

БАКАЛАВРСЬКА РОБОТА

БР. ІІІ - 38.00.00.000 ІІЗ

Група ІІз-21-1к

Пасічник Максим

2025

Івано-Франківський національний технічний університет нафти і газу
Інститут інформаційних технологій
Кафедра інженерії програмного забезпечення

Пасічник Максим Ігорович

(прізвище, ім'я, по батькові)

УДК 004.942
(індекс)

БАКАЛАВРСЬКА РОБОТА

Розробка системи розпізнавання номерних знаків автомобілів
(назва роботи)

Інженерія програмного забезпечення

(назва освітньої програми)

121– Інженерія програмного забезпечення

(шифр і назва спеціальності)

Робота містить результати власних досліджень, використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело:

Здобувач освітнього ступеня Пасічник Максим Ігорович
(підпис, ініціали та прізвище здобувача)

Науковий керівник Бандура Вікторія Валеріївна, доцент
(підпис, прізвище, ім'я, по батькові, науковий ступінь, вчене звання керівника)

Допущено до захисту
Завідувач кафедри

доц. Бандура В.В.
(посада) (підпис) (дата) (ініціали та прізвище)

Івано-Франківський національний технічний університет нафти і газу

Інститут, факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Ступінь вищої освіти бакалавр

Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. кафедрою _____

ІПЗ

доцент. _____

В.В. Бандура

“ _____ ” _____ 2025 р.

ЗАВДАННЯ НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТОВІ

Пасічник Максим Ігорович

(прізвище, ім'я, по-батькові)

1. Тема проекту (роботи) "Розробка системи розпізнавання номерних знаків автомобілів"

керівник проекту (роботи) к.т.н., доцент Бандура Вікторія Валеріївна

затвержені наказом вищого навчального закладу від “ 28 ” квітня 2025 р. № 264/7

2. Строк подання студентом проекту (роботи) 10 червня 2025 р.

3. Вихідні дані до проекту (роботи) Результати і матеріали отримані під час проходження переддипломної практики

4. Зміст розрахунково - пояснювальної записки(перелік питань, які потрібно розробити)

1 Загальні принципи та інструменти розробки ігор

2 Засоби та методи розробки ігор

3 Unity-ігра: вплив на користувачів та її аналіз

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

1. Бінаризація образу номерного знаку (рис 1.1, ст 16).

2. Головна сторінка сайту «НомерОк». (рис 1.3, ст 21).

3. Приклади роботи бібліотеки OpenCV.(рис 2.4, ст. 39)

4. Параметри детектора розпізнавання символів (рис 3.4, ст 50).

5. Папка CarPictures61 (рис 4.1, ст 61).

6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

2. Дата видачі завдання 2025 р.

Керівник _____

(підпис)

Завдання прийняв до виконання _____

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту	Примітка
1	Визначення та обґрунтування теми роботи	15.02.2025	виконано
2	Огляд існуючих концепцій, рішень та сервісів в даній області	25.02.2025	виконано
3	Побудова моделі або алгоритму власного рішення	15.03.2025	виконано
4	Документування реалізації власного оригінального рішення вибраними засобами	25.04.2025	виконано
5	Оформлення пояснювальної записки бакалаврської роботи	10.06.2025	виконано

Студент _____

(підпис)

Керівник роботи _____

(підпис)

АНОТАЦІЯ

Бакалаврська робота містить 82 сторінок, 7 таблиць, 33 рисунків, список використаних джерел із 29 найменування,

Метою роботи є розробка системи розпізнавання номерних знаків автомобілів на базі бібліотеки OpenCV.

Об'єктом дослідження: процес розробки системи розпізнавання номерних знаків автомобілів на базі бібліотеки OpenCV і мови програмування Python.

Предмет дослідження: здійснення огляд та аналіз програмних систем розпізнавання номерних знаків, проведення дослідження над актуальністю розроблення та впровадження даної системи розпізнавання номерних знаків автомобілів.

У першому розділі - проведено та представлено детальний аналіз існуючих аналогів, проаналізовано алгоритми та методи розпізнавання символів номерних знаків.

У другому розділі - представлено розробку структурної схеми програмної системи для розпізнавання автомобільних знаків. Обґрунтовану вибір інструментальних засобів розробки, а також вибір алгоритму для розпізнавання символів на номерних знаках.

У третьому розділі - реалізовано розробку алгоритму функціонування системи ідентифікаційних номерних знаків, дописати згідно методички.

У четвертому розділі йдеться про технічну реалізацію системи автоматичного розпізнавання автомобільних номерних знаків.

П'ятий розділ описує практичні аспекти реалізації системи в реальному середовищі:

КЛЮЧОВІ СЛОВА: РОЗПІЗНАВАННЯ НОМЕРНИХ ЗНАКІВ, КОМП'ЮТЕРНИЙ ЗІР, OPENCV, ОБРОБКА ЗОБРАЖЕНЬ, OCR, EASYOCR, TESSERACT, МАШИННЕ НАВЧАННЯ, CANNY, SOBEL, PYCHARM, PYTHON, АВТОМАТИЗОВАНА ІДЕНТИФІКАЦІЯ.

ABSTRACT

The bachelor's thesis contains 82 pages, 7 tables, 33 figures, a list of 29 references,

The purpose of the work is to develop a car license plate recognition system based on the OpenCV library.

Object of research: the process of developing a car license plate recognition system based on the OpenCV library and the Python programming language.

Subject of the study: review and analysis of license plate recognition software systems, research on the relevance of the development and implementation of this license plate recognition system.

In the first section, a detailed analysis of existing analogues is carried out and presented, algorithms and methods for recognizing license plate characters are analyzed.

In the second section, the development of a structural diagram of a software system for recognizing license plates is presented. The choice of development tools, as well as the choice of an algorithm for recognizing license plate characters, is substantiated.

In the third section, the development of the algorithm for the functioning of the system of identification license plates is implemented, completed according to the methodology.

The fourth section discusses the technical implementation of the automatic license plate recognition system.

The fifth section describes the practical aspects of implementing the system in a real environment:

KEYWORDS: LICENSE PLATE RECOGNITION, COMPUTER VISION, OPENCV, IMAGE PROCESSING, OCR, EASYOCR, TESSERACT, MACHINE LEARNING, CANNY EDGE DETECTION, SOBEL OPERATOR, PYCHARM, PYTHON, AUTOMATED IDENTIFICATION.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. ОГЛЯД ТА АНАЛІЗ СИСТЕМ РОЗПІЗНАВАННЯ НОМЕРНИХ ЗНАКІВ	11
1.1. Актуальність розробки та впровадження систем розпізнавання номерних знаків автомобілів	11
1.2. Аналіз алгоритмів і методів розпізнавання символів номерних знаків	12
1.3. Огляд існуючих систем розпізнавання номерних знаків автомобілів	16
1.3.1. Програма розпізнавання автомобільних номерів «НомерОК»	18
1.3.2. Програма розпізнавання автомобільних номерів «Відеоконтроль-Рубіж»	21
1.3.3. Система OpenALPR	23
1.4. Постановка завдання	26
1.5. Висновки по розділу	27
РОЗДІЛ 2. ВИБІР ТА ОБҐРУНТУВАННЯ ПРОГРАМНИХ ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ СИСТЕМИ	29
2.1. Розробка структурної схеми програмної системи	29
2.2. Вибір інструментів розробки	30
2.3. Вибір алгоритму розпізнавання символів номерних знаків	34
2.4. Вибір бібліотеки комп'ютерного зору OpenCV	35
2.5. Вибір засобів розробки бази даних	38
2.6. Висновки по розділу	39
РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА ТЕСТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ	40
3.1. Розробка алгоритму функціонування системи розпізнавання номерних знаків	40

					БР.ІІІ – 38.00.00.000 ПЗ						
Змн.	Арк.	№ докум.	Підпис	Дата	Розробка системи розпізнавання номерних знаків автомобілів Пояснювальна записка			Літ.	Арк.	Акрушів	
Розроб.		Пасічник М.І.								8	
Перевір.		Бандура В.В.									
Реценз.		Піх В.Я.									
Н. Контр.		Піх М.М.									
Затверд.		Бандура В.В.			ІФНТУНГ ПЗ-21-1К						

3.2. Розробка алгоритму ідентифікації транспортних засобів	42
3.3. Розробка інтерфейсу користувача	45
3.4. Реалізація алгоритму роботи програмного забезпечення	47
3.5. Тестування функціоналу програмної системи	52
3.6. Висновки по розділу	56
РОЗДІЛ 4. РОЗРОБКА СИСТЕМИ РОЗПІЗНАВАННЯ НОМЕРНИХ ЗНАКІВ.....	57
4.1. Основні результати дослідження	57
4.2. Інтеграція компонентів системи	64
4.3. Підсумки та напрямки вдосконалення системи розпізнавання	67
4.4. Висновки по розділу	68
РОЗДІЛ 5. ВПРОВАДЖЕННЯ СИСТЕМИ РОЗПІЗНАВАННЯ АВТОМОБІЛЬНИХ НОМЕРІВ.....	70
5.1. Підготовка до інтеграції системи	70
5.2. Налаштування системи на етапі впровадження	71
5.3. Особливості практичного використання системи	74
5.4. Висновки по розділу	75
ВИСНОВКИ	76
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	77
БІБЛІОГРАФІЧНА ДОВІДКА	

ВСТУП

У сучасних умовах розвитку технологій комп'ютерного зору автоматизація розпізнавання номерних знаків автомобілів є ключовим фактором підвищення ефективності в різних галузях, таких як транспортна логістика, безпека та паркувальні системи. Щоденно камери відеоспостереження фіксують велику кількість зображень транспортних засобів, що потребує швидкої та точної обробки для ідентифікації номерних знаків. Відповідно, виникає потреба в автоматизованих системах, які дозволяють ефективно розпізнавати, обробляти та інтегрувати дані з номерних знаків у інформаційні системи для подальшого використання.

