенційним клієнтом.

## РОЗДІЛ 1

### АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

#### 1.1 Ознайомлення із системами розпізнавання номерних знаків

Автоматичне розпізнавання та ідентифікація зображень номерів транспортних засобів є важливою технологією, яка використовує оптичне розпізнавання символів для зчитування номерних знаків автомобілів і отримання даних про їхнє місцезнаходження. В сучасному світі, системи розпізнавання та ідентифікації номерних знаків широко використовуються правоохоронними органами для різних адміністративних цілей, наприклад, для перевірки реєстрації транспортних засобів або наявності дійсної ліцензії.

Одним з прикладів такої системи є Eyedea, яка є високоточним програмним забезпеченням для автоматичного розпізнавання номерних знаків. Вона базується на функціях штучного інтелекту, які навчені розпізнавати номерні знаки. Ця система може виявляти та зчитувати всі номерні знаки на зображенні, а також розпізнавати спеціальні символи та коди країн. Вона підтримує як однорядкові, так і багаторядкові номерні знаки, а також номерні таблички вантажівок. Деякі номерні знаки можуть мати різні графічні елементи на початку, в середині або в кінці. Деякі знаки можуть використовувати незвичні шрифти або унікальні формати. Основними перевагами цієї системи автоматичного розпізнавання номерів автотранспорту є здатність розпізнавати широкий спектр форматів номерних знаків та зчитувати всі видимі символи на зображенні.

Такі системи дозволяють значно полегшити процес ідентифікації номерних знаків та автоматизувати збір даних про автомобілі. Основні переваги використання автоматичного розпізнавання номерів автотранспорту включають:

1. Ефективність і швидкість: Системи автоматичного розпізнавання здатні швидко зчитувати та ідентифікувати номерні знаки без необхідності вручну вводити їх. Це дозволяє економити час і зусилля,



особливо при масовому перевірці автомобілів на дорозі або на парковках.

2. Висока точність: Сучасні системи розпізнавання номерів здатні до високої точності визначати символи на номерних знаках. Вони враховують різні шрифти, графічні елементи та унікальні формати номерних знаків, що дозволяє їм ефективно працювати з різноманітними видами номерів.
3. Автоматизація процесу: Системи автоматичного розпізнавання забезпечують автоматизований збір даних про автомобілі, такі як номери знаків і місцезнаходження. Це дозволяє легко вести облік транспортних засобів, перевіряти їх реєстрацію та виконувати різні адміністративні процедури.
4. Застосування в правоохоронних органах: Системи автоматичного розпізнавання номерів широко використовуються правоохоронними органами для перевірки реєстрації автомобілів, виявлення викрадених або шуканих транспортних засобів, а також для збирання доказів у розслідуваннях.
5. Покращення безпеки: Використання систем розпізнавання номерів дозволяє забезпечити більшу безпеку на дорогах. Ці системи можуть виявляти порушення правил дорожнього руху, такі як перевищення швидкості або проїзд на червоне світло, і автоматично фіксувати такі порушення. Вони можуть також служити важливим інструментом для виявлення автомобілів, пов'язаних зі злочинністю, або для моніторингу дорожнього руху на потенційно небезпечних ділянках.
6. Підтримка логістичних та транспортних систем: Системи автоматичного розпізнавання номерів можуть бути використані для покращення логістичних та транспортних систем. Наприклад, вони можуть автоматично відстежувати рух транспортних засобів на складах або в логістичних центрах, сприяючи ефективному управлінню і складанню маршрутів.

7. Застосування у системах безпеки паркування: Системи автоматичного розпізнавання номерів також використовуються для контролю та безпеки паркування. Вони можуть автоматично реєструвати номерні знаки автомобілів, які в'їжджають на парковку, і відстежувати їх час перебування. Це дозволяє забезпечити безпеку паркування, виявити порушення правил паркування та забезпечити ефективне використання паркінгових місць.

В цілому, автоматичне розпізнавання та ідентифікація зображень номерів транспортних засобів є потужним інструментом, який сприяє покращенню ефективності, безпеки та управління автомобільним транспортом.

Так, Plate Recognizer є ще одним прикладом автоматичної системи ідентифікації транспортних засобів, яка використовує нейронні мережі для розпізнавання та розшифрування номерних знаків. Ця система має кілька особливостей, які сприяють її ефективності та гнучкості:

1. Налаштування під конкретні потреби: Plate Recognizer може бути налаштований для роботи під різними умовами, такими як кут камери або рівень освітлення. Це дозволяє досягати кращих результатів розпізнавання в конкретних умовах спостереження.
2. Використання двох нейронних мереж: Система використовує дві нейронні мережі - одна для ідентифікації номерних знаків на зображенні, а інша для розшифрування кожного символу номерного знаку. Це дозволяє досягати високої точності розпізнавання.
3. Покращення алгоритмів: Plate Recognizer постійно вдосконалює свої алгоритми для роботи з різними факторами, такими як сонячні відблиски, розмите зображення, швидкість руху транспортного засобу, нічний час тощо. Це дозволяє системі бути більш універсальною та ефективною незалежно від умов зйомки.

4. Робота в реальному часі: Система може працювати в режимі реального часу, що дозволяє швидко розпізнавати номерні знаки транспортних засобів під час спостереження на дорозі або на парковках.

Завдяки цим особливостям Plate Recognizer є потужним інструментом для автоматичної ідентифікації номерних знаків транспортних засобів у різних умовах.

Першу систему автоматичного розпізнання зображень номерів транспортних засобів було винайдено у 1976 році у відділі наукового розвитку поліції у Великобританії. Системи ідентифікації номерних знаків використовують різноманітні методи обробки зображень для виявлення, нормалізації та покращення зображень номерних знаків. Після цього застосовується оптичне розпізнавання символів для виділення текстової інформації з номерних знаків, [1].

Розвиток систем розпізнавання номерних знаків вплинув на дорожнє регулювання у світі. Наприклад, в нідерландських номерних знаках змінився шрифт у 2002 році, щоб полегшити їх розпізнавання системами автоматичного розпізнавання. Кожна країна може мати власний стиль і формат номерних знаків, і системи автоматичного розпізнавання повинні враховувати ці відмінності, щоб бути ефективними.

Зараз багато програм систем автоматичного розпізнавання номерних знаків індивідуально адаптовані для потреб кожної країни. Це включає розпізнавання різних розмірів шрифтів, розташування тексту та міжнародних варіантів номерних знаків.

Ці системи є важливим інструментом для поліції та інших правоохоронних органів, оскільки вони допомагають виявляти та ідентифікувати транспортні засоби, перевіряти їх статус та дотримання правил дорожнього руху.

Так, для систем автоматичного розпізнавання та ідентифікації зображень номерних знаків транспортних засобів можуть використовуватися різні типи камер, які включають камери контролю за дотриманням правил дорожнього руху, камери замкнутого телебачення та мобільні пристрої, що прикріплюються до транспортних засобів.

Деякі системи можуть використовувати інфрачервоні камери для отримання чіткіших зображень номерних знаків, особливо в умовах обмеженого освітлення.

Розташування камер є важливим етапом при встановленні системи ідентифікації номерних знаків. Необхідно ретельно визначити кількість камер і їх точне розташування для досягнення оптимальних результатів. Однак це може бути складно з урахуванням різноманітних перешкод та унікальності середовища.

Для регулювання дорожнього руху потрібні камери, які можуть охоплювати кілька смуг і здатні зчитувати номерні знаки на великій швидкості. У випадку зйомки номерних знаків на припаркованих автомобілях можуть використовуватися камери з меншим радіусом дії та меншою фокусною відстанню. Для парковок з перпендикулярними місцями для автомобілів можуть знадобитися спеціальні камери з короткою фокусною відстанню.

Для держав, де використовуються задні номерні знаки, може виникнути додаткова проблема, оскільки передня камера може бути неефективною при зустрічному русі. В таких випадках одна з камер може бути повернута назад, щоб забезпечити ефективну ідентифікацію номерних знаків.

## **1.2 Труднощі при ідентифікації номерних знаків автомобілів**

Так, власники транспортних засобів іноді застосовують різні методи для ухилення від систем автоматичного розпізнавання та ідентифікації номерних

знаків та камер контролю за дотриманням правил дорожнього руху загалом. Один з таких методів полягає у використанні засобів, які сприяють зменшенню відбивання світла від номерного знаку, що ускладнює визначення його місцезнаходження або створення достатньо високого контрасту для зчитування.

Наприклад, деякі власники автомобілів можуть встановлювати прозорі кришки на номерний знак або використовувати спеціальні спреї, які призначені для зниження відбиття світла, [2]. Однак, варто відзначити, що в більшості юрисдикцій використання таких засобів є незаконним і суперечить чинним законам. У деяких країнах встановлено законодавство, що забороняє використання таких пристосувань для номерних знаків.

Таким чином, хоча існують спроби уникнути систем автоматичного розпізнавання номерних знаків, використання таких методів не є законним та може мати правові наслідки. Крім того, розробники систем розпізнавання номерних знаків постійно вдосконалюють свої технології, щоб протидіяти таким спробам ухилення та забезпечити ефективне функціонування систем контролю дорожнього руху.

Ви правильно вказали на низку можливих труднощів, з якими стикаються системи розпізнавання номерних знаків транспортних засобів. Ось деякі з них:

Погана роздільна здатність: Якщо номерний знак знаходиться далеко від камери або використовується низькороздільна камера, це може призвести до поганої якості зображення номерного знаку, що ускладнює його розпізнавання.

1. Розмите зображення: Під час руху транспортного засобу зображення номерного знаку може бути розмитим через швидкість або вібрації. Це може призвести до втрати деталей та ускладнити процес розпізнавання.

2. Погане освітлення та низький контраст: Якщо номерний знак знаходиться у погано освітленому місці або стикається з проблемою відбиття світла або тіней, це може знизити контраст і зробити номерний знак менш видимим для системи розпізнавання.
3. Закриття номерного знаку: Присутність об'єктів, які перекривають номерний знак, такі як багаж, фари або інші частини автомобіля, може ускладнити процес розпізнавання.
4. Різні дизайни номерних знаків: Різні країни або штати можуть мати власні стандарти та дизайни для номерних знаків. Це може включати різні формати, шрифти, кольори та символи. Наявність різних стандартів ускладнює розпізнавання, особливо при перетинанні кордонів.
5. Зміна кута огляду та смуги руху: Якщо транспортний засіб змінює смугу руху або номерний знак знаходиться під кутом до камери.
6. Варіативність номерних знаків: Номерні знаки можуть відрізнятися в залежності від типу транспортного засобу, наприклад, особові автомобілі, вантажівки, мотоцикли тощо. Це може включати розмір, форму та розташування номерного знаку. Різноманітність цих варіантів ускладнює розпізнавання системи.
7. Зміна шрифту та символів: У деяких випадках власники транспортних засобів можуть змінювати шрифт або символи на номерному знаку, що ускладнює його розпізнавання. Наприклад, змінюються пропорції, ширина ліній, наявність декоративних елементів тощо.
8. Шум та спотворення: Наявність шуму, спотворень або інтерференції на зображенні номерного знаку може створювати проблеми для системи розпізнавання. Це може включати дотримання часткових перешкод, дорожній рух, погодні умови та інші фактори, які можуть впливати на якість зображення.

Для вирішення цих труднощів розробники систем розпізнавання номерних знаків використовують різні техніки, такі як обробка зображень, фільтрація шуму, використання алгоритмів розпізнавання та навчання машинного зору. Також важливо постійно вдосконалювати алгоритми та технології для пристосування до змін у дизайнах номерних знаків та умовах експлуатації.

### **1.3 Використання систем розпізнавання номерних знаків**

Так, автоматичне розпізнавання зображень номерів транспортних засобів має широкий спектр застосувань і може бути корисним для багатьох завдань та сфер діяльності. Ось деякі приклади:

1. Поліція та забезпечення безпеки: Системи розпізнавання номерних знаків можуть бути використані правоохоронними органами для виявлення викрадених або шуканих автомобілів. Також їх можна використовувати для виявлення порушників дорожнього руху, пошуку автомобілів, що здійснюють незаконну діяльність, або для моніторингу дорожнього контролю.
2. Системи паркування: Автоматичне розпізнавання номерних знаків може бути використане для контролю за паркуванням, включаючи системи автоматичної оплати, виявлення порушників, контроль за часом паркування та розрахунок плати.
3. Логістика та відправлення: Системи розпізнавання номерних знаків можуть бути використані для відстеження транспортних засобів в логістичних ґанках. Це дозволяє контролювати переміщення вантажу, впевнитися, що вантажні автомобілі прибувають вчасно та на правильне місце, а також автоматизувати процеси відправлення та отримання.
4. Безпека та контроль доступу: Системи розпізнавання номерних знаків можуть використовуватися для контролю доступу до обмежених зон або для виявлення сторонніх відвідувачів, які не повинні перебувати

в певному районі. Це може бути корисно для охорони приватних територій, підприємств або громадських об'єктів.