Актуальність роботи

У контексті стрімкого зростання кількості транспортних засобів і необхідності їх автоматизованої ідентифікації системи розпізнавання номерних знаків (ANPR) набувають особливої ваги. Ручна обробка даних із зображень номерних знаків є трудомісткою, схильною до помилок і не відповідає вимогам масштабованості сучасних систем. Застосування технологій комп'ютерного зору, зокрема бібліотеки OpenCV, дозволяє автоматизувати процеси розпізнавання, підвищити точність і швидкість обробки, а також забезпечити інтеграцію з базами даних для ефективного управління інформацією. Дослідження в цій сфері є актуальним з наукової та практичної точок зору, оскільки сприяє оптимізації процесів у сферах безпеки, паркування та транспортного контролю.

Метою роботи є розробка ефективної системи розпізнавання номерних знаків автомобілів на базі бібліотеки OpenCV, яка забезпечує автоматичне виявлення, обробку та збереження даних із зображень номерних знаків у реальному часі.

Для реалізації цієї мети у роботі застосовано бібліотеку комп'ютерного зору OpenCV, сучасні алгоритми розпізнавання символів, а також технології для

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

створення баз даних і користувацького інтерфейсу, що забезпечують стабільність і зручність використання системи.

Завдання дослідження:

1. Проаналізувати предметну область розпізнавання номерних знаків автомобілів та сучасні методи їх обробки.
2. Вивчити існуючі алгоритми, технології та системи розпізнавання, такі як «НомерОК», «Відеоконтроль-Рубіж» і OpenALPR.
3. Розробити архітектуру системи розпізнавання номерних знаків на базі OpenCV.
4. Реалізувати функціональні компоненти системи, включаючи алгоритми розпізнавання та ідентифікації транспортних засобів.
5. Розробити користувацький інтерфейс і засоби інтеграції з базами даних.
6. Провести тестування та оцінити ефективність роботи системи.

Об’єкт дослідження: Процеси автоматизованого розпізнавання номерних знаків автомобілів.

Предмет дослідження: Методи та інструменти реалізації систем розпізнавання номерних знаків на базі бібліотеки OpenCV, включаючи алгоритми обробки зображень і інтеграцію з інформаційними системами.

Методи дослідження:

- Аналітичний метод — для вивчення предметної області та технологій розпізнавання;
- Системний аналіз — для розробки архітектури системи;
- Моделювання — для створення діаграм алгоритмів і взаємодій;
- Програмна реалізація — для створення функціональних компонентів і інтерфейсу;
- Експериментальний метод — для тестування та оцінки системи.

Наукова новизна

Удосконалено підхід до автоматизованого розпізнавання номерних знаків шляхом інтеграції бібліотеки OpenCV з оптимальними алгоритмами обробки

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

зображень, що забезпечує високу точність і швидкість розпізнавання в реальному часі.

Практичне застосування

Розроблена система може бути використана в системах безпеки, паркувальних комплексах, транспортних логістичних платформах і правоохоронних органах для автоматизованого розпізнавання та обробки даних номерних знаків.

Бакалаврська робота містить 82 сторінок, 7 таблиць, 29 рисунків, список використаних джерел із 30 найменування,

					БР.ІІІ - 38.00.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

РОЗДІЛ 1. ОГЛЯД ТА АНАЛІЗ ПРОГРАМНИХ СИСТЕМ РОЗПІЗНАВАННЯ НОМЕРНИХ ЗНАКІВ

1.1. Актуальність розроблення та впровадження систем розпізнавання номерних знаків автомобілів.

Технологія розпізнавання номерних знаків - важлива технологія для автоматизації процесу реєстрації транспортних засобів, моніторингу порушень ПДР та забезпечення безпеки автотранспорту. Сьогодні існує велика кількість програмних комплексів, які забезпечують розпізнавання номерних знаків.

Однією з найвідоміших систем є OpenALPR (Open Automatic License Plate Recognition), яка використовує нейронні мережі для розпізнавання номерних знаків. OpenALPR - це відкрита програма, яка має дуже високу точність розпізнавання. Ця система підтримує розпізнавання номерних знаків різних країн, включаючи США, Канаду, Європу, Азію та Австралію.

Ще однією популярною програмною системою є Tesseract OCR - бібліотека розпізнавання тексту, розроблена корпорацією Google. Вона також використовується для розпізнавання автомобільних номерів і має високу точність розпізнавання. Вона також підтримує розпізнавання тексту з різних мов, що дозволяє використовувати її для розпізнавання номерних знаків з різних країн.

Ще одна програмна система - ANPR (Automatic Number Plate Recognition), розроблена компанією ARH Inc., яка також використовує нейронні мережі з метою розпізнавання номерних знаків. Ця система підтримує розпізнавання номерних знаків різних країн і має досить високу швидкість розпізнавання.

Проте, незважаючи на те, що ці програмні системи мають високу точність і швидкість розпізнавання, в деяких випадках вони можуть давати помилкові результати, зокрема, у випадках, коли номерний знак знаходиться в нестандартному положенні або має пошкодження чи дефекти. Іншим важливим фактором для ефективної роботи програмних систем розпізнавання номерних

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

позначок є якість зображення, яка може впливати на точність і швидкість розпізнавання.

Окрім того, сьогодні стрімко розвиваються технології штучного інтелекту та машинного навчання, що дає можливість створювати ще точніші та швидші системи розпізнавання номерних знаків. Наприклад, з'являються системи, які використовують для навчання глибокі нейромережі та великі масиви даних.

Загалом, програмні системи розпізнавання автомобільних номерів є важливою технологією для автоматизації дорожнього руху та безпеки на дорогах. Вони мають високу точність і швидкість ідентифікації, але в деяких ситуаціях можуть давати помилкові результати. З розвитком технологій машинного навчання та штучного інтелекту можна очікувати подальшого розвитку та вдосконалення цих систем.

Крім зазначених систем, важливою бібліотекою для створення власних рішень у сфері комп'ютерного зору є **OpenCV (Open Source Computer Vision Library)**. OpenCV — це одна з найпоширеніших бібліотек з відкритим кодом для обробки зображень та відео, яка широко використовується для створення систем автоматичного розпізнавання номерних знаків. Вона забезпечує великий набір інструментів для роботи з відеопотоком у реальному часі, виявлення контурів, сегментації об'єктів, фільтрації зображень тощо.

Завдяки підтримці мов програмування, таких як Python, C++ та Java, OpenCV легко інтегрується в різні програмні платформи та дозволяє розробляти гнучкі, налаштовувані системи. Її поєднання з іншими бібліотеками, зокрема Tesseract OCR для розпізнавання символів, дає змогу створити повноцінну систему розпізнавання номерних знаків із можливістю локалізації, сегментації та інтерпретації символів номеру.

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

1.2 Аналіз алгоритмів та методів розпізнавання символів номерних знаків.

Розпізнавання символів автомобільних номерів є важливим завданням в області оброблення зображень та комп'ютерного зору. Існує багато алгоритмів і методів розпізнавання символів номерних знаків, і кожен з них має як свої переваги, так і недоліки. Розглянемо деякі з них нижче:

– Метод шаблонів. Цей метод полягає у порівнянні зразка номерного знака зі зображенням на вхідному зображенні. Зразки можуть бути зібрані за допомогою фотографування чи сканування реальних номерних знаків. Проте, недоліком цього методу є неефективність у випадках, коли номерні знаки несуть незвичні шрифти, написи, та інші додаткові елементи.