5. Управління трафіком: Системи розпізнавання номерних знаків можуть використовуватися для моніторингу дорожнього руху і управління трафіком. Це дозволяє збирати дані про об'єм трафіку, швидкість руху транспорту та виявляти проблемні ділянки доріг для подальшого аналізу та планування.
6. Безпека в громадських місцях: Системи розпізнавання номерних знаків можуть бути використані для забезпечення безпеки в громадських місцях, наприклад, на парковках, стадіонах або аеропортах. Вони можуть допомогти виявити автомобілі, що становлять загрозу безпеці, або вести контроль за в'їздом та виїздом транспорту.
7. Екологічний моніторинг: Системи розпізнавання номерних знаків можуть використовуватися для контролю за викидами шкідливих речовин транспортними засобами. Вони дозволяють виявити автомобілі, що не відповідають стандартам екологічної безпеки, і вжити відповідних заходів для зменшення впливу на навколишнє середовище.

Автоматичне зчитування і оплата дорожнього збору: Системи розпізнавання номерних знаків можуть використовуватися для автоматичного зчитування номерних знаків транспортних засобів і стягнення дорожнього збору або плати за проїзд

Так, розширена аналітика відеоданих на базі машинного навчання відкриває нові можливості для автоматичної ідентифікації номерів автотранспорту та аналізу пов'язаних даних.

Розширена аналітика відеоданих з машинним навчанням значно покращує можливості автоматичного розпізнавання номерів транспортних засобів та дозволяє ефективно використовувати цю інформацію для



забезпечення безпеки, запобігання злочинам та поліпшення управління транспортним потоком.

Так, останні інновації у сфері систем автоматичного розпізнавання зображень номерів транспортних засобів дійсно сприяють використанню таких систем для забезпечення безпеки периметра та контролю доступу в державних установах та приватному секторі, [3].

Деякі конкретні застосування включають:

Захист чутливих об'єктів: у рамках національної безпеки, державні установи, такі як посольства, школи, аеропорти, морські порти, військові та федеральні будівлі, правоохоронні та державні установи, впроваджують системи автоматичного розпізнавання зображень номерів транспортних засобів для виявлення потенційних загроз і захисту об'єктів від передбачуваних актів тероризму.

Приватний сектор: Багато приватних закладів, таких як казино, лікарні, музеї, автостоянки та курорти, також використовують системи автоматичного розпізнавання номерів транспортних засобів для забезпечення безпеки своїх об'єктів. Вони можуть розробляти власні бази даних та порівнювати їх з інформацією про клієнтів, VIP-персон, критично важливий персонал або списки заборонених осіб.

Сервісні додатки: Приватні системи розпізнавання автотранспорту також надають сервісні додатки для керування паркуванням та логістики. Наприклад, вони можуть відстежувати транспортні засоби постачальників або служб підтримки для ефективного управління логістикою та підвищення продуктивності.

Впровадження систем автоматичного розпізнавання зображень номерів транспортних засобів приводить до численних переваг для державних установ і приватного сектору. Деякі з цих переваг включають:

Підвищення безпеки: Системи автоматичного розпізнавання номерів транспортних засобів дозволяють виявляти потенційні загрози та небажану присутність на об'єктах. Швидка ідентифікація викрадених автомобілів, автомобілів, пов'язаних зі злочинними діями, або транспортних засобів з небезпечними особами допомагає запобігти злочинам і підвищує рівень безпеки в цих областях.

Ефективний контроль доступу: Системи автоматичного розпізнавання номерів можуть бути використані для контролю доступу до обмежених або конфіденційних об'єктів. За допомогою таких систем можна встановлювати доступ лише для авторизованих транспортних засобів та вести облік в'їздів і виїздів.

Поліцейські застосування: Поліція та правоохоронні органи використовують системи автоматичного розпізнавання номерів для виявлення порушників правил дорожнього руху, небезпечних водіїв та автомобілів, що знаходяться у розшуку. Це допомагає покращити дорожню безпеку, зменшити кількість порушень та сприяє більш ефективній роботі поліції.

Управління паркуванням: Системи автоматичного розпізнавання номерів можуть бути використані для ефективного управління паркуванням. Вони дозволяють автоматично контролювати доступ до парковок, встановлювати плату за паркування та виявляти недозволене паркування.

Покращення логістики: Впровадження систем розпізнавання номерів транспортних засобів.

Безпека на дорозі: Системи контролю дорожнього руху допомагають забезпечувати безпеку на дорозі шляхом виявлення порушень правил дорожнього руху та небезпечних ситуацій. Наприклад, системи автоматичного розпізнавання номерів можуть виявляти автомобілі з неправильною реєстрацією, використанням заборонених маршрутів або

швидкісних порушень. Це дозволяє правоохоронним органам ефективно вживати заходів щодо безпеки на дорозі та зменшення порушень.

Покращення реагування на аварії: Зібрані дані про дорожні інциденти та аварії можуть бути використані для швидкого реагування та координації дій служб екстреної допомоги. Інформація про місце події, рух транспорту та інші фактори можуть бути передані службам швидкої допомоги для оперативного прибуття на місце та надання необхідної допомоги.

Поліпшення планування і розробки інфраструктури: Зібрані дані про трафік, рух транспорту та інші параметри можуть бути використані для поліпшення планування та розробки інфраструктури. Аналіз даних може допомогти виявити місця з великим навантаженням, проблемні перехрестя або необхідність будівництва нових доріг. Це дозволяє ефективно розподіляти ресурси та вдосконалювати дорожню мережу для забезпечення оптимального руху транспорту.

Загалом, системи контролю дорожнього руху з камерами відеоспостереження та системами автоматичного розпізнавання номерів транспортних засобів мають великий потенціал для покращення безпеки руху.

Системи автоматичного розпізнавання зображень номерів транспортних засобів використовуються в різних сферах діяльності, зокрема й для перелічених вами цілей. Надалі розглянемо кожен з цих областей детальніше:

1. Перетин кордону: Системи автоматичного розпізнавання номерів можуть використовуватися для автоматичного контролю транспортних засобів, які перетинають кордон. Це дозволяє здійснювати швидку та ефективну перевірку номерів автомобілів та порівнювати їх з базами даних, що містять інформацію про викрадені, шукаємі або небезпечні транспортні засоби.

2. Вилючення автомобілів: Системи автоматичного розпізнавання можуть допомагати правоохоронним органам у вилученні автомобілів, які мають проблеми з платежами, страховкою або розшуком. Вони можуть автоматично впізнавати номери таких автомобілів та сповіщати відповідні служби для подальшої дії.
3. Вимірювання середньої швидкості транспортного засобу на великих відстанях: Системи автоматичного розпізнавання можуть фіксувати номери транспортних засобів на різних ділянках дороги та обчислювати час, який автомобіль займає для проходження між цими точками. За допомогою цих даних може бути розрахована середня швидкість транспортного засобу на певній відстані.
4. Забезпечення виконання законів про переїзд транспортних засобів екстреної допомоги: Системи автоматичного розпізнавання можуть використовуватися для визначення транспортних засобів екстреної допомоги.

#### **1.4 Висновок до першого розділу**

Розвиток інформаційних систем розпізнавання та ідентифікації зображень номерів транспортних засобів пройшов значний шлях і знайшов широке застосування у різних сферах діяльності. Принципи роботи таких систем полягають у використанні відеозаписів або зображень транспортних засобів, на яких присутні номерні знаки. Система застосовує алгоритми розпізнавання, щоб автоматично виявити та ідентифікувати номерні знаки на зображеннях.

Однак, виконання ідентифікації номерних знаків може зустрічати труднощі через різноманітні фактори, такі як погане освітлення, різні типи транспортних засобів, забруднення або пошкодження номерних знаків, а також різні шрифти і стилі написання номерів.

На сьогоднішній день існують різні системи розпізнавання номерних знаків автотранспорту, які використовуються у практиці. Вони можуть включати в себе спеціалізоване обладнання, таке як камери відеоспостереження, оптичні читачі, системи комп'ютерного зору та програмне забезпечення, яке здійснює обробку і аналіз зображень.

Використання інформаційних систем автоматичного розпізнавання та ідентифікації зображень номерів транспортних засобів є доречним у різних сферах діяльності.

## РОЗДІЛ 2

### АНАЛІЗ ТА ВИБІР ІНСТРУМЕНТІВ РОЗПІЗНАВАННЯ АВТОМОБІЛЬНИХ НОМЕРІВ

#### 2.1 Технологія розпізнавання автомобільних номерів OCR

Оптичне розпізнавання символів (Optical Character Recognition, OCR) є технологією, яка використовується для автоматичного перетворення тексту, надрукованого або написаного вручну, на комп'ютерно-читабельний формат. OCR використовується для розпізнавання символів і знаків на зображеннях, отриманих з різних джерел, таких як сканери, фотографії або відеозаписи. На рис.2.1 наведено схематичне представлення OCR [4].

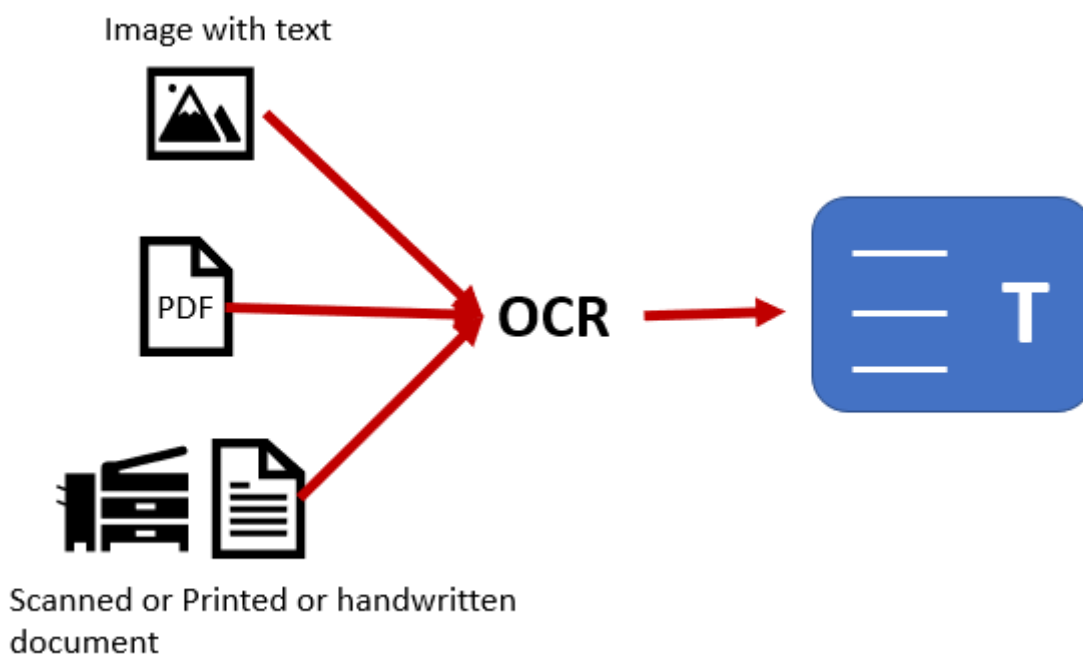


Рис. 2.1. Структурна схема OCR

Основні принципи роботи OCR включають наступні кроки.

Крок 1. Попередня обробка зображення. Зображення, що містить текст, піддається попередній обробці, яка може включати такі етапи, як

вирівнювання, фільтрація шуму, підсилення контрасту тощо. Метою цього етапу є поліпшення якості зображення та виділення тексту.

Крок 2. Сегментація. Зображення розділяється на окремі області, що містять окремі символи або групи символів. Цей етап допомагає виділити окремі символи для подальшого розпізнавання.

Крок 3. Витягнення ознак. Кожен символ або група символів аналізується з точки зору їхніх геометричних ознак, таких як форма, розмір, контур тощо. Також можуть використовуватись інші характеристики, наприклад, текстурні ознаки.

Крок 4. Класифікація. На основі витягнутих ознак кожен символ або група символів класифікуються за допомогою моделі, яка навчена розпізнавати різні символи. Це може бути статистична модель або модель машинного навчання, така як нейронна мережа.