– Метод розпізнавання символів за зразками передбачає, що для всіх можливих зображень символів існують шаблони для всіх можливих образів символів. Ухвалення рішення про те, чи належить поточне зображення, чи належить зображення символу, отримане на етапі сегментації, отриманому з номерного знаку на етапі сегментації, до певного класу символів здійснюється за наступними критеріями мінімум (максимум) визначеної метрики схожості зображення символу та його шаблону. На даний момент поточне зображення символу номера відноситься до того класу символів, значення кореляції якого з шаблоном є максимальним. Таким чином, метрикою в даному випадку є коефіцієнт кореляції випадку (точніше абсолютне значення показника кореляції).

$K =$

$$\left| \frac{\sum(X[i]-\bar{X})(Y[i]-\bar{Y})}{\sqrt{\sum(X[i]-\bar{X})^2}\sqrt{\sum(Y[i]-\bar{Y})^2}} \right|, \quad (1.1)$$

де $X[i]$ та $Y[i]$ - поточні положення елементів пікселів елементів зображення та шаблону відповідно, а середні значення елементів пікселів

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

зображень та шаблону відповідно.

– Метод нейронних мереж. Метод нейронних мереж заснований на створенні штучних нейронних мереж, які навчаються розпізнавати символи номерних знаків. Для цього, спочатку необхідно навчити мережу розпізнавати окремі символи, а потім використовувати її для розпізнавання повних номерних знаків. Цей метод зазвичай працює добре, але вимагає значного обсягу навчальних даних та велику кількість обчислювальних ресурсів. Насамперед, виконується бінаризація зображення пластинки, спираючись на уявлення про тип чистих чисел, зображено на рисунку 1.1.

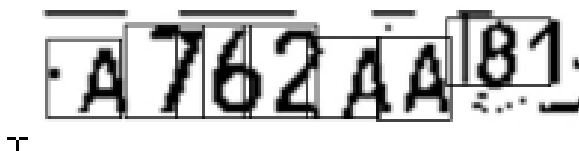


Рисунок 1.1 – Бінаризація образу номерного знаку.

Робота нейромережевого алгоритму спрощується до визначення залежності вагового множника зв'язку між двома нейронами від кількості прикладів нейронами від кількості прикладів, що їх підтверджують. цю залежність. Цільова функції згідно з цим алгоритмом, повинна зводити до мінімуму квадрат помилки при навчанні.

$$\min \sum_i (T_i - y_i)^2, \quad (1.2)$$

де, T_i - встановлене значення вихідної характеристики для i -го прикладу;
 y_i - розраховане значення вихідної характеристики.

Підготовка до тренування нейронної мережі:

– Навчальна вибірка генерується шляхом накладання спотвореного зображення на оригінал зображень, символів шрифту, що використовується в номерних знаків.

– Символи шрифту накладаються на зображення підганяються під заданий розмір пікселів.

До зображень застосовують такі операції спотворення:

– Масштабування (без збереження пропорцій оригінального зображення).

– Обрізання країв зображення знизу, зверху, праворуч, ліворуч (максимум на 2 пікселі).

– Поворот в діапазоні $[-15^\circ, 15^\circ]$ з кроком.

– Розмивання випадкових прямокутних областей оригіналу або всього зображення.

– "Шум" зображення символів.

Перевага цього методу обробки вибірки полягає в тому, що відсутність необхідності вручну проставляти мітки класів зображень, оскільки ми завжди знаємо, зображення якого персонажа ми в даний момент обробляємо. Використання алгоритму навчання нейронної мережі зі зворотним розповсюдженням помилки.

Ключовим аспектом роботи нейронної мережі є її навчання, яке полягає у встановленні зв'язків (синапсів) між нейронами та визначення сили цих зв'язків (вагові коефіцієнти).

Алгоритми навчання нейронних мереж спрощуються до того, що визначають залежність вагового коефіцієнта зв'язку між двома нейронами від кількості прикладів, які підтверджують цю залежність, зображено на рисунку 1.2.

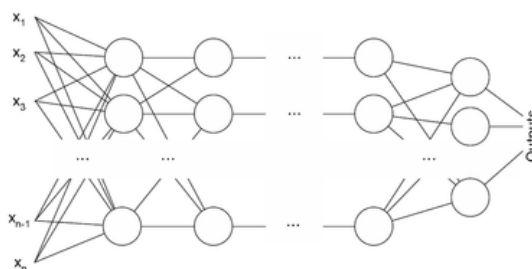


Рисунок 1.2 – Мультишарова нейронна мережа зі зворотним поширенням помилок.

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

Згідно з цим принципом, нейронна мережа повинна "навчатися", порівнюючи її із зразком з тисяч зображень номерних знаків.

– Метод знаходження контурів. Цей метод полягає у визначенні контурів номерних знаків та використанні характеристик цих контурів для розпізнавання символів. Для цього використовуються алгоритми обробки зображень, такі як Canny Edge Detection та Hough Transform. Цей метод досить ефективний, але вимагає досить складних алгоритмів обробки зображень та може бути досить чутливим до шуму на вхідному зображенні.

– Метод каскад. Цей метод базується на використанні алгоритмів машинного навчання, таких як AdaBoost, для створення каскадів класифікаторів, які поступово відкидають неправильні варіанти розпізнавання символів. Кожен класифікатор може використовувати різні характеристики для розпізнавання символів, такі як форму та контур. Цей метод досить ефективний та стійкий до шуму та змін у світловому середовищі.

Кожний з цих методів має свої переваги та недоліки, тому рішення про вибір методу залежить від конкретного завдання розпізнавання символів на номерному знаку. Так, наприклад, шаблонний метод може бути ефективним, коли потрібно розпізнати символи на номерних знаках зі стандартним шрифтом, а каскадний метод може бути корисним, коли номерні знаки мають різноманітний шрифт і додаткові елементи.

1.3 Огляд та аналіз відомих систем розпізнавання номерних знаків автомобілів.

Системи розпізнавання номерних знаків автомобілів (англ. Automatic License Plate Recognition, ALPR) використовуються для автоматичного зчитування номерних знаків з фотографій або відео. Ці системи зазвичай складаються з камер, які фіксують зображення номерних знаків, а потім

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

використовують алгоритми розпізнавання для отримання текстової інформації з цих зображень.

Нижче наведено огляд декількох відомих систем розпізнавання номерних знаків автомобілів:

1. OpenALPR - це відкрита система розпізнавання номерних знаків автомобілів, яка може працювати як в режимі онлайн, так і офлайн. Система підтримує розпізнавання номерних знаків з більш ніж 70 країн світу та має високу точність розпізнавання.

2. Tattile - це італійська система розпізнавання номерних знаків, яка може бути встановлена як на стаціонарні камери, так і на автомобілі. Система має високу швидкість роботи та точність розпізнавання.

3. Vigilant Solutions - ця система розпізнавання номерних знаків використовується в поліції та інших правоохоронних органах США. Вона має високу точність розпізнавання та може працювати як в режимі онлайн, так і офлайн.

4. Bosch - це система розпізнавання номерних знаків, яка може бути встановлена як на стаціонарні камери, так і на автомобілі. Система має високу швидкість роботи та точність розпізнавання та підтримує розпізнавання номерних знаків з більш ніж 60 країн світу.

5. PlateSmart - це система розпізнавання номерних знаків, яка може працювати як в режимі онлайн, так і офлайн. Вона має високу точність розпізнавання та може підтримувати розпізнавання номерних знаків з більш ніж 80 країн світу.

6. Hikvision - це китайська система розпізнавання номерних знаків, яка може бути встановлена як на стаціонарні камери, так і на автомобілі. Система має високу швидкість роботи та точність розпізнавання.

7. Neurala - це система розпізнавання номерних знаків, яка використовує нейронні мережі для розпізнавання зображень. Система має високу точність розпізнавання та може працювати на вбудованих пристроях з обмеженими ресурсами.

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

Функціонал даних систем для розпізнавання номерних знаків:

- Розпізнавання номерних знаків з фотографій та відео.
- Автоматичне визначення реєстраційного номера автомобіля.
- Збереження даних про номерні знаки та їх час та місце фіксації.
- Відстеження руху автомобілів на основі номерних знаків.

Інтеграція з базами даних, що містять інформацію про викрадені або шукані автомобілі.

Одним з найбільш поширених застосувань систем розпізнавання номерних знаків є забезпечення безпеки на дорогах та допомога правоохоронним органам у виконанні їхніх обов'язків. Вони також можуть бути використані для збору даних про трафік, контролю швидкості та паркування, а також для управління паркомісцями та управління транспорту.

1.3.1 Програма розпізнавання автомобільних номерів «НомерОК».

"НомерОК" - це програма для розпізнавання номерних знаків автомобілів, розроблена в Україні. Вона дозволяє автоматично розпізнавати номерні знаки на фото та відео, що дозволяє використовувати її в різних сферах: від реєстрації в'їзду до паркінгу до забезпечення безпеки на дорозі та виконанні правоохоронних функцій [1].