Крок 5. Пост-обробка. Отримані результати розпізнавання піддаються пост-обробці для виправлення помилок та вдосконалення результатів. Цей етап може включати застосування алгоритмів корекції помилок, які виправляють неправильно розпізнані символи на основі контексту або шаблонів символів. Також можуть використовуватись методи пост-обробки, такі як з'єднання розпізнаних символів у слова або номери, видалення зайвих символів тощо.

Застосування OCR в розпізнаванні автомобільних номерів дозволяє автоматично виділяти та розпізнавати номерні знаки на зображеннях автомобілів. Це має широкий спектр застосувань, включаючи автоматизовану систему контролю швидкості, системи безпеки на дорогах, системи контролю доступу, парковочні системи та багато інших.

Переваги використання OCR в розпізнаванні автомобільних номерів [4]:

- автоматизація: OCR дозволяє автоматизувати процес розпізнавання номерних знаків, що зменшує необхідність у вручну виконуваних операціях;
- швидкість: Завдяки автоматичному розпізнаванню, OCR забезпечує швидку обробку та ідентифікацію номерних знаків у реальному часі;
- висока точність: З використанням оптимізованих алгоритмів та тренуванням моделей OCR може досягати високої точності розпізнавання символів.

Обмеження використання OCR в розпізнаванні автомобільних номерів:

- вплив умов зйомки: Результати OCR можуть бути суттєво погіршені у разі низької якості зображень, наприклад, при наявності шуму, тості, низькому контрасті або неякісному освітленні. Поганий стан номерного знака або його пошкодження також можуть впливати на точність розпізнавання;
- різноманітність форматів та стилів номерних знаків: Різні країни та регіони можуть мати власні стандарти та варіанти номерних знаків, що ускладнює розпізнавання через неоднорідність форматів, шрифтів та стилів;
- мовні особливості: Номерні знаки можуть містити специфічні символи, літери або знаки пунктуації, що можуть бути проблемними для розпізнавання, особливо якщо використовуються різні алфавіти чи складні системи квитанцій;
- висока вимогливість до продуктивності: Оперативне розпізнавання номерних знаків в реальному часі вимагає потужних обчислювальних ресурсів та оптимізованих алгоритмів, що може бути викликом у випадку масштабних систем або обмеженого обладнання;
- вплив зовнішніх факторів: Розпізнавання автомобільних номерів може бути ускладненим у випадку наявності перешкод, наприклад, інших автомобілів, дорожньої смути або об'єктів на задньому плані.



Незважаючи на ці обмеження, OCR є потужним інструментом для розпізнавання автомобільних номерів, здатним до досягнення високої точності при оптимальних умовах та належній налаштуванні. При виборі відповідного рішення для розпізнавання автомобільних номерів, слід враховувати особливості конкретного контексту використання та оптимальні умови для забезпечення максимальної ефективності OCR.

Додатково, важливо враховувати питання безпеки та конфіденційності при використанні OCR для розпізнавання автомобільних номерів. Збір та збереження даних, пов'язаних з номерними знаками, потребує належних заходів захисту для запобігання несанкціонованому доступу та зловживанню.

## **2.2 Технологія обробки зображень та комп'ютерного зору**

Методи обробки зображень та комп'ютерного зору є важливими компонентами для розпізнавання автомобільних номерів. Вони допомагають покращити якість зображення, виділити інформативні ознаки та виконати необхідні операції для точного розпізнавання номерних знаків. Нижче описані деякі з найпоширеніших методів обробки зображень та комп'ютерного зору, що застосовуються в контексті розпізнавання автомобільних номерів:

### **2.2.1 Пре-обробка зображення:**

- зміна розміру (resizing): Зображення може бути зменшене або збільшене до стандартних розмірів для полегшення подальшої обробки та зменшення обчислювальних витрат;
- перетворення кольору (color transformation): Можуть використовуватись різні перетворення кольору, такі як перетворення в чорно-біле зображення або зменшення кількості кольорів, для спрощення аналізу;
- фільтрація (filtering): Застосування фільтрів для зменшення шуму, розмиття зображення або виділення особливих ознак;

- підсилення контрасту (contrast enhancement): Методи, що дозволяють покращити контрастність зображення, зробити його більш чітким та виділити деталі.

#### 2.2.2 Сегментація зображення:

- бінаризація (binarization): Процес перетворення зображення в чорно-біле, де об'єкти відокремлені від фону шляхом заданого порогу;
- виокремлення контурів (contour extraction): Виділення контурів об'єктів на зображенні для подальшого аналізу та розпізнавання;
- сегментація за кольором (color-based segmentation): Розділення об'єктів на зображенні за їхнім кольором, що допомагає від важити номерні знаки на зображенні.

#### 2.2.3 Виділення ознак номерного знаку:

- детектори країв (edge detectors): Виявлення границь та контурів на зображенні, що можуть вказувати на наявність номерних знаків;
- виділення регіонів інтересу (Region of Interest - ROI): Визначення та виділення областей на зображенні, де ймовірно розташовані номерні знаки, для подальшої обробки;
- екстракція особливих ознак (feature extraction): Виділення характерних ознак номерних знаків, таких як форма, текстура, контури тощо, для подальшого використання в алгоритмах розпізнавання.

#### 2.2.4 Класифікація та розпізнавання:

- методи класифікації: Використання різних алгоритмів класифікації, таких як методи машинного навчання (наприклад, нейронні мережі, метод опорних векторів) або статистичні підходи, для визначення символів на номерному знаку.
- шаблонне співставлення (template matching): Порівняння зразків номерних знаків з обробленим зображенням для знаходження відповідності.

– нейромережеві алгоритми: Використання глибоких нейронних мереж для розпізнавання номерних знаків, що дозволяє досягти високої точності.

Оскільки більшість алгоритмів не здатні точно знайти номерний знак, проводиться нормалізація та сегментація зображення номера. Після цього, на вихідному зображенні вирізається рамка, що вміщує автомобільний номер, як показано на рисунку 2.2. Враховуються такі проблеми, як невідомий нахил рамки, можливе забруднення на номері або розмитість зображення. Ці проблеми є лише частиною викликів, які потрібно вирішити для успішної подальшої обробки системою.



Рис. 2.2. Результат після пре-обробки

Метод Віоли-Джонса [5] є одним із широко використовуваних методів для передпошуку областей із зображеннями, які можуть містити автомобільні номерні знаки. Цей метод базується на використанні класифікаторів зображень, навчених на наборах даних з позитивними та негативними зразками.

Опередній пошук з використанням методу Віоли-Джонса зазвичай включає наступні етапи:

- Побудова позитивного та негативного навчального набору даних. Позитивні зразки представляють собою зображення номерів автомобілів, а негативні зразки - зображення без номерів. Ці набори даних використовуються для навчання класифікаторів.

- Навчання класифікатора. Використовуючи позитивний та негативний навчальний набір даних, проводиться тренування класифікатора. Метод Віоли-Джонса використовує ансамбль слабких класифікаторів, які утворюють сильний класифікатор.

- Створення скануючого вікна. Визначається розмір та форма вікна, яке буде сканувати зображення з метою виявлення можливих областей з номерними знаками.

- Сканування зображення. Зображення проходить через скануюче вікно, яке зміщується та масштабується по всьому зображенню для пошуку можливих областей з номерними знаками.

- Виявлення областей з номерними знаками: Застосовується класифікатор Віоли-Джонса до кожної області, яку скануюче вікно знаходить. Якщо область відповідає певним критеріям, вона вважається потенційною областю з номерним знаком.

Ці методи обробки зображень та комп'ютерного зору спільно використовуються для досягнення точного та надійного розпізнавання автомобільних номерів. Комбінація цих методів забезпечує відокремлення номерних знаків від фону, виділення їх особливих ознак та класифікацію символів, що дозволяє отримати дієвий інструмент для автоматизованого розпізнавання номерів автотранспорту.

## **2.3 Технології розпізнавання зображень номерних знаків в існуючих системах**

В цьому пункті розглянемо популярні системи розпізнавання номерних знаків.

### **2.3.1 HikVision**

Ця система розпізнавання номерних знаків розроблена Hikvision Digital Technology [5], яка є світовим лідером у галузі систем відеоспостереження та

безпеки. Розпізнавання автомобільних номерів підтримується на всіх камерах 4-ї серії DS-2CD4xxx Smart-IP.

Система розпізнавання автономерів від HikVision пропонується у трьох різних комплектаціях. Перший варіант дозволяє використовувати лише камеру. За допомогою браузера ви можете підключитися до камери і створити базу даних з фіксованим набором номерів. Коли автомобіль проїжджає, камера автоматично управляє шлагбаумом. Якщо номер автомобіля знаходиться у білому списку, шлагбаум відкривається. Якщо номера немає у списку, шлагбаум залишається закритим.

Однак, цей варіант має деякі обмеження. Наприклад, дана система не зберігає дані про час або напрямок проїзду, тому немає можливості встановити інформацію про те, хто і коли проїжджав через шлагбаум або побудувати звіти. Крім того, для створення "чорних" і "білих" списків автомобільних номерів потрібно виконати цю дію на кожній окремій камері. Якщо камер багато, це може зайняти значний час.

Отже, система розпізнавання номерних знаків від HikVision має свої переваги та обмеження, які варто враховувати при виборі рішення для ваших потреб.

### **2.3.2 Axis**

Завдяки використанню відкритої платформи Axis Communications – АСАР [6], сторонні розробники мають можливість створювати програми, які можна безпосередньо встановлювати на IP камери. І саме за допомогою цієї платформи реалізована можливість розпізнавання автомобільних номерів на камерах Axis.

Одним з розробників програмного забезпечення для розпізнавання автомобільних номерів є компанія FF Group. Вони розробили додаток, який

може бути встановлений на камери Axis. На сьогоднішній день цей додаток адаптовано для країн Євросоюзу, Ізраїлю та Туреччини.

Це означає, що власники камер Axis можуть скористатися цим рішенням для розпізнавання автомобільних номерів у відповідних регіонах. Цей додаток від FF Group дозволяє забезпечити високу точність розпізнавання та зручне інтегрування з існуючою інфраструктурою камер.

Ця можливість розпізнавання номерних знаків на камерах Axis, реалізована завдяки відкритій платформі АСАР, розширює функціональні можливості камер і дозволяє використовувати їх для різних сценаріїв застосування, включаючи системи безпеки, контроль доступу, автоматизований паркувальний контроль та інші.

### **2.3.3 Трал-Паркінг**

Система розпізнавання автомобільних номерів в Трал-Паркінг побудована на базі компактних готових модулів під назвою "Трал-Паркінг 2" [7]. Кожен модуль складається з аналогової відеокамери та контролера, який відповідає за обробку зображення, перевірку автомобільних номерів та керування виконавчими пристроями, що підключені до його релейних виходів.

Ця система має два варіанти реалізації модулів. Перший варіант - придбати готовий виріб в герметичному корпусі з рівнем захисту IP66. Другий варіант - придбати окремо контролер із камерою для самостійної установки.

Програмне забезпечення для розпізнавання автомобільних номерів вже записане в пам'ять контролера, і сам контролер підключається до комп'ютера за допомогою ТСР/ІР протоколу через інтерфейс NetCore Паркінг. Це дозволяє переглядати події та налаштовувати модулі розпізнавання в режимі реального часу.

Контролери можуть працювати автономно, маючи внутрішній USB-порт для підключення зовнішнього пристрою, який зберігає базу номерів та записує події проїзду. Кількість контролерів в системі може бути будь-якою, проте важливо, щоб вони всі були підключені до однієї локальної мережі, щоб забезпечити онлайн перегляд та налаштування.

Інформація про події проїзду включає фотографію автомобіля з його державним реєстраційним знаком, час проїзду, напрямок руху, результат розпізнавання номера та його приналежність до певної групи.

Для перегляду відео з точок проїзду, редагування списку номерів автомобілів, передачі їх в пам'ять контролерів та зберігання архіву подій використовується робоча станція. Жодного іншого способу роботи з модулями розпізнавання немає. Система розпізнавання забезпечує точність до 92%, що гарантує надійні результати при розпізнаванні автомобільних номерів.

Ця система розпізнавання автомобільних номерів на основі модулів "Трал-Паркінг 2" є ефективним і зручним рішенням, яке дозволяє контролювати проїзд автомобілів, зберігати відомості про них та забезпечувати безпеку на парковках та в інших об'єктах з великим потоком транспорту.

#### **2.3.4 Номер**

Номер є програмно-апаратним комплексом, призначеним для розпізнавання автомобільних номерів та керування виконавчими пристроями. Цей модуль здатен розпізнавати автомобільні державні реєстраційні знаки України, Ізраїлю та більшості європейських країн з високою ймовірністю до 97% [8].

Модуль НомерОк може працювати як з IP, так і з аналоговими відеокамерами. Управління зовнішніми пристроями здійснюється за

допомогою релейних блоків-контролерів "Барбос" і ICP CON PET-7060. Ці контролери не тільки видають керуючі сигнали, але також можуть приймати сигнали від інших пристроїв, таких як фотоелементи, індукційні петлі та інші, що мають сумісний вихідний сигнал.

Інтерфейс роботи з контролерами вбудований в основне програмне забезпечення. Кількість відеокамер, які можуть бути одночасно підключені до модуля розпізнавання, програмно обмежена до 8.

У модулі НомерОк реалізовано два варіанти роботи з базою даних. Перший варіант передбачає встановлення сервера бази даних SQLite разом з модулем розпізнавання на одному локальному комп'ютері, що забезпечує автономну роботу. Другий варіант передбачає використання сервера бази даних Firebird, де кілька терміналів з модулем розпізнавання працюють з однією базою. У такому випадку не потрібна окрема клієнтська частина, і для віддаленого адміністрування встановлюється додаткове програмне забезпечення "НомерОк". Зв'язок з сервером бази даних встановлюється постійно. Кількість терміналів з модулем розпізнавання, які можуть працювати з однією базою даних, необмежена.

У модулі НомерОк існує можливість налаштування тривожних подій для окремого номера або групи номерів. Тривожна подія може бути викликана наступними способами:

- відображення текстової інформації будь-якого змісту, яка дозволяє прийняти відповідні дії, такі як затримка, перегляд або пропуск автомобіля;
- відтворення звукових сигналів, де для кожної події можна налаштувати свій звуковий файл;
- використання світлової індикації, для чого до контролера підключаються сигнальні лампи або світло діоди;



– активація зовнішніх пристроїв за допомогою релейних виходів контролерів.

Крім того, модуль НомерОк інтегрується з системою контролю доступу (СКУД). Сервер розпізнавання номерів СКУД Gate самостійно не приймає рішень, а лише розпізнає номер автомобіля та передає його як ідентифікатор безпосередньо в контролер Gate-8000 Авто. Таким чином, сервер розпізнавання стає зчитувачем ідентифікаторів для контролера Gate, і вся робота системи доступу виконується в штатному режимі згідно з типовими сценаріями та принципами класичної СКУД.

### **2.3.5 Модуль iPera EX-LPR**

Модуль iPera EX-LPR є спільною розробкою компаній iPera і Exacq Technologies Inc [9]. Цей модуль є частиною системи відеоспостереження ExacqVision і призначений для автоматичного розпізнавання автомобільних номерів, які потрапляють в поле зору відеокамери, та їх реєстрації.

Для коректної роботи модуля необхідно встановити програмне забезпечення відеоспостереження ExacqVision. Для невеликої системи це програмне забезпечення може бути встановлене на один комп'ютер.

Особливістю цього модуля є можливість пошуку автомобільних номерів у всьому кадрі або в певній області, яка встановлюється при налаштуванні модуля. Це має вплив на обчислювальну потужність комп'ютерного обладнання, яке використовується.

Кількість відеокамер, які можуть бути підключені одночасно до модуля розпізнавання, обмежена апаратними можливостями використовуваного обладнання. Для забезпечення належної швидкості роботи модуля розробники рекомендують використовувати одне ядро процесора на один канал розпізнавання.

У разі автономної роботи модуль використовує тривожні виходи відеокамер для керування виконавчими пристроями. Однак, модуль має інтеграційні можливості для розширення цієї функціональності.

Модуль розпізнавання EX-LPR є сервер-клієнт додатком, заснованим на сервері бази даних MySQL. База даних може бути збережена на будь-якому комп'ютері або сервері з встановленим сервером MySQL та доступом по мережі.

Всі клієнтські підключення безкоштовні і не мають обмежень за їх кількістю. Через таке підключення можна переглядати події, налаштовувати систему, редагувати списки номерів та створювати звіти.

Розробники також реалізували другий тип клієнтського підключення - веб-інтерфейс. Цей інтерфейс надає зручний спосіб взаємодії з модулем розпізнавання EX-LPR через веб-браузер. Завдяки цьому, користувачі можуть отримати доступ до функцій модуля, переглядати відеозаписи, налаштовувати параметри та здійснювати інші дії безпосередньо через веб-інтерфейс.

Таким чином, модуль iPera EX-LPR є потужним рішенням для автоматичного розпізнавання автомобільних номерів, спроектованим у співпраці з компанією Exacq Technologies Inc. Він інтегрується з системою відеоспостереження ExacqVision, пропонуючи різноманітні можливості розпізнавання та керування номерами, широкий вибір підключених камер та гнучкість в управлінні виконавчими пристроями.

### **2.3.6 АТАРУ ANPR SDK**

АТАРУ ANPR SDK [10] є спеціалізованим програмним продуктом, який пропонує технологію автоматичного розпізнавання номерів автомобілів. Цей продукт призначений для системних інтеграторів та розробників прикладних програм, які прагнуть використовувати функціонал автоматичного

розпізнавання номерів у своїх додатках. АТАРУ ANPR SDK відрізняється високою точністю розпізнавання, оскільки використовує один з найточніших продуктів в галузі OCR - движок OCR, розроблений ABBYY Software House.

Крім того, в системі АТАРУ ANPR SDK реалізована функція автоматичної геометричної корекції перекосу або нахилу, яка дозволяє вирівнювати номери автомобілів навіть при наявності похилої або нахиленої позиції (до 30°). Ця функція забезпечує додаткову гнучкість та надійність в процесі розпізнавання номерів.

Отже, АТАРУ ANPR SDK є потужним інструментом для інтеграції технології автоматичного розпізнавання номерів автомобілів у рішення системних інтеграторів та розробників прикладних програм. Використання продукту дозволяє досягти високої точності розпізнавання та забезпечити корекцію геометричних перекосів для надійної і ефективної роботи.

## **2.4 Вибір OpenCV інструменту для розпізнавання автомобільних номерів**

Вибір інструментів розпізнавання автомобільних номерів на основі OpenCV відкриває широкі можливості для розробки систем автоматичного розпізнавання. OpenCV є відкритою бібліотекою комп'ютерного зору, яка надає набір функцій та алгоритмів для обробки зображень і відео.

Одним з ключових критеріїв при виборі інструментів розпізнавання автомобільних номерів на основі OpenCV є його потужність та ефективність. OpenCV забезпечує широкий спектр функцій для обробки зображень, включаючи фільтрацію, виявлення контурів, розпізнавання символів та візуалізацію. Ці функції дозволяють ефективно виконувати процес розпізнавання автомобільних номерів з високою точністю.

Крім того, OpenCV є популярним та активно розвиваючимся інструментом з великою спільнотою розробників. Це означає, що ви

отримуєте доступ до багатьох документаційних матеріалів, прикладів реалізації, а також підтримки спільноти, яка може надати допомогу та поради під час роботи з розпізнаванням автомобільних номерів на основі OpenCV.

Додатковою перевагою використання OpenCV є його кросплатформеність, що дозволяє розробляти рішення для різних операційних систем, таких як Windows, Linux, macOS і багатьох інших.

Загалом, вибір інструментів розпізнавання автомобільних номерів на основі OpenCV відкриває широкі можливості для створення потужних та ефективних систем розпізнавання. Завдяки потужності OpenCV, доступності документації та підтримки спільноти. Це дозволяє створювати різноманітні функції для обробки зображень та виявлення автомобільних номерів. Зокрема, OpenCV надає можливість застосування алгоритмів просторової фільтрації, виявлення контурів, виокремлення регіонів із зображеннями номерів, а також використання алгоритмів машинного навчання для розпізнавання символів на номерних знаках.

При виборі інструментів розпізнавання автомобільних номерів на основі OpenCV, можна враховувати можливості інших додаткових бібліотек або інструментів, які можуть підтримувати їхні потреби. Наприклад, для покращення точності розпізнавання можна використовувати додаткові бібліотеки для обробки зображень або навчання моделей машинного навчання.

Крім того, важливим фактором є ефективне використання ресурсів обчислювальної системи. OpenCV надає можливості оптимізації та прискорення обробки зображень, такі як використання паралельних обчислень на багатоядерних процесорах або використання апаратного прискорення, якщо таке є в наявності.

Загалом, вибір інструментів розпізнавання автомобільних номерів на основі OpenCV вимагає ретельного аналізу потреб проекту, наявних ресурсів

і можливостей бібліотеки. Професіонали в галузі комп'ютерного зору та розробки програмного забезпечення можуть ефективно використовувати OpenCV для реалізації потужних та надійних систем розпізнавання автомобільних номерів.

## РОЗДІЛ 3

### РОЗРОБКА ТА ПЕРЕВІРКА ЕФЕКТИВНОСТІ СИСТЕМИ ПОШУКУ АВТОМОБІЛЬНИХ ЗНАКІВ

#### 3.1 Інтеграція OpenCV бібліотеки комп'ютерного зору в систему

Для ефективної обробки зображень одним з найкращих варіантів є використання системи комп'ютерного зору.

OpenCV [11] є відкритою бібліотекою алгоритмів комп'ютерного зору, яка спеціально розроблена для обробки зображень і реалізована на мовах програмування C/C++. Крім того, є варіанти реалізації для Python, Java, Ruby, Matlab, Lua та інших мов програмування.

OpenCV надає можливість використовувати її як в академічних, так і в комерційних цілях, оскільки поширюється під ліцензією BSD. Ця бібліотека включає реалізацію різних алгоритмів, які були згадані в попередніх розділах. Фактично, OpenCV є комплексом типів даних, функцій та класів, призначених для обробки зображень засобами алгоритмів комп'ютерного зору.

Основні модулі бібліотеки можна побачити на прикладі нижче (див. рисунок 3.1). Крім цих модулів, OpenCV також надає широкий спектр інструментів і функцій для роботи з зображеннями.

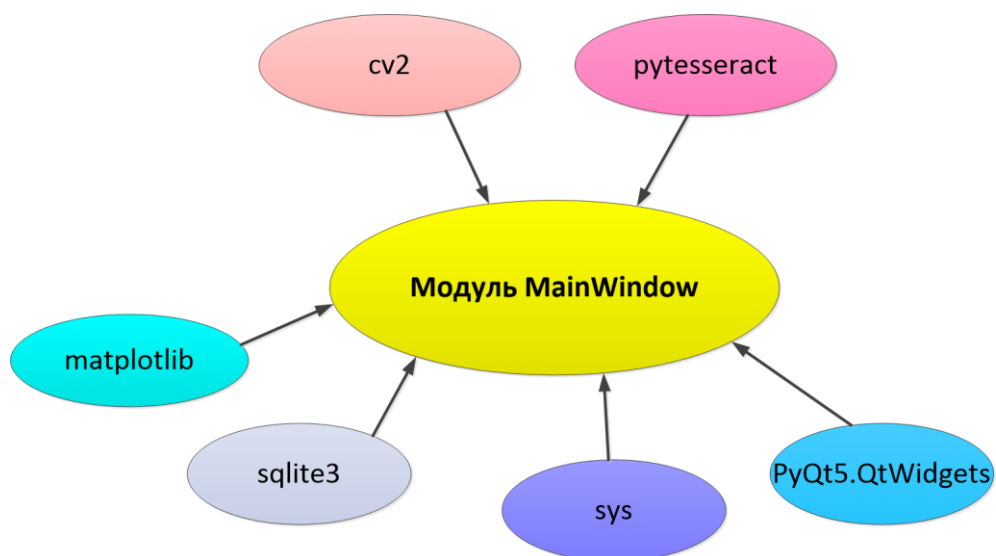


Рис. 3.1. Структурна схема, використаних модулів

В основному модулі програми, яким є `MainWindow`, використовуються наступні модулі:

- `PyQt5.QtWidgets`: Модуль для створення графічного інтерфейсу користувача;
- `cv2`: Модуль `OpenCV` для обробки зображень;
- `sys`: Модуль для доступу до функцій та змінних, пов'язаних з інтерпретатором Python;
- `sqlite3`: Модуль для роботи з базою даних SQLite;
- `matplotlib.pyplot`: Модуль для візуалізації даних та графічного представлення зображень;
- `matplotlib.image`: Модуль для завантаження та обробки зображень.