Основні функції "НомерОК":

- розпізнавання номерних знаків на фото та відео;
- підтримка розпізнавання номерних знаків з більш ніж 20 країн світу;
- можливість налаштування режимів розпізнавання залежно від умов зйомки;
- збереження даних про номерні знаки та їх час та місце фіксації;
- можливість інтеграції з іншими системами або програмами.

Програма "НомерОК" може бути корисною для власників автопарків головна сторінка, зображено на рисунку 1.3, паркінгів та інших місць, де потрібно вести облік проходження транспортних засобів. Вона також може бути

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

використана для забезпечення безпеки на дорогах та допомоги правоохоронним органам у виконанні їхніх обов'язків.



Рисунок 1.3 – Головна сторінка сайту «НомерОк»

Програма "НомерОК" має безкоштовну версію для некомерційного використання, зображено на рисунку 1.6, та платні версії з різними можливостями та обсягом розпізнавання. Також доступні АРІ для інтеграції з іншими програмами та сервісами. Головні переваги і недоліки програми «НомерОк», представлено в таблиці 1.4.

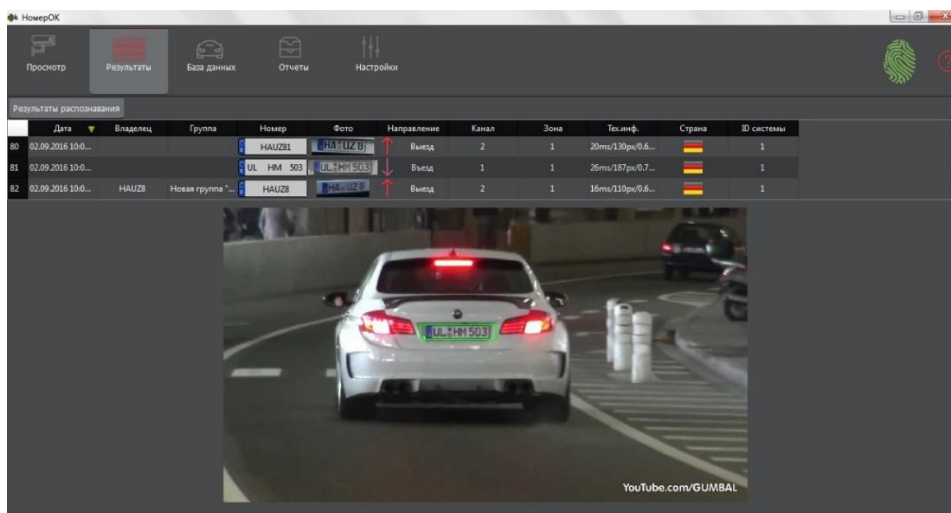


Рисунок 1.4 – Демо версія «НомерОк».

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

Переваги і недоліки «НомерОк».

Переваги	Недоліки
Широкі можливості: програма може розпізнавати номерні знаки на фото та відео, підтримує розпізнавання номерних знаків з більш ніж 20 країн світу, має можливість налаштування режимів розпізнавання залежно від умов зйомки та інтеграції з іншими програмами.	Платні версії: програма має платні версії з різними можливостями та обсягом розпізнавання, що може бути недоцільним для користувачів з обмеженим бюджетом
Висока точність розпізнавання: програма має високу точність розпізнавання номерних знаків, що дозволяє використовувати її для реєстрації в'їзду на паркінг, виконання правоохоронних функцій та інших завдань, де важлива точність розпізнавання	Залежність від умов зйомки: як і будь-яка програма розпізнавання номерних знаків, "НомерОК" залежить від умов зйомки, таких як освітлення та кут зйомки, що може впливати на точність розпізнавання
Збереження даних: програма зберігає дані про номерні знаки та їх час та місце фіксації, що дозволяє вести облік проходження транспортних засобів	Обмеження за кількістю номерних знаків: програма має обмеження за кількістю номерних знаків, які може розпізнати одночасно, що може бути недостатньо для деяких застосувань

Мінімальні вимоги до сервера:

процесор - Intel Core i5 4440

відеокарта (інтегрована) - OpenGL 2.0 оперативна пам'ять - DDR 4Gb

жорсткий диск - HDD 500Gb ОС - Windows XP/7/8/10, server 8 (32/64 bit) [2].

1.3.2 Програма розпізнавання автомобільних номерів «Відеоконтроль-Рубіж».

Відеоконтроль-Рубіж – апаратно-програмний комплекс цілодобового контролю стану дорожнього руху на вулицях населених пунктів та автодорогах.

Комплекс призначений для відеоспостереження та відеофіксації, автоматичного аналізу транспортного потоку та розпізнавання державних

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

номерних знаків транспортних засобів у реальному часі, збереження отриманої інформації у власній базі даних та пошуку розпізнаного номера у підключених до системи реєстрах.

До складу системи «Відеоконтроль-Рубіж» входить мережа рубежів контролю, які складаються з камер відеоспостереження та інфрачервоних прожекторів, вимірювачів швидкості й обчислювальних модулів. Також до складу системи входять канали передавання даних, автоматизовані робочі місця операторів на стаціонарних та пересувних постах ДАІ, територіальні сервери, робочі місця на центральних постах і спеціалізоване програмне забезпечення, встановлене на комп'ютерах системи.

Комплекс, зображено на рисунку 1.5, розроблено інженерами української компанії «ОЛЛІ Транс». Тестування системи проходило у 2010 році на Столичному шосе у Києві та на стаціонарному посту «Биківня» на виїзді з Києва у бік Чернігова. Після прийняття остаточного рішення систему розгорнуто по всій Україні. Від початку закупівель у липні 2011 року по січень 2012 року комплекси було встановлено у 46 точках, перелік яких надав портал «Наші Гроші». Станом на листопад 2013 року встановлено вже 95 стаціонарних комплексів, розпочато виробництво мобільних версій.



Рисунок 1.5 – Інтерфейс системи «ВІДЕОКОНТРОЛЬ-Рубіж».

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

Комплекс «Відеоконтроль-Рубіж» автоматично розпізнає однорядкові номери та зберігає у архіві розпізнані номери та зображення транспортних засобів. Для знаків, які можна візуально розрізнити, імовірність розпізнавання не нижча за 90 % незалежно від часу доби. Програмне забезпечення комплексу проводить пошук розпізнаного номера по базах розшуку та оперативних орієнтуваннях. При збігу номера з занесеним у базу, в тому числі з нечіткою маскою пошуку, комплекс видає сповіщення оператору для затримання транспорту. Сигнал помилки роботи при неможливості розпізнати забруднений чи неправильно закріплений номерний знак також може бути причиною зупинки автомобіля.

«Рубіж» також проводить моніторинг стану дорожнього руху у реальному часі та автоматично генерує статистичні звіти, необхідні для аналізу транспортних потоків.

Комплекс може автоматично відстежувати порушення правил дорожнього руху. Для контролю швидкості руху до системи приєднуються мікрохвильові або лазерні вимірювачі швидкості. Також комплекс може сприймати сигнали від світлофора та розміщеної під дорожнім полотном магнітної петлі-детектора наявності автомобіля. Аналізом отриманих зображень виявляються порушення дорожньої розмітки, наприклад, рух по зустрічній смузі, перетин суцільної лінії, порушення рядності руху, зупинка у недозволеному місці та не пропускання пішохода на нерегульованому пішохідному переході, проїзд на сигнал світлофора, що забороняє рух.

При необхідності комплекс може керувати зовнішніми виконавчими пристроями — світлофорами, шлагбаумами, світловими та звуковими сигналами. Команди на зовнішні пристрої, наприклад, на перемикання стану світлофора, видаються при виявленні транспортного засобу, державний номер якого знаходиться у відповідній зовнішній базі даних

Комплекси «Відеоконтроль-Рубіж» випускаються у стаціонарному та мобільному виконаннях для розміщення на постах ДАІ та у патрульних автомобілях. За допомогою каналів передавання даних комплекси об'єднуються

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24

у загальнодержавну розподілену систему відеоспостереження, відеозапису, протоколювання проходження транспортними засобами рубежів контролю та перевірки розпізнаних номерів по пошукових реєстрах. При спрацюванні системи інформація надходить на центральний пульт у Києві

Статистика Державної автомобільної інспекції Міністерства внутрішніх справ України свідчить про високу ефективність роботи комплексів «Відеоконтроль-Рубіж». Так, вже у 2010 році під час дослідної експлуатації експериментального зразка на Столичному шосе у Києві комплекс виявив 28 транспортних засобів, що перебували у розшуку, понад 400 — власники яких були позбавлені права керування і 85 тих, що розшукувалися виконавчою службою як майно боржників.