Ці модулі використовуються для різних функцій, таких як створення віконного інтерфейсу, обробка зображень, взаємодія з базою даних та відображення зображень та результатів розпізнавання.

CV (Computer Vision) - це модуль, який забезпечує обробку зображень та розпізнавання об'єктів за допомогою комп'ютерного зору. Він надає широкі можливості для роботи з зображеннями і виконання різноманітних операцій. Основні функціональні можливості модуля CV включають:

1 Застосування базових операцій над зображеннями, таких як фільтрація зображення за допомогою алгоритмів контурів, геометричні перетворення зображення, зміна кольорів та інші.

2 Аналіз зображень, включаючи виділення важливих ознак, застосування морфологічних операцій, пошук контурів та створення гістограм.

3 Виявлення номерів автомобілів.

Цей модуль дозволяє розробникам виконувати різноманітні завдання в галузі комп'ютерного зору, досліджувати зображення, виявляти об'єкти та робити аналіз зображень у своїх додатках та системах.

### 3.2 Розробка архітектури пошуку номерних знаків авто

Для реалізації системи пошуку номерних знаків враховано такі аспекти як: вхідні дані, передпроцесинг, виявлення областей інтересу, вирізання номерних знаків, препроцесинг номерних знаків, розпізнавання номерних знаків та відображення результатів. Структурна схема архітектури представлена на рис.3.2.



Рис. 3.2. Архітектура, розробленої системи



Вхідні дані: Система отримує зображення, на якому потрібно здійснити пошук номерного знака. Це може бути фотографія, відеопотік або послідовність кадрів.

Розглянемо етапи навчання класифікатора для модуля знаходження номерів після вже попередньо описаного алгоритму з використанням каскадного класифікатора Хаара, що є один з найпростіших способів визначення кута нахилу зображення полягає у застосуванні.

Каскадний класифікатор Хаара - це алгоритм машинного навчання, який використовується для виявлення об'єктів на зображеннях. Він оснований на використанні характеристик Хаара, які є прямокутними фільтрами, що описують різні особливості об'єктів, наприклад, зміна яскравості між двома областями зображення.

Каскадний класифікатор Хаара складається з декількох послідовно застосованих класифікаторів, які шукають характеристики Хаара на зображенні. Кожен класифікатор складається з навченого моделі машинного навчання, яка використовується для розпізнавання об'єктів. Послідовне застосування класифікаторів дозволяє відфільтрувати області зображення, що не містять об'єкт, і зосередитися на потенційних областях з об'єктами.

У програмній реалізації в Пайтоні, бібліотека OpenCV надає функціонал для роботи з каскадним класифікатором Хаара. Наведений нижче лістинг на рис.3.3 демонструє реалізацію перетворення Хаара для виявлення прямих ліній на зображенні в мові Python з використанням бібліотеки OpenCV.

Алгоритм можна описати як: припустимо, що ми маємо квадратне зображення розміром  $N \times N$  пікселів. Функції Хаара визначаються шляхом вибору двох прямокутних областей (так званих фільтрів): позитивного і негативного фільтрів.

Позитивний фільтр представляє собою прямокутну область, де піксельні значення дорівнюють одному числу (наприклад, +1), інші піксельні

значення дорівнюють іншому числу (наприклад, -1), а всі інші піксельні значення рівні нулю.

Негативний фільтр є оберненим до позитивного фільтра: піксельні значення в області позитивного фільтра дорівнюють -1, піксельні значення в області негативного фільтра дорівнюють +1, а всі інші піксельні значення рівні нулю.

Функції Хаара можуть бути визначені для різних форм та розмірів прямокутних областей. Наприклад, прямокутні області можуть бути квадратними, прямокутними або прямокутними з випуклими частинами.

Загалом, функції Хаара можуть бути використані для виявлення різних особливостей на зображеннях, таких як ребра, кути, лінії тощо. При використанні каскадного класифікатора Хаара, навчання проводиться шляхом визначення оптимальних ваг та порогових значень для різних характеристик Хаара, що дозволяє класифікатору ефективно визначати наявність об'єктів.

```
import cv2
import numpy as np

def recognize_license_plate(image_path):
    # Завантажуємо зображення
    image = cv2.imread(image_path)

    # Перетворюємо зображення в чорно-біле
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    # Застосовуємо фільтр для видалення шуму та полегшення розпізнавання
    gray = cv2.bilateralFilter(gray, 11, 17, 17)

    # Використовуємо класифікатор Хаара для знаходження регіонів із номерними знаками
    cascade_path = "path_to_cascade_xml_file.xml" # шлях до файлу каскадного класифікатора
    cascade = cv2.CascadeClassifier(cascade_path)
    license_plates = cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5, minSize=(25, 25))
```

Рис. 3.3. Програмна реалізація перетворення Хаара

З лістингу наведеного на рис.3.3, можна зробит такі висновки :

1. Завантаження зображення. Використовуючи шлях `image_path`, функція `cv2.imread()` завантажує зображення з файлу і зберігає його в змінній `image`.

2. Перетворення зображення в чорно-біле. Зображення `image` конвертується в градації сірого за допомогою функції `cv2.cvtColor()`. Результат зберігається в змінній `gray`.

3. Застосування фільтра для видалення шуму. Змінна `gray` проходить крізь фільтр `cv2.bilateralFilter()`, який допомагає видалити шуми на зображенні та полегшити подальше розпізнавання. Отриманий результат зберігається знову в змінній `gray`.

4. Використання класифікатора Хаара для знаходження регіонів з номерними знаками: У цьому кроці використовується каскадний класифікатор Хаара, який побудований на основі навчальних зображень, що містять номерні знаки. Каскадний класифікатор завантажується з файлу `cascade_path` за допомогою `cv2.CascadeClassifier()`. За допомогою функції `cascade.detectMultiScale()`, класифікатор шукає можливі регіони з номерними знаками на зображенні `gray`. Отримані регіони зберігаються в змінній `license_plates` у вигляді прямокутників областей (координати та розміри).

Після локалізації зображення номера, його необхідно підготувати для подальшої обробки. Це досягається за допомогою модуля сегментації та обробки зображення. Одним з етапів цієї обробки є нормалізація зображення номера. Для досягнення цього результату застосовується алгоритм бінаризації, в даному випадку алгоритм Оцу. Цей алгоритм дозволяє розділити пікселі зображення на два класи: "корисні" та "фонові", шляхом обчислення оптимального порогу, який мінімізує дисперсію між цими двома класами. Результат наведено на рис. 3.4.



Рис. 3.4. Пошук номерного знака

Наступним кроком є обертання зображення на кут, який був визначений на попередньому етапі. Крім того, можна відразу вирізати зображення за допомогою знайдених ліній після застосування Хаара. На прикладі, ці дії показані на рисунку 3.5.



Рис. 3.5. Результат повороту зображення

Після виділення області номерного знаку необхідно покращити контрастність зображення. Це можна зробити шляхом лінійного розтягнення гистограми зображення. Після цього отримуємо правильно орієнтований чорно-білий номер. Наступним кроком буде застосування фільтра LoG (Laplacian of Gaussian), який описується такою формулою [12]:

$$LoG(x, y) = -\frac{1}{\pi \cdot \sigma^4} * \left(1 - \frac{x^2 + y^2}{2\sigma^2}\right) * e^{\frac{x^2 + y^2}{2\sigma^2}}, \quad (3.1)$$

На рисунку 3.6 наведено результат фільтра LoG.



Рис. 3.6. Фільтр LoG для номерного знака

Після бінаризації настає етап обробки зображення. Спочатку необхідно вирівняти номер таким чином, щоб він був горизонтально розташований. Для цього потрібно знайти всі горизонтальні лінії на номері та розрахувати кути відхилення відносно цих ліній. Потім обчислюється середнє арифметичне цих кутів, і номер повертається на відповідний кут.

Після стабілізації номера у горизонтальне положення здійснюється обрізка зображення з використанням алгоритму контурів. На зображенні присутні всі елементи, які можна помістити в контур. Потрібно вирізати перший знайдений контур, який зазвичай є контуром номерного знака, а не його окремих символів.

Після цього етапу обробки зображення завершується, і можна переходити до сегментації символів на отриманому зображенні.

Наступний етап це сегментація. Під час сегментації необхідно забезпечити, щоб при виділенні символів, таких як Н, В, цифри 9, 0, 8 та інші, не було непотрібних елементів. Необхідно перевіряти, чи не перетинаються ці символи з будь-якою іншою областю всередині них. Якщо цю перевірку не виконувати, то під час сегментації будуть виділятися зайві елементи, як показано на рисунку 3.7.