1.3.2 Система OpenALPR.

OpenALPR - це система розпізнавання номерних знаків з відкритим кодом, що розроблена компанією OpenALPR Technology, Inc. Система використовує глибоке навчання та машинне навчання для розпізнавання номерів автомобілів з відео або зображень її робота зображена на рисунку 1.6.

OpenALPR може бути використаний для різних цілей, таких як контроль швидкості, безпека на дорозі, паркування та моніторинг приватної власності. Він може бути інтегрований з камерами відеоспостереження та іншими системами, що дозволяє автоматизувати процес розпізнавання номерних знаків, зображено на рисунку 1.7.

OpenALPR працює в режимі реального часу та може розпізнавати номерні знаки в різних країнах світу, включаючи Україну. Він має високу точність розпізнавання та швидкість обробки, що робить його досить ефективним рішенням для автоматизації різних процесів, пов'язаних з номерними знаками автомобілів зображено на рисунку 1.8.

Ключові переваги і недоліки системи для розпізнавання знаків OpenALPR, представлені в таблиці 1.4.

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

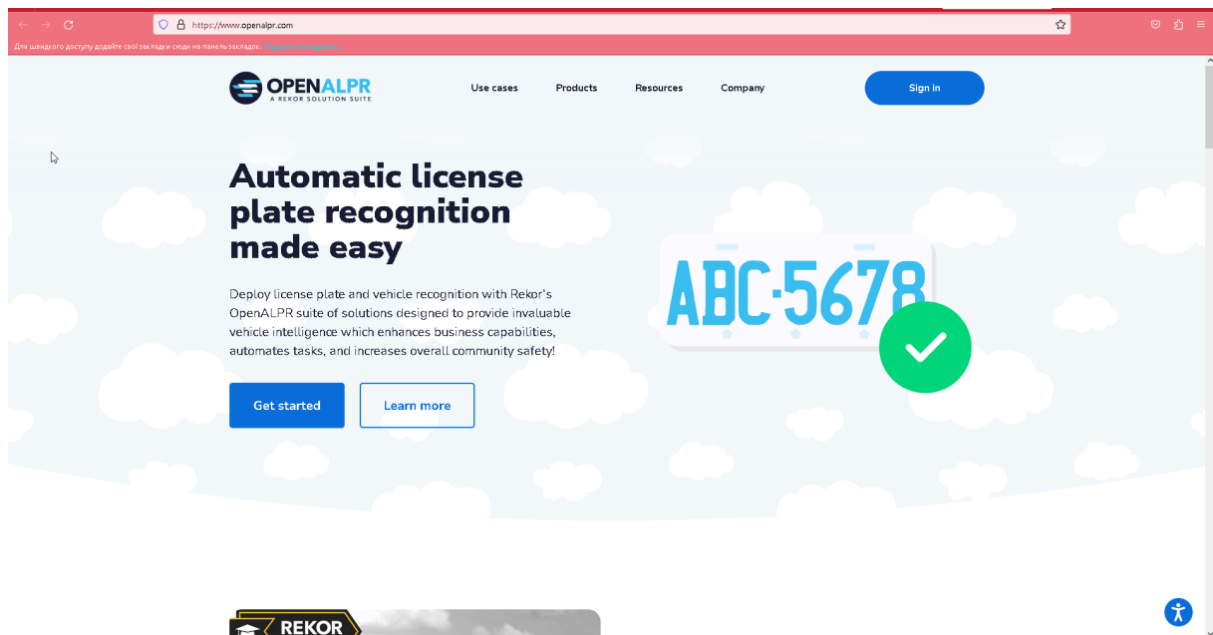


Рисунок 1.6 – Головна сторінка сайту OpenALPR.

Таблиця 1.4

Переваги і недоліки «OpenALPR».

Переваги	Недоліки
Відкритий код та можливість редагування вихідного коду	Вимагає відповідної апаратної бази для працювання в режимі реального часу
Висока точність розпізнавання номерних знаків	Потребує встановлення та налаштування перед використанням
Швидкість обробки зображень та відео	Призводить до збору та зберігання великої кількості персональних даних
Можливість інтеграції з різними системами та обладнанням	Можливість помилок розпізнавання в разі неякісного зображення або освітлення
Підтримка номерних знаків різних країн світу, включаючи Україну	Потребує наявності стабільного зв'язку з Інтернетом для використання деяких функцій
Висока ефективність та автоматизація процесів	_____

Зазначені переваги та недоліки системи OpenALPR є загальними та можуть мати вагому чи меншу вагомість в залежності від конкретного використання та встановлення системи [3].

Наприклад, для використання системи на стаціонарній постійній основі, можливість відкритого коду та наявність можливості редагування вихідного

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

коду може бути значущою перевагою, тоді як в інших випадках, таких як використання системи в мобільних додатках, можливість редагування вихідного коду може не мати особливого значення [4].

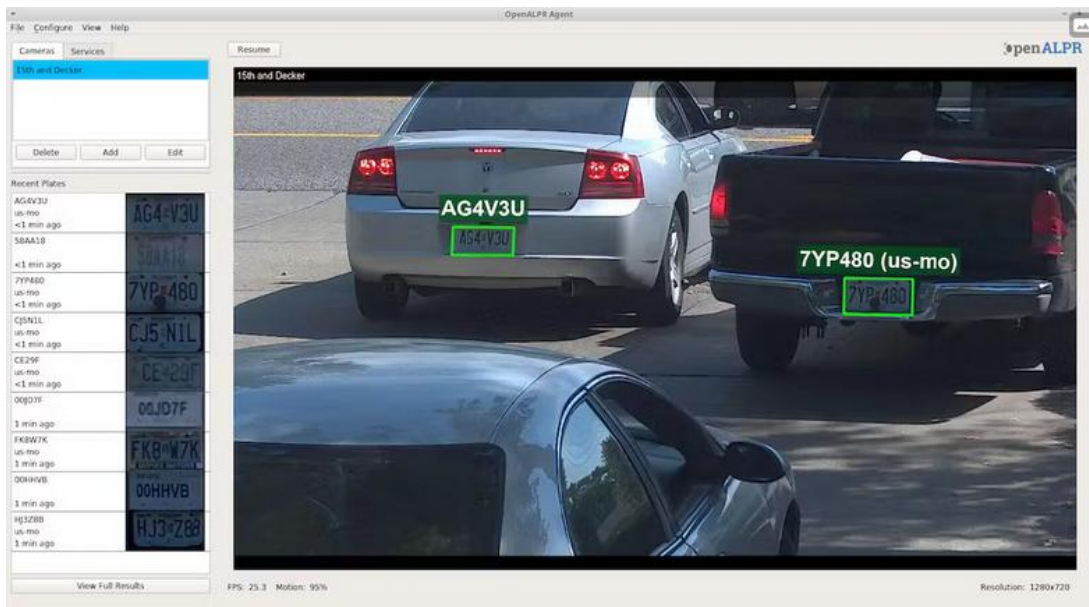


Рисунок 1.7 – Приклад роботи OpenALPR.

Site	Camera	Plate Number	Vehicle	Direction	Confidence	Time
St Louis Office	Main Entrance	S292312	Black Chevrolet Sedan	Entering	94.90	1:35:12 pm
St Louis Office	Main Entrance	UF1X6D	White Lexus Sedan		94.96	1:35:11 pm
St Louis Office	Main Entrance	D3998EV	Silver		93.78	1:35:07 pm
St Louis Office	Main Entrance	6YNB77	Silver		94.44	1:35:04 pm
St Louis Office	Main Entrance	MD2R2V	Red C		94.82	1:34:59 pm
St Louis Office	Main Entrance	4AP309	Silver		94.99	1:34:56 pm
St Louis Office	Main Entrance	2191666	White		88.70	1:34:52 pm
St Louis Office	Main Entrance	3SR196	Blue F		94.84	1:34:50 pm
St Louis Office	Main Entrance	2FV896	White Hyundai SUV	Exiting	94.97	1:34:46 pm

Рисунок 1.8 – Приклад роботи OpenALPR.

1.4 Постановка завдання.

Об'єкт проекту: система розпізнавання номерних знаків автомобілів на базі бібліотеки OpenCV.

Предмет проекту: розробка системи розпізнавання номерних знаків автомобілів на базі бібліотеки OpenCV з застосування мови програмування Python.

Мета проекту полягає у розробці та впровадженні системи розпізнавання номерних знаків автомобілів.

Завдання проекту:

1. Визначити актуальність розроблення даної системи.
2. Провести аналіз алгоритмів та методів розпізнавання номерних знаків.
3. Провести аналіз систем аналогів, знайти їхні переваги і недоліки.
4. Розробити структурну схему програмної системи.
5. Розробити базу даних.
6. Розробити інтерфейс користувача.
7. Реалізувати роботу алгоритму програмного забезпечення.
8. Провести тестування програмної системи.