Рис. 3.7. Виділення зайвих областей

На цьому етапі пропонується виконати видалення областей, програмний код наведено на рис.3.8, а результат коду на рисунку 3.9.

```
def remove_regions(image):
    # Виконуємо попередню обробку зображення (бінаризацію, згладжування тощо)
    processed_image = preprocess_image(image)

    # Застосовуємо алгоритм видалення областей
    # Це може включати пошук контурів, видалення областей на підставі розміру або інших критеріїв
    result_image = remove_regions_algorithm(processed_image)

    return result_image

# Завантажуємо зображення
image = cv2.imread("image_path.jpg")

# Викликаємо функцію видалення областей
result = remove_regions(image)

# Відображаємо результат
cv2.imshow("Result", result)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Рис. 3.8. Програмна реалізація видалення зайвих областей



Рис 3.9. Результат правильного виділення елементів

Модуль розпізнавання символів використовується для отримання кінцевого результату - розпізнаного номера у текстовому форматі. В даному випадку можна використовувати нейронну мережу для навчання, але цей процес може зайняти значний час. Альтернативою є вирізання кожного символу з номерів на всіх зображеннях, що дає певну вибірку символів, яку потрібно класифікувати.

Також можна використовувати готові шрифти, які використовуються українськими автомобільними номерами. Це дозволяє спростити процес розпізнавання, оскільки заздалегідь відомі символи, їх форма і структура.

Щоб навчити шрифти на розпізнавання швидше, було вирішено використовувати алгоритм класифікації k-Nearest Neighbors (k-Найближчих Сусідів).

k-Nearest Neighbors (k-Найближчих Сусідів) є одним з найпростіших алгоритмів класифікації, що використовується в області Машинного Навчання. Завдання класифікації полягає у віднесенні об'єктів до певних класів, які визначаються заздалегідь. Кожен об'єкт представляється у N-вимірному векторному просторі, де кожен вимір визначається за певними критеріями. Наприклад, у випадку класифікації човнів, вимірами можуть бути розмір, співвідношення сторін, колір, довжина, вартість і т.д. У випадку класифікації текстів, використовуються матриці як представлення об'єктів.

Для навчання нам потрібні об'єкти та їх класи. Ми формуємо навчальну вибірку, включаючи об'єкти з їх відповідними класами, яку складаємо вручну відповідно до поставленої задачі. Класифікація може включати більше двох класів, і кожен об'єкт може належати до кількох класів.

Для класифікації кожного об'єкта з навчальної вибірки ми пропускаємо його через наступні етапи: обчислюємо відстань до кожного об'єкта з навчальної вибірки. Вибираємо k найближчих об'єктів з навчальної вибірки, виміряних відстань до яких найменша. Клас об'єкта визначається як клас, котрий найчастіше зустрічається серед k найближчих сусідів.

Отже, для розпізнавання символів ми починаємо з вирізання кожної букви і цифри зі зображень шрифтів. Потім ми приводимо кожен елемент до одного розміру і зберігаємо їх у файловій формі - окремі файли для літер та окремі файли для цифр.

Навчання алгоритму k-Nearest Neighbors вимагає наявності навчальної вибірки, де кожен об'єкт має відповідний клас. В нашому випадку, навчальна вибірка складається з вирізаних символів зі зображень шрифтів і відповідних класів (літери або цифри).

Наступним кроком є використання навченої моделі k-Nearest Neighbors для класифікації символів на зображеннях номерів. Для кожного символу на зображенні ми обчислюємо його відстань до всіх об'єктів у навчальній вибірці. Потім вибираємо k найближчих сусідів символу і визначаємо його клас на основі найпоширенішого класу серед цих сусідів.

Таким чином, використовуючи алгоритм k-Nearest Neighbors, ми можемо класифікувати символи на зображеннях номерів та отримати кінцевий результат у текстовому форматі.

### 3.3 Розробка баз даних

Для розробки баз даних для вашої системи потрібно використовувати відповідні інструменти і технології. У даній роботі використовується SQLite, який є легким вбудовуваним реляційним СУБД.

Основні кроки для розробки баз даних для вашої системи включають:

- 1 Встановлення SQLite: Переконайтеся, що у вас встановлений SQLite на вашому пристрої. Якщо ви використовуєте Python, то більшість постачальників Python включають SQLite в свій розподіл.

- 2 Створення таблиць: Визначте структуру бази даних, яка включатиме таблиці та їх стовпці. В кодї, який ви надали, для бази даних "users.db" створюється таблиця "users" зі стовпцями "username" та "password". Для бази



даних "results.db" створюється таблиця "results" зі стовпцями "image\_path" та "license\_plate".

3 Виконання SQL-запитів: Використовуйте SQL-запити для взаємодії з базою даних. У вашому коді функції `authenticate_user()`, `add_user()` та `recognize_license_plate()` виконують SQL-запити для перевірки авторизації користувача, додавання нового користувача та збереження результатів розпізнавання номера автомобіля.

4 З'єднання з базою даних: Встановіть з'єднання з базою даних, щоб виконувати SQL-запити. У вашому коді використовуються функції `sqlite3.connect()` для встановлення з'єднання з базою даних та `conn.close()` для закриття з'єднання після виконання запитів.

5 Обробка помилок: При роботі з базами даних важливо враховувати можливі помилки, такі як недоступність бази даних або некоректні SQL-запити. Додайте відповідну обробку помилок у ваш код для забезпечення стабільності та надійності системи.

### 3.3.1 База даних "users.db"

Назва бази даних: `users.db`.

Структура бази даних:

Таблиця "users" зі стовпцями:

`username` (тип: текст) - зберігає ім'я користувача

`password` (тип: текст) - зберігає пароль користувача

Функція `authenticate_user(username, password)`:

Використовується для перевірки авторизації користувача в базі даних "users.db".

Встановлює з'єднання з базою даних, виконує SQL-запит, щоб перевірити наявність запису, в якому значення стовпців "username" та "password" відповідають переданим значенням.

Повертає результат перевірки авторизації.

Функція `add_user(username, password)`:

Використовується для додавання нового користувача до бази даних "users.db".

Встановлює з'єднання з базою даних, виконує SQL-запит для вставки нового запису з переданими значеннями "username" та "password".

Зберігає зміни у базі даних.

### 3.3.1 База даних "results.db":

База даних

Назва бази даних: results.db

Структура бази даних:

Таблиця "results" зі стовпцями:

`image_path` (тип: текст) - зберігає шлях до зображення

`license_plate` (тип: текст) - зберігає розпізнаний номер автомобіля

Функція `recognize_license_plate()`:

Використовується для розпізнавання номера автомобіля на зображенні та збереження результату в базу даних "results.db".

Зображення, з якого потрібно розпізнати номер, відображається у вікні за допомогою бібліотеки Matplotlib.

Після розпізнавання номера автомобіля, результат (номер) зберігається у базі даних "results.db".

На рисунку 3.10 наведено зв'язки з нашою БД.

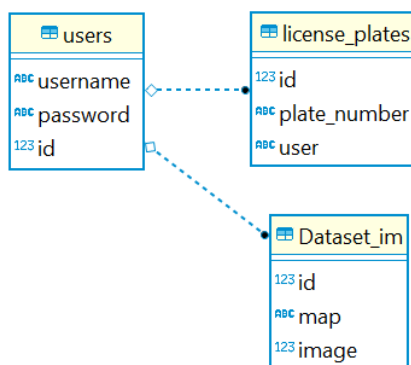


Рис. 3.10. Побудова зв'язків БД

Діаграми, які використовуються для опису поведінки компонентів системи, включають достатньо велику кількість паралельних процесів. На діаграмі діяльності кожен стан відповідає за виконання певних простих операцій. Перехід до наступного стану відбувається лише після завершення поточної операції. За такою логікою, діаграма діяльності може бути розглянута як приватний випадок діаграми станів. Якщо ми хочемо візуалізувати особливості реалізації операцій класів та надати алгоритми їх виконання, можна використати діаграми діяльності. Крім того, діаграма розгортання (рис. 3.11) надає графічне представлення програмного забезпечення і допомагає раціонально організувати його компоненти, що впливає на продуктивність і безпеку системи.



Рис. 3.11. Діаграма станів

На рисунку 3.12 наведено результат зареєстрованих користувачів БД.

username	password
admin	admin
admin	admin
oleg	oleg
admim	admin

Рис. 3.12. Структура заповнених даних

### 3.4 Розробка та дослідження програмного застосунку

У нашій програмі ми використовували бібліотеку PyQt5 Python [12] для створення інтерфейсу. На початку програми було створено головне вікно за допомогою класу `QMainWindow`. У конструкторі цього класу ми встановлювали заголовок вікна за допомогою методу `setWindowTitle`.

Далі ми визначили метод `initUI`, який ініціалізував елементи інтерфейсу. В цьому методі ми створили різні елементи, такі як `QLabel`, `QPushButton` і `QLineEdit`, і встановили їхні властивості, такі як текст, положення і обробники подій.

Наприклад, ми створили `QLabel` з назвою "Оберіть зображення" і встановили його положення за допомогою методу `move`. Так само ми створили кнопки "Відкрити зображення", "Розпізнати номер" і "Авторизація" і встановили їх положення.

Для кожної кнопки ми визначили обробник подій, який виконує певні дії при натисканні кнопки. Наприклад, при натисканні кнопки "Відкрити зображення" викликається метод `open_image`, який відкриває діалогове вікно для вибору зображення.

Ми також використовували методи `setEnabled` для блокування або розблокування кнопок залежно від стану програми. Наприклад, кнопки "Відкрити зображення" і "Розпізнати номер" були заблоковані до моменту успішної авторизації користувача.

Щоб показати вікно програми, ми створили об'єкт `QApplication`, створили головне вікно `MainWindow`, викликали метод `show` для відображення вікна та запустили головний цикл за допомогою `app.exec()`.

На рисунку 3.13 представлено головний інтерфейс програми.

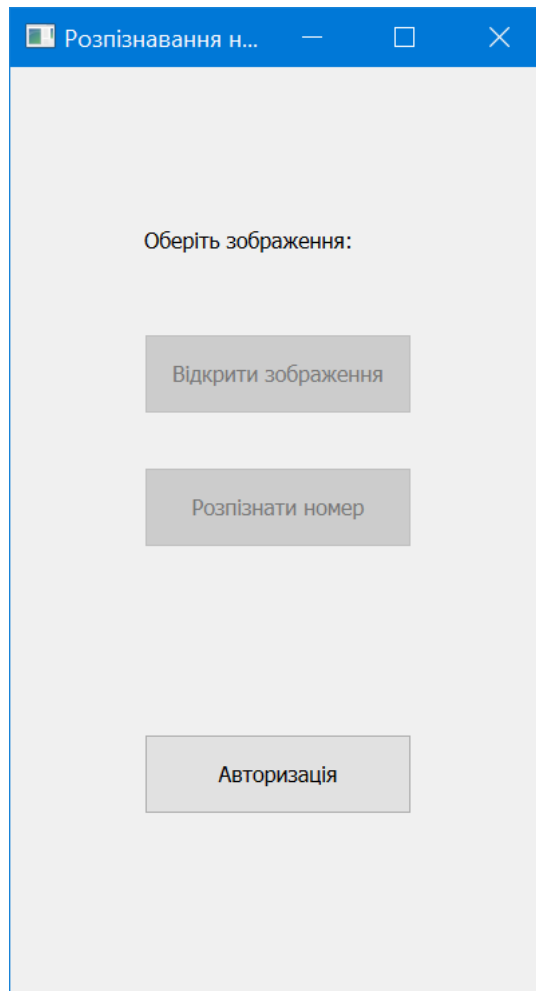


Рис. 3.13. Головне вікно програми

Далі ми створили вікно авторизації за допомогою класу `QDialog`, який є підкласом вікна відображення діалогового вікна в `PyQt5`.

1 У конструкторі класу `LoginWindow` ми виконали наступні дії:

2 Встановили заголовок вікна за допомогою методу `setWindowTitle`.

3 Створили елементи інтерфейсу, такі як `QLabel` для відображення тексту "Ім'я користувача" і "Пароль", і `QLineEdit` для введення ім'я користувача і пароля.

Створили кнопки "Увійти" і "Зареєструватися" за допомогою класу `QPushButton`.

Після створення елементів інтерфейсу ми встановили їх положення за допомогою методу `move`, щоб розташувати їх на вікні. Результат наведено на рис.3.14.

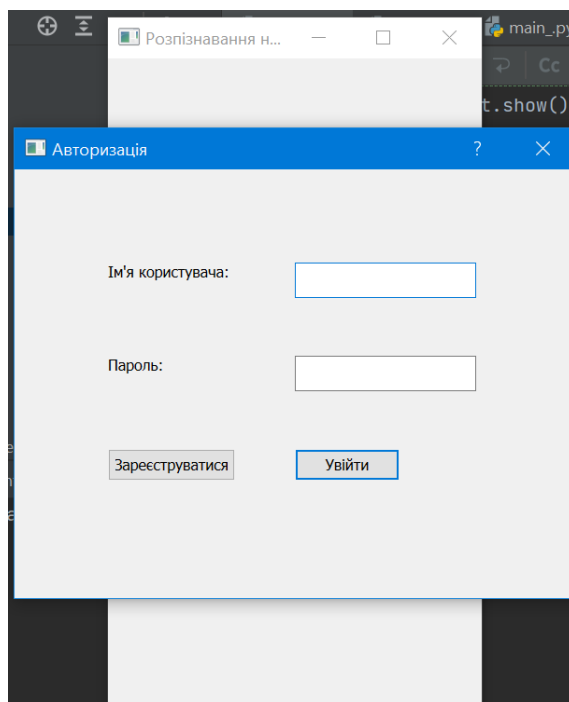


Рис. 3.14. Вікно «Авторизація»

Для кнопки "Увійти" ми встановили обробник подій `clicked`, який викликає метод `login`. Аналогічно, для кнопки "Зареєструватися" ми встановили обробник подій `clicked`, який викликає метод `register` (рис.3.15).

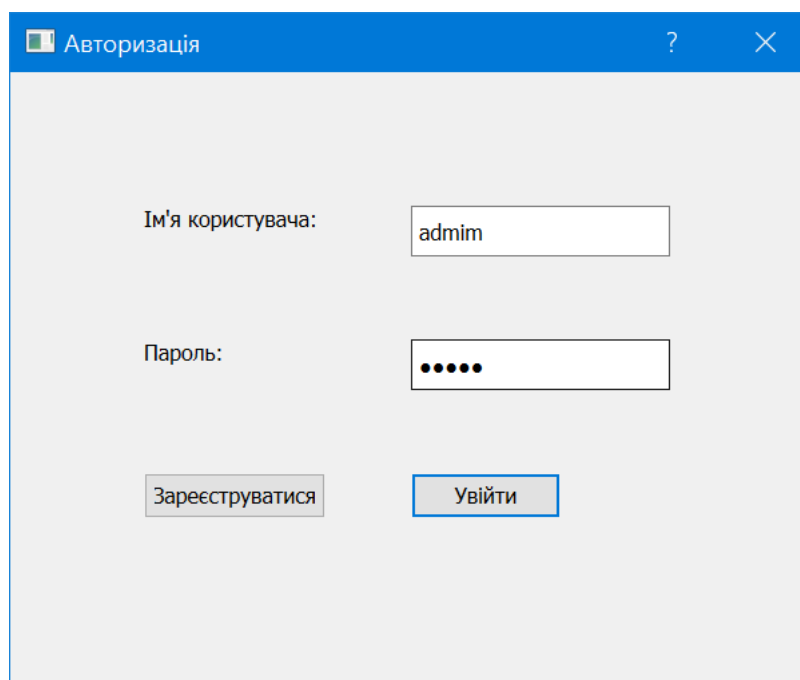


Рис. 3.15. Заповнення даних

Метод `login` перевіряє, чи введені ім'я користувача і пароль відповідають значенням в базі даних. Для цього ми використали об'єкт `sqlite3.connect` для підключення до бази даних, виконали запит до таблиці `users` з використанням введених даних, і викликали метод `fetchone` для отримання результату. Якщо результат не є `None`, то авторизація вважається успішною і метод `assert` викликається для закриття вікна авторизації.

Метод `register` додає нового користувача до бази даних. Ми використовували той самий підхід з підключенням до бази даних і виконанням запиту `INSERT` для додавання нового рядка до таблиці `users`.

При успішній авторизації вікно авторизації закривається, і в головному вікні програми кнопки "Відкрити зображення" і "Розпізнати зображення» (рис.3.16).

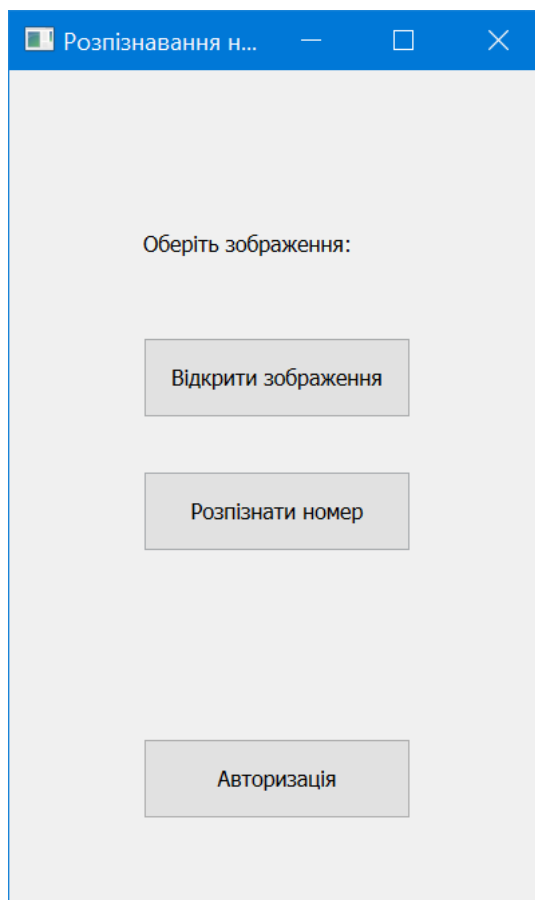


Рис. 3.16. Активація кнопок при успішній авторизації

Таким чином, для розуміння взаємодії користувачів з програмою представимо на концептуальному рівні у вигляді діаграму прецедентів (рис.3.17).

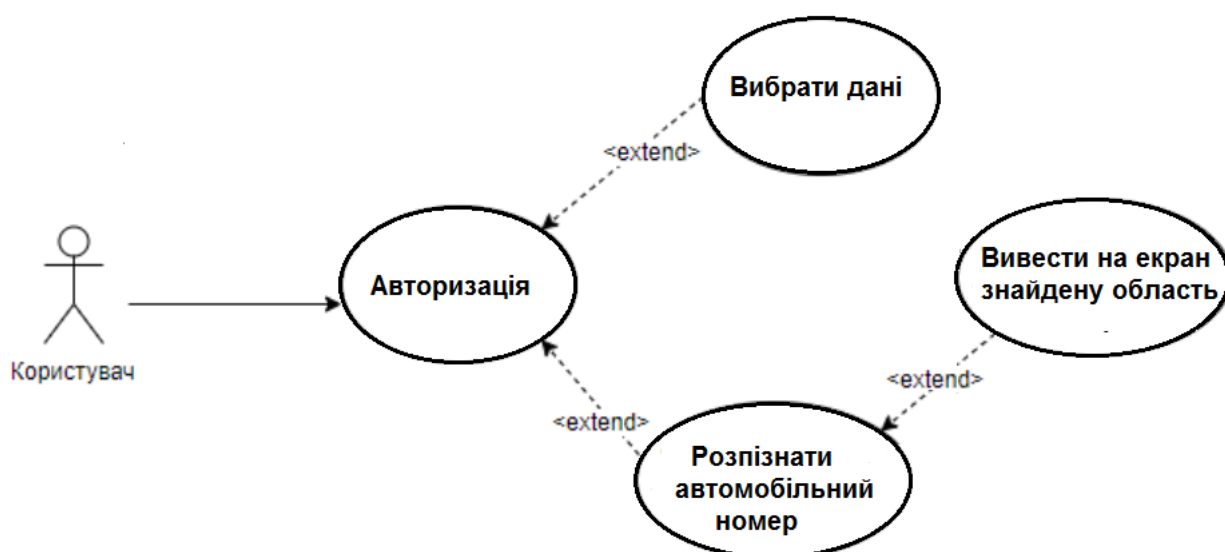


Рис. 3.17. Діаграма варіантів використання програмного застосунку  
Користувач обирає дані, натиснувши на кнопку «Відкрити зображення», відкриється директорія, де користувач обирає дані (рис.3.18).

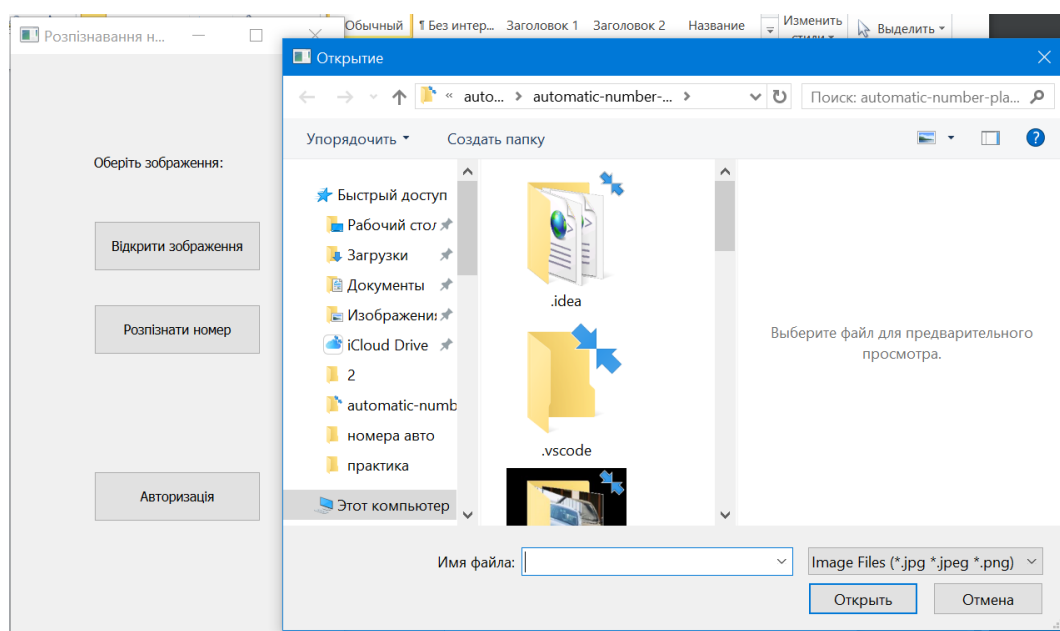


Рис. 3.18. Вибір даних

Завершальним є робота користувача з кнопкою «Розпізнати номер», результат наведено на рис.3.19.



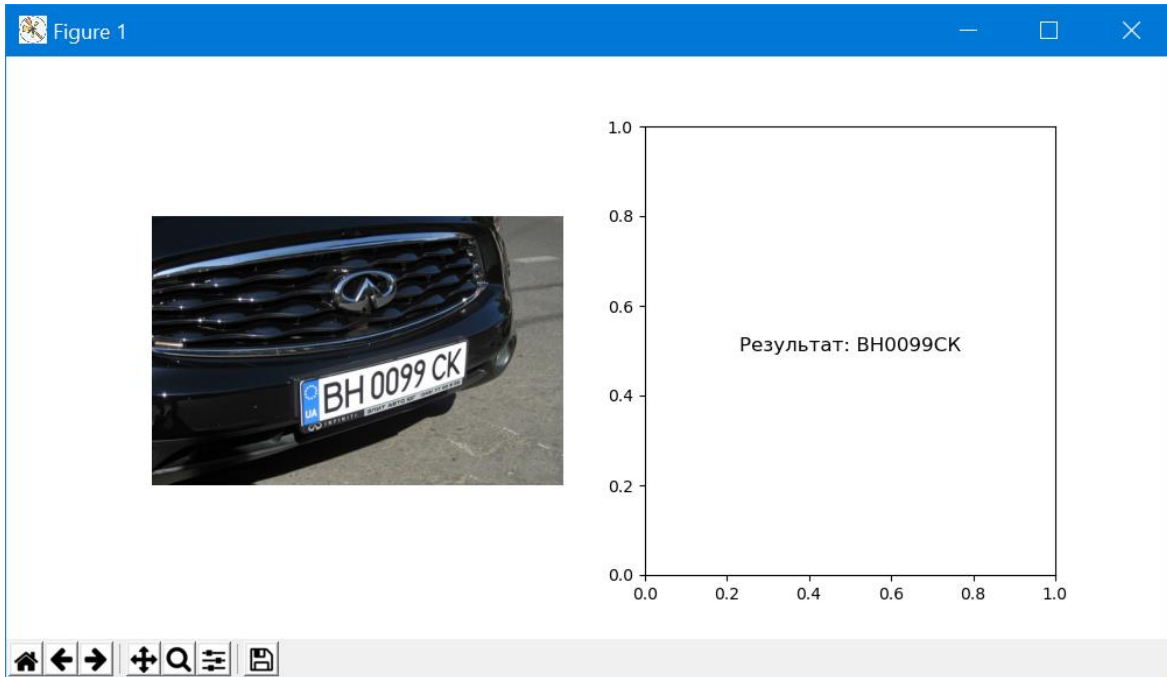


Рис. 3.19. Результат розпізнавання

На рисунку 3.20 представлена діаграма послідовності, яка ілюструє процес розпізнавання автомобільних номерів у випадку успішного розпізнавання.

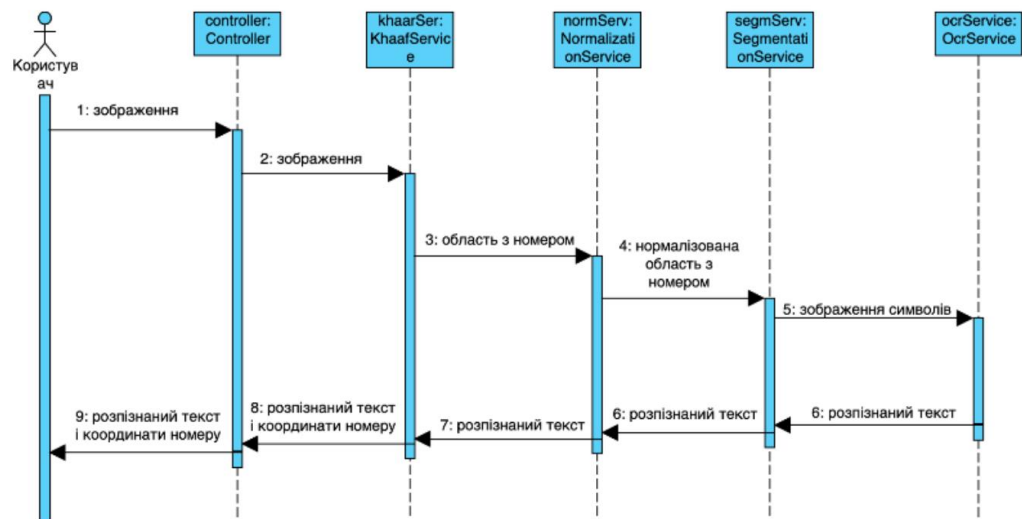


Рис. 3.20. Діаграма послідовності дій

Під час тестування програми було виявлено кілька недоліків, які впливають на її ефективність і точність розпізнавання автомобільних номерів. Основні недоліки включають:

1 Обмежена працездатність у нічний час: Система не здатна адекватно розпізнавати номери автомобілів на фотографіях, знятих у темряві. Це може бути пов'язано з обмеженнями алгоритму розпізнавання або недостатньою якістю зображення в умовах недостатнього освітлення. Результат наведено на рис.3.21.

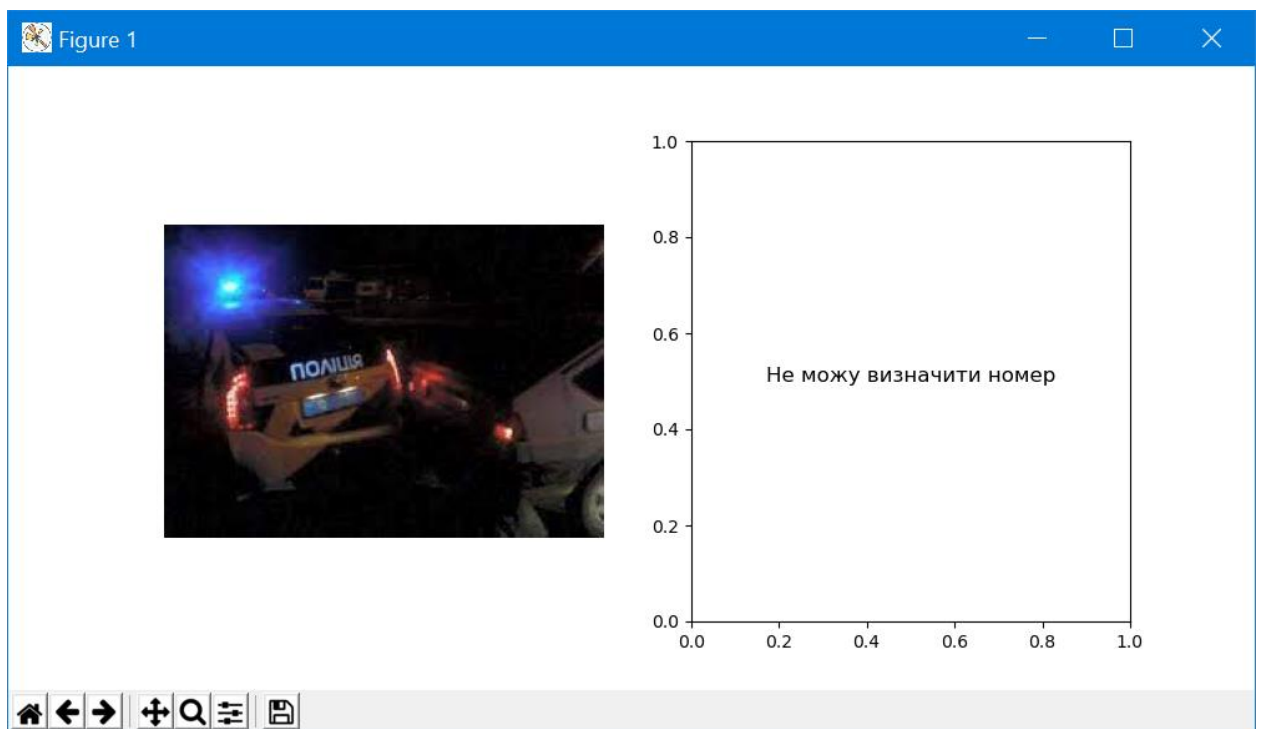


Рис. 3.21. Розпізнавання номера авто у нічний час

2 Неспроможність розпізнати номери на нечітких фото: Система має проблеми з розпізнаванням номерів на фотографіях низької роздільної здатності або з великою кількістю шуму. Це може призводити до неправильного розпізнавання або невдачі взагалі.

3 Віддалені автомобілі не розпізнаються: Система має обмеження щодо відстані, з якої можна успішно розпізнати номер автомобіля. Віддалені

транспортні засоби можуть бути нерозпізнані або розпізнані з низькою точністю (рис.3.22).

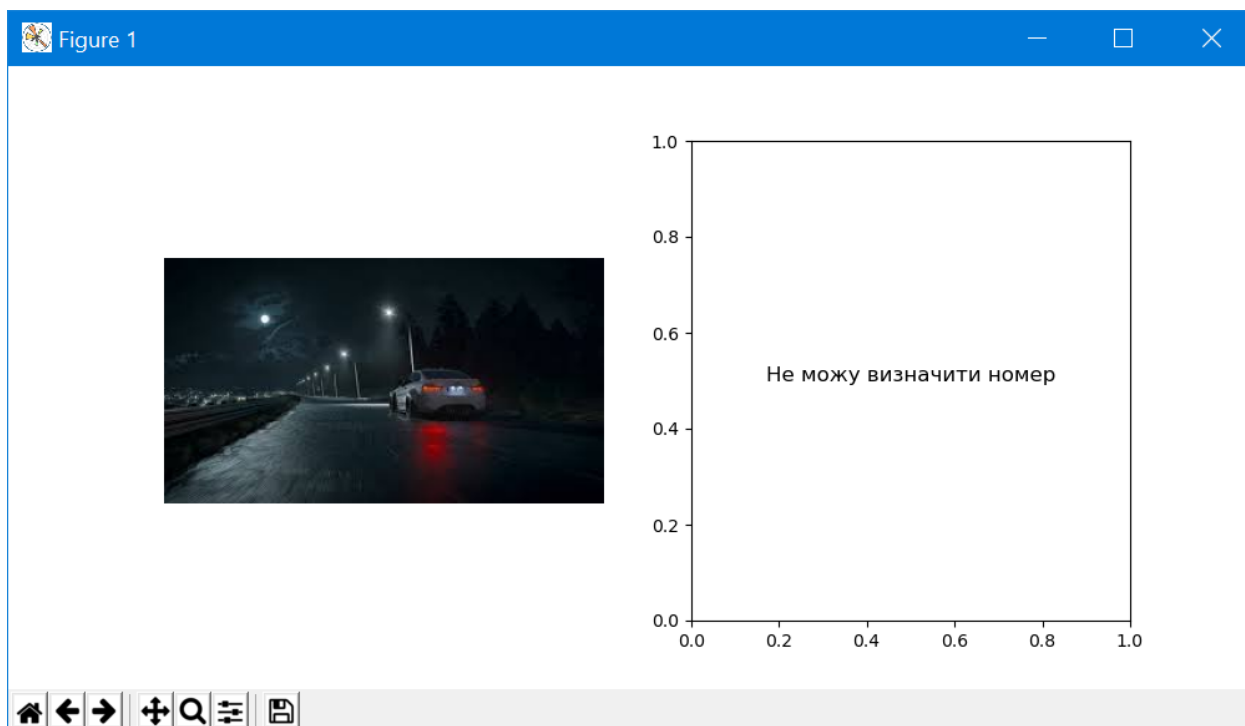


Рис. 3.22. Розпізнавання номерів при великій відстані

Ці недоліки можуть бути вирішені шляхом вдосконалення алгоритмів розпізнавання, вдосконалення обробки зображень і використання більш потужних алгоритмів машинного навчання. Також можливими покращеннями можуть бути використання спеціального освітлення для поліпшення якості зображення та застосування оптичних засобів для збільшення розпізнаваної відстані.

## ВИСНОВКИ

В результаті написання кваліфікаційної роботи було проведено дослідження та розроблено застосунок для розпізнавання номерів автотранспорту з використанням бібліотеки OpenCV. Застосунок дозволяє користувачам завантажувати зображення автомобільних номерів, виконувати їх обробку та розпізнавання, і відображати результат на екрані.

У результаті проведених експериментів та тестування було отримано задовільні результати з точністю розпізнавання номерів. Застосунок продемонстрував здатність визначати номери автомобілів на чітких та близьких зображеннях з високою точністю.

Однак, під час тестування було виявлено деякі недоліки, такі як обмежена працездатність у нічний час, неспроможність розпізнати номери на нечітких фотографіях та нездатність розпізнати номери на віддалених транспортних засобах. Ці недоліки можуть потребувати подальших досліджень та вдосконалень алгоритмів розпізнавання та обробки зображень.

В цілому, розроблений застосунок є початковим кроком у розробці системи розпізнавання номерів автотранспорту. Він використовує потужні можливості бібліотеки OpenCV та демонструє принципи обробки зображень та розпізнавання номерів. Подальші дослідження та розвиток можуть включати в себе вдосконалення алгоритмів розпізнавання, врахування різних умов зйомки та адаптацію системи до роботи в нічний час, а також розширення діапазону розпізнавання для віддалених транспортних засобів.

Застосунок може бути корисним у багатьох сферах, таких як парковки, дорожні контрольні пункти, системи безпеки та багато інших. Він може сприяти автоматизації та поліпшенню процесу розпізнавання номерів, що забезпечує більш швидку та ефективну обробку даних.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Палієнко В. Г. дослідження методів класифікації веб-документів на основі machine learning //ББК 32 I 74. 2017. – С. 146.
2. Shashank Arokar, Visual Character Recognition using Artificial Neural; MGM's College of Engineering and Technology, University of Mumbai, India 2014.
3. Y. Lecun, B. Boser, J. S. Denker, D. Henderson, Backpropagation Applied to Handwritten Zip Code Recognition, Neural Computation, 2017. 541-551p.
4. Tamkang Journal of Science and Engineering, Vol. 13, No. 4, pp. 433- 442 (2010) «A Real-Time Mobile Vehicle License Plate Detection and Recognition», Kuo-Ming Hungand Ching-Tang Hsieh.
5. HikVision [Електронний ресурс]:– Режим доступу: <https://www.hikvision.com/>.
6. Axis Communications [Електронний ресурс]:– Режим доступу: <https://www.axis.com/>.
7. ParkHelp [Електронний ресурс]:– Режим доступу: <https://www.parkhelp.com/>
8. Scheidt & Bachmann [Електронний ресурс]:– Режим доступу: <https://www.scheidt-bachmann.de/>
9. SKIDATA [Електронний ресурс]:– Режим доступу <https://www.skidata.com/>
10. Caleb Henry. Japan mulls seven-satellite QZSS system as a GPS backup. SpaceNews : website. URL: <https://spacenews.com/japan-mulls-seven-satellite-qzssystem-as-a-gps-backup/>
11. Joseph Howse, Joe Minichino. Learning OpenCV 4 Computer Vision with Python", Packt Publishing, 2018, 622 p.
12. Wes McKinney. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython" , O'Reilly Media, 2017, 544 p.

## Додаток А. Фрагмент лістингу програми