1.5 Висновки по розділу.

У першому розділі було проведено детальний аналіз існуючих програмних систем і технологій розпізнавання номерних знаків автомобілів. Розглянуто актуальність розробки подібних систем у контексті сучасних вимог до безпеки дорожнього руху, автоматизації паркування, контролю доступу та відеоспостереження.

Огляд таких систем, як **OpenALPR**, **Tesseract OCR**, **ANPR**, **НомерОК**, **Відеоконтроль-Рубіж** та **OpenCV**, дозволив зробити висновок, що найбільш ефективними на сьогодні є гібридні підходи, які поєднують класичні методи

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

обробки зображень з алгоритмами штучного інтелекту, зокрема нейронними мережами. Ці підходи забезпечують високу точність, швидкість роботи та здатність адаптуватися до різних умов зйомки.

Проаналізовано основні методи розпізнавання символів: метод шаблонів, метод кореляції, метод нейронних мереж, метод контурного аналізу та каскадний метод. Встановлено, що найперспективнішими є нейронні мережі, які демонструють найвищу точність за умови наявності якісної тренувальної вибірки. Проте класичні методи, такі як шаблонне порівняння чи каскадні класифікатори, залишаються актуальними для швидких, ресурсоефективних рішень.

Вивчення систем-аналогів дозволило визначити їхні **переваги** (висока точність, можливість інтеграції, робота в реальному часі) та **недоліки** (високі вимоги до апаратного забезпечення, чутливість до якості зображення, платна ліцензія тощо). Це створює передумови для розробки власного рішення, яке б поєднало переваги існуючих систем та врахувало їхні недоліки.

На завершення, у підрозділі 1.4 було сформульовано мету, об'єкт, предмет і завдання дипломного проєкту, які визначають подальші етапи розробки системи розпізнавання номерних знаків на базі бібліотеки OpenCV із використанням мови програмування Python.

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		29

РОЗДІЛ 2. ВИБІР ТА ОБГРУНТУВАННЯ ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ СИСТЕМИ

2.1 Розроблення структурної схеми програмної системи.

Загальна структура системи розпізнавання (зчитування) номерних знаків транспортних засобів подана на рисунку 2.1.

Система містить наступні апаратні та програмні модулі:

- відеокамеру;
- плату відеозахоплення зображення;
- програмний модуль локалізації номерного знаку;
- програмний модуль розпізнавання номерного знаку;
- зовнішню базу даних.

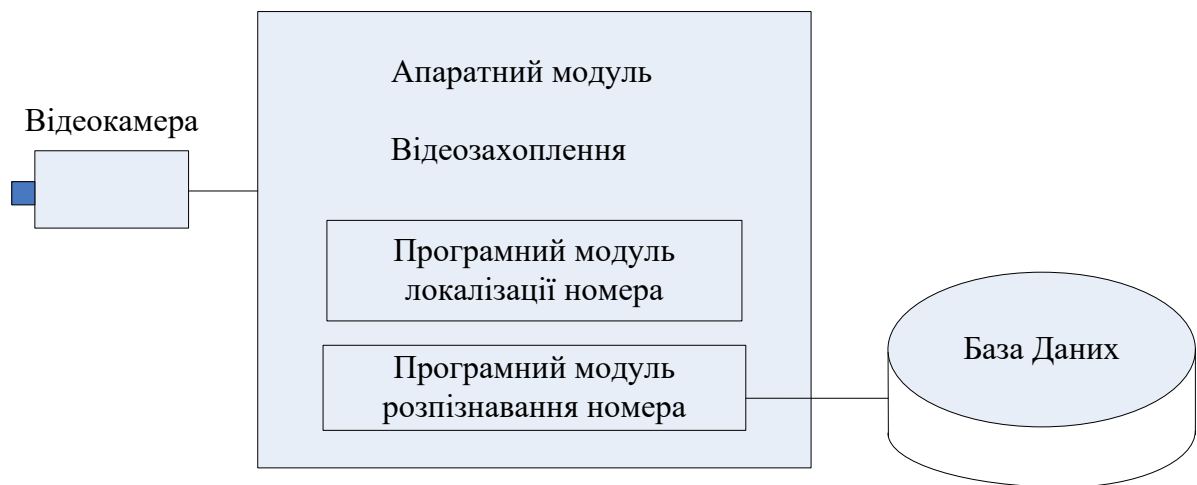


Рисунок 2.1 – Загальна структура системи розпізнавання номерних знаків.

Зображення з відеокамери поступає на вхід алгоритму локалізації. Програмний детектор знаходить активний (рухомий) транспортний засіб, після чого спеціальним алгоритмом визначається положення номерного знаку. Наступним кроком виділений номер розпізнається програмою оптичного розпізнавання символів [5].

База даних, залежно від поставлених перед системою завдань, може

мати різноманітну схему. Основними завданнями є: фіксація номера, дати і часу появи транспортного засобу у полі зору відеокамери; запис стоп-кадру автомобіля або вирізання відео; перевірка номерного знаку на факт вчинення порушень правил дорожнього руху, викрадення та інших зловживань.

Відеокамери залежно від завдань, що використовуються в системі розпізнавання, діляться на групи:

- безкорпусні камери – мають, як правило, розміри з сірникову коробку, поставляються виробником без корпусу і можуть вмонтовуватися у відповідні предмети;

- мініатюрні відеокамери мають невеликі розміри, власний корпус і можуть бути прикріплені як на кронштейни, так і на поворотні пристрої;

- швидкісні купольні відеокамери оснащені, як правило, швидкісним поворотним пристроєм, завдяки якому відеокамери можуть обертатися із швидкістю до 400 град/сек в горизонтальній площині і повертатися у вертикальну площину до 160 град;

- чорно-білі відеокамери представляють найбільш популярний клас камер відеоспостереження, оскільки вони мають невисоку ціну і, як правило, складають основу системи відеоспостереження. Залежно від призначення, чорно-білі відеокамери діляться на моделі для роботи усередині приміщення і зовні, які використовуються у системах розпізнавання номерних знаків;

- кольорові відеокамери використовують в системі, коли необхідно передавати на відеомонітор зображення в кольорі.

2.2 Вибір інструментальних засобів розробки.

Для початку роботи над проектом, було обрано мову програмування Python [6].

Python - це високорівнева, інтерпретована, гнучка та динамічна мова програмування. Вона була створена у 1991 році Гвідо ван Россумом і з тих пір стала однією з найпопулярніших мов програмування у світі. Python зазвичай

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		31

використовують для широкого спектру завдань, включаючи розробку веб-додатків, наукових обчислень, обробку даних, створення ігор та багато іншого. Один з найбільших переваг Python полягає в тому, що вона має простий синтаксис, який дозволяє швидко створювати програми, і в той же час вона має велику кількість бібліотек та інструментів, що допомагає розширити її функціональність і робить її більш гнучкою [7].

Python має багато особливостей, які роблять її привабливою для розробників програмного забезпечення. Одна з таких особливостей - це мультипарадигмальність. Python підтримує різні парадигми програмування, включаючи процедурне програмування, об'єктно-орієнтоване програмування та функціональне програмування.

Python також має велику та активну спільноту розробників та користувачів, що забезпечує наявність великої кількості корисних ресурсів, включаючи документацію, форуми та бібліотеки.

У загальному, Python - це потужна та проста мова програмування, яка може бути використана для різних задач. Її особливості, такі як мультипарадигмальність, велика кількість бібліотек та переносимість, забезпечують швидку та ефективну розробку програмного забезпечення. Також щоб підтвердити всі перераховані фактори, які були наведені вище, представлена таблиця 2.1.

Таблиця 2.1

Особливості мови програмування Python.

Переваги	Недоліки
Простота: Python має простий та лаконічний синтаксис, який дозволяє розробникам швидко та легко написати програму.	Швидкість виконання: Python може бути повільнішим у порівнянні з іншими мовами програмування, такими як C або C++, через те, що вона є інтерпретованою мовою.

<p>Мультипарадигмальність: Python підтримує різні парадигми програмування, що дозволяє розробникам використовувати різні стилі програмування в залежності від задачі.</p>	<p>Обмеження потужності: Python може мати обмеження у потужності при обробці великих обсягів даних або при написанні високопродуктивних програм.</p>
<p>Велика кількість бібліотек та інструментів: Python має велику кількість бібліотек та інструментів, що допомагає розширити його функціональність та робить його більш гнучким.</p>	<p>Відсутність строгих типів даних: Python має динамічну типізацію, що може призвести до помилок під час виконання програми.</p>
<p>Переносимість: програми, написані на Python, можуть працювати на різних операційних системах, що дозволяє розробникам створювати кросплатформові програми.</p>	<p>Залежність від зовнішніх бібліотек: Python може вимагати встановлення зовнішніх бібліотек, щоб розширити її функціональність.</p>

Для роботи с мовою Python було обрано середовище зображене на рисунку 2.2.