```

import cv2
import imghdr
import pytesseract

def recognize_license_plate(image_path):
    # Завантажуємо зображення
    image = cv2.imread(image_path)

    # Перетворюємо зображення в чорно-біле
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    # Застосовуємо фільтр для видалення шуму та полегшення розпізнавання
    gray = cv2.bilateralFilter(gray, 11, 17, 17)

    # Використовуємо класифікатор Хаара для знаходження регіонів із номерними знаками
    cascade_path = "path_to_cascade_xml_file.xml"
    cascade = cv2.CascadeClassifier(cascade_path)
    license_plates = cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5, minSize=(25, 25))

    # Проходимося по знайдених регіонах з номерними знаками
    for (x, y, w, h) in license_plates:
        # Виділяємо регіон з номерним знаком
        plate = gray[y:y + h, x:x + w]

        # Використовуємо бібліотеку Tesseract для розпізнавання тексту
        plate_text = pytesseract.image_to_string(plate, config='--psm 7')

        # Виводимо результати розпізнавання
        print("Розпізнаний номерний знак:", plate_text)

        # Відображаємо прямокутник навколо номерного знака на вихідному зображенні
        cv2.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 3)

        # Виводимо зображення з виділеними номерними знаками
        cv2.imshow("Розпізнавання номерів автотранспорту", image)
        cv2.waitKey(0)

    # Закриваємо вікно
    cv2.destroyAllWindows()

# Виклик функції для розпізнавання номерів автотранспорту на зображенні
image_path = "path_to_image_file.jpg" # Потрібно вказати шлях до файлу зображення
recognize_license_plate(image_path)

def recognize_license_plate(image, plate):
    # Виділяємо регіон з номерним знаком
    (x, y, w, h) = plate
    plate_image = image[y:y + h, x:x + w]

    # Використовуємо бібліотеку Tesseract для розпізнавання тексту
    plate_text = pytesseract.image_to_string(plate_image, config='--psm 7')

    return plate_text

def draw_license_plates(image, license_plates):

```

```
# Проходимося по знайдених регіонах з номерними знаками
for (x, y, w, h) in license_plates:
    # Відображаємо прямокутник навколо номерного знака на вихідному зображенні
    cv2.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 3)

# Виводимо зображення з виділеними номерними знаками
cv2.imshow("Розпізнавання номерів автотранспорту", image)
cv2.waitKey(0)
cv2.destroyAllWindows()

def main(image_path, cascade_path):
    # Завантажуємо зображення
    image = cv2.imread(image_path)

    # Попередня обробка зображення
    preprocessed_image = preprocess_image(image)

    # Знаходження номерних знаків на зображенні
    license_plates = detect_license_plates(preprocessed_image, cascade_path)

    # Розпізнавання номерних знаків
    for plate in license_plates:
        plate_text = recognize_license_plate(preprocessed_image, plate)
        print("Розпізнаний номерний знак:", plate_text)
image_path = "path_to_image_file.jpg" # Потрібно вказати шлях до файлу зображення
cascade_path = "path_to_cascade_xml_file.xml" # Потрібно вказати шлях до файлу каскадного
класифікатора

#Виклик головної функції
main(image_path, cascade_path)
```