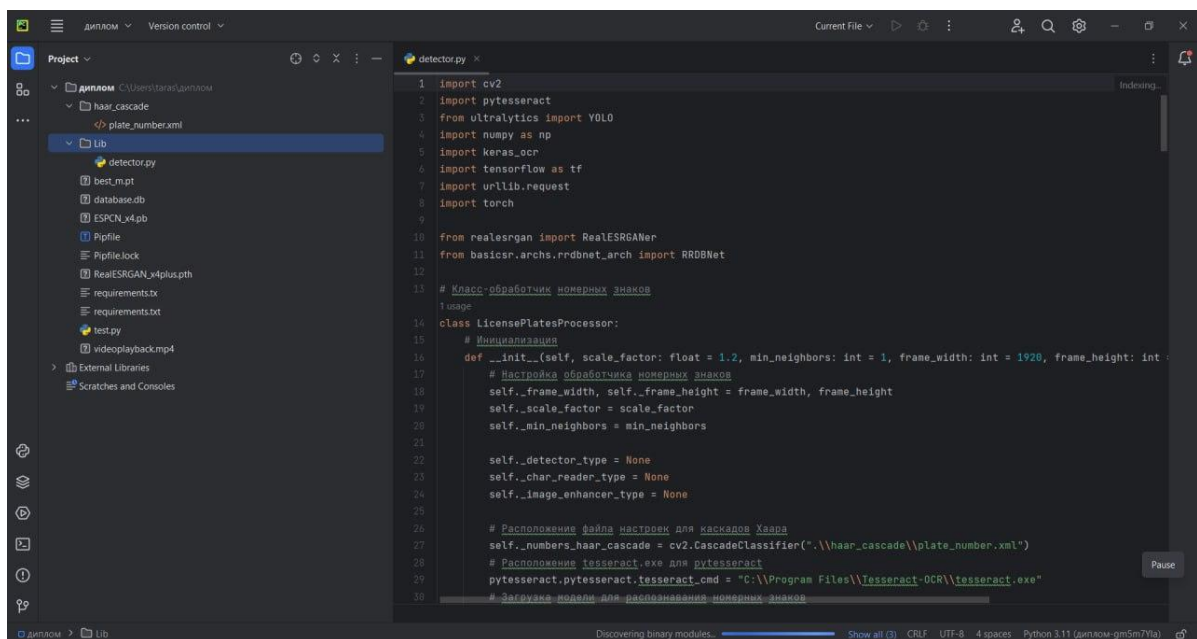


Рисунок 2.2 - Середовище розробки PyCharm.

PyCharm – це інтегроване середовище розробки (IDE) для мови програмування Python, яке розробляється компанією JetBrains. PyCharm має широкий набір функцій, які допомагають розробникам Python зробити їх

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

роботу більш продуктивною та ефективною [8].

Деякі з функцій PyCharm включають:

- Підтримка віртуальних середовищ: PyCharm дозволяє створювати та використовувати віртуальні середовища Python, щоб забезпечити чисте тестування коду та уникнути конфліктів залежностей.

- Автодоповнення та візуалізація коду: PyCharm надає автодоповнення коду, що забезпечує швидкий доступ до методів, функцій та атрибутів класів. Також воно має можливість візуалізації коду, що дозволяє зрозуміти, як код працює та як він може бути вдосконалений [9].

- Інструменти для відлагодження коду: PyCharm має можливість відлагодження коду, що дозволяє шукати та усувати помилки в коді.

- Керування версіями: PyCharm інтегрується з системами керування версіями, такими як Git, що дозволяє розробникам працювати зі своїми проектами та зберігати їх історію версій.

- Підтримка різних фреймворків: PyCharm підтримує різні фреймворки, такі як Django, Flask, Pyramid та інші.

Провівши детальний аналіз середовища розробки PyCharm, було зроблено висновки про переваги і недоліки використання даної IDE які представлені в таблиці 2.2.

Таблиця 2.2

Особливості IDE PyCharm.

Переваги	Недоліки
Розширена функціональність: PyCharm має широкий набір функцій, які дозволяють розробникам ефективніше працювати над своїми проектами. Це включає розширену підтримку Python, інтеграцію зі сторонніми інструментами та багато іншого.	Висока вартість: PyCharm є платним продуктом, що може стати проблемою для деяких користувачів, особливо для початківців.
Підвищує продуктивність: PyCharm дозволяє розробникам працювати швидше та ефективніше, забезпечуючи різноманітні функції, які допомагають у розробці, налагодженні та тестуванні коду.	Важкий для навчання: PyCharm має багато функцій, які можуть бути складними для новачків. Розробникам потрібен час, щоб навчитися використовувати всі функції PyCharm.

Продовження таблиці 2.2

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34

Кращий редагувальник коду: PyCharm має потужний та зручний редагувальник коду, що дозволяє розробникам створювати та редагувати код швидше та ефективніше.	Вимогливий до ресурсів ПК: PyCharm вимагає певного рівня апаратних можливостей для оптимальної роботи. Це може стати проблемою для користувачів зі слабкими ПК або обмеженими ресурсами.
--	--

2.3 Вибір алгоритму розпізнавання символів номерних знаків.

Під час вибору алгоритму розпізнавання символів номерних знаків автомобілів, необхідно враховувати такі фактори:

1. Швидкість: алгоритм повинен працювати швидко, щоб забезпечити розпізнавання символів в реальному часі.
2. Точність: алгоритм повинен бути достатньо точним, щоб уникнути помилкового розпізнавання номерних знаків.
3. Надійність: алгоритм повинен бути надійним та стійким до змін у зовнішніх умовах, таких як зміна освітлення, погодних умов тощо.
4. Простота: алгоритм повинен бути простим у реалізації та підтримці.

Для розпізнавання символів номерних знаків автомобілів можна використовувати різні алгоритми, такі як:

5. Метод шаблонів (Template Matching): цей метод використовує шаблони символів номерних знаків та порівнює їх зі зображеннями для пошуку відповідностей.
6. Метод зіставлення зразків (Pattern Recognition): цей метод використовує навчання з вчителем для розпізнавання символів.
7. Методи машинного навчання (Machine Learning): ці методи використовуються для розпізнавання символів номерних знаків, використовуючи навчальні дані для тренування моделі.
8. Нейронні мережі (Neural Networks): цей метод використовує нейронні мережі для розпізнавання символів.

Кожен з цих методів має свої переваги та недоліки, тому вибір алгоритму залежить від конкретних вимог проекту та умов використання.

Наприклад, метод шаблонів є досить простим та швидким, проте він може бути не ефективним у випадках, коли зображення символу зі значною деградацією або коли номерний знак має відмінності у формі чи розмірі.

Метод зіставлення зразків є більш точним та може показувати кращі результати у порівнянні з методом шаблонів, проте він вимагає навчання з вчителем та розробки складних алгоритмів [10].

Методи машинного навчання та нейронні мережі можуть дати ще кращі результати, проте вони вимагають значної кількості навчальних даних та ресурсів для навчання та застосування моделі.

Залежно від конкретних вимог проекту та ресурсів, можна вибрати той метод, який найкраще підходить для розробки системи розпізнавання номерних знаків автомобілів на базі бібліотеки OpenCV. Також можна поєднувати різні методи для досягнення кращих результатів та оптимальної швидкості та точності.

2.4 Вибір бібліотеки комп'ютерного зору OpenCV.

OpenCV (Open Source Computer Vision Library) - це відкрита бібліотека комп'ютерного зору, яка дозволяє розробникам реалізувати багато функцій комп'ютерного зору на різних платформах та мовах програмування, включаючи C++, Python, Java та інші. OpenCV забезпечує можливість машинного навчання та глибинного навчання, що дозволяє розробникам використовувати цю бібліотеку для розробки інтелектуальних систем зі здатністю до навчання та аналізу даних. Крім того, OpenCV має активну спільноту розробників та користувачів, яка надає підтримку, оновлення та розробку бібліотеки, що дозволяє покращувати та розширювати її можливості з часом [11].

Найпоширенішими функціями OpenCV є:

- робота з зображеннями та відео (зчитування, запис, обрізка, обробка, збереження);
- обробка та аналіз зображень (фільтрація, розмиття, розпізнавання облич, виявлення країв, розпізнавання форм);

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

- відстеження руху об'єктів;
- розпізнавання об'єктів на зображеннях та відео за допомогою навчальних моделей машинного навчання (наприклад, нейронних мереж);
- створення 3D-моделей зображень та відео.

OpenCV є також популярною серед науковців та студентів, які займаються дослідженням та розробкою у галузі комп'ютерного зору, оскільки бібліотека дозволяє їм ефективно виконувати завдання, пов'язані з обробкою зображень та відео, без необхідності написання власного програмного коду. Крім того, OpenCV має високу швидкодію, що дозволяє оброблювати великі обсяги даних у режимі реального часу. Роботу бібліотеки OpenCV можна побачити на рисунках 2.3, 2.4.

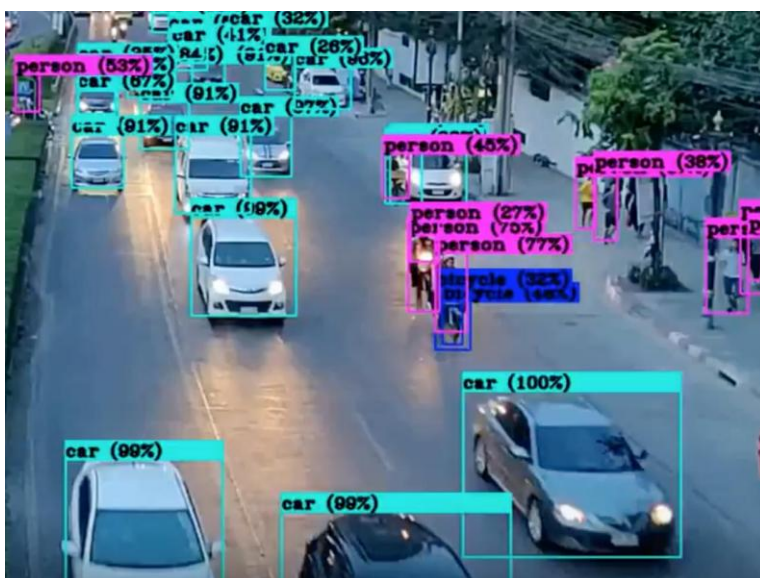


Рисунок 2.3 - Приклади роботи бібліотеки OpenCV.

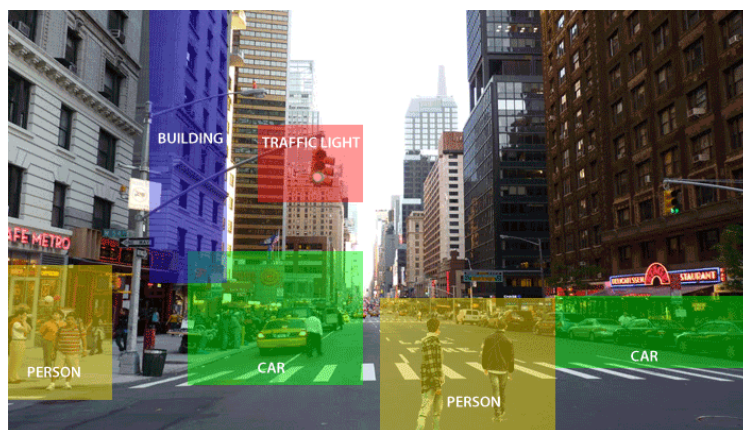


Рисунок 2.4 - Приклади роботи бібліотеки OpenCV.

Також можна виділи наступні переваги та недоліки OpenCV, які подано в таблиці 2.3.

Таблиця 2.3

Особливості бібліотеки OpenCV.

Переваги	Недоліки
Відкритість і безкоштовність: OpenCV є вільним програмним забезпеченням, що означає, що ви можете використовувати її безкоштовно і змінювати її відповідно до ваших потреб.	Складність використання: OpenCV має велику кількість функцій та можливостей, що робить її складною для вивчення та використання.
Багата функціональність: OpenCV має багато функцій, що дозволяє вирішувати різноманітні завдання комп'ютерного зору. Вона підтримує роботу зі зображеннями та відео, аналіз зображень, відстеження руху об'єктів та інші завдання.	Висока вимогливість до апаратної конфігурації: OpenCV може вимагати великої кількості ресурсів комп'ютера, зокрема відеокarti зі значними обчислювальними можливостями.
Підтримка різних мов програмування: OpenCV підтримує різні мови програмування, зокрема C++, Python, Java, MATLAB та інші, що дозволяє використовувати її в різних проектах	Потреба в знанні математичних методів: для ефективного використання OpenCV потрібно мати знання з області комп'ютерного зору та математичних методів, які використовуються в алгоритмах комп'ютерного зору.
Документація: OpenCV має докладну документацію, яка допомагає користувачам вивчити її функціонал та використовувати її для вирішення практичних завдань.	Не завжди ефективна робота зі слабо контрастними зображеннями: OpenCV може мати проблеми з роботою зі слабо контрастними зображеннями або зображеннями низької якості, що може вплинути на якість розпізнавання.
Підтримка активного співтовариства: OpenCV має велике активне співтовариство, що дозволяє швидко отримати відповіді на питання та розв'язати проблеми з використанням бібліотеки.	Неможливість розпізнати рукописний текст: OpenCV не підтримує розпізнавання рукописного тексту, що є одним з обмежень її використання.

2.5. Вибір засобів розроблення бази даних.

Для створення бази даних було обрано SQLite - це вбудовувана реляційна база даних, яка не потребує окремого сервера для своєї роботи і зберігає дані в одному файлі на диску. SQLite, зображено на рисунку 2.5, зазвичай використовується в мобільних додатках та веб-сайтах, де не потрібна велика

потужність бази даних та високі навантаження. SQLite підтримує стандартні операції з базами даних, такі як SELECT, INSERT, UPDATE та DELETE, а також має вбудовану підтримку транзакцій та реплікації даних. SQLite доступний для використання в багатьох мовах програмування, включаючи C, C++, Java, Python та інші [12].



Рисунок 2.5 - Середовище бази даних SQLite.

SQLite є досить легким та швидким рішенням для зберігання та обробки невеликих обсягів даних, таких як дані користувачів, історія перегляду, налаштування додатків та інші. Він може працювати на різних операційних системах, включаючи Windows, macOS та Linux. SQLite є безкоштовним та відкритим програмним забезпеченням з ліцензією Public Domain, що дозволяє його використання в комерційних проектах без обмежень [13].

У SQLite є декілька особливостей, які роблять його корисним для використання в окремих випадках, наприклад:

- Висока продуктивність при роботі з невеликими обсягами даних та малими навантаженнями
- Вбудована підтримка транзакцій та гарантована цілісність даних
- Низький рівень залежності від зовнішніх компонентів, що дозволяє легко вбудовувати SQLite в різноманітні додатки
- Підтримка стандартних SQL-операцій, які забезпечують сумісність з більшістю інших систем керування базами даних

У разі потреби можна легко перенести дані з SQLite на інші бази даних, що мають більшу потужність та масштабованість, наприклад, на MySQL або PostgreSQL.

Провівши детальний аналіз середовища бази даних SQLite, було зроблено висновки про переваги і недоліки використання даної бази даних які представлені в таблиці 2.4.

Таблиця 2.4

Переваги і недоліки SQLite.

Переваги	Недоліки
Легкість встановлення та налаштування, що дозволяє швидко розпочати роботу з базою даних.	Обмежена масштабованість та продуктивність, що обмежує його використання в великих проектах та з великими обсягами даних .
Швидка робота з невеликими обсягами даних та малими навантаженнями.	Не підтримує розподілені бази даних та не має вбудованої механізму реплікації
Вбудована підтримка транзакцій та гарантована цілісність даних .	Обмежена підтримка запитів на вибірку даних зі складними умовами та групуваннями .

Продовження таблиці 2.4

Низький рівень залежності від зовнішніх компонентів та легка вбудова в різноманітні додатки .	Відсутність підтримки деяких типів даних, які можуть бути корисними в певних випадках .
Підтримка стандартних SQL-операцій, що дозволяє легко перенести базу даних на інші системи керування базами даних.	

2.6 Висновки по розділу

У другому розділі розроблено структурну схему програмної системи, яка забезпечує модульність і масштабованість. Обґрунтовано вибір бібліотеки OpenCV як основного інструменту для обробки зображень завдяки її широким можливостям у задачах комп'ютерного зору. Визначено оптимальний алгоритм

розпізнавання символів номерних знаків, враховуючи баланс між швидкістю та точністю. Вибір інструментів розробки, зокрема засобів для створення бази даних, забезпечує ефективне зберігання та обробку даних. Проведений аналіз підтвердив доцільність обраних технологій для реалізації системи, яка відповідає функціональним і нефункціональним вимогам.

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		41

РОЗДІЛ 3. ПРОЕКТУВАННЯ І ТЕСТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ

3.1. Розробка алгоритму функціонування системи ідентифікації номерних знаків.

Для опису загального принципу роботи програми ідентифікації транспортних засобів, було розроблено структурну схему (рис. 3.1).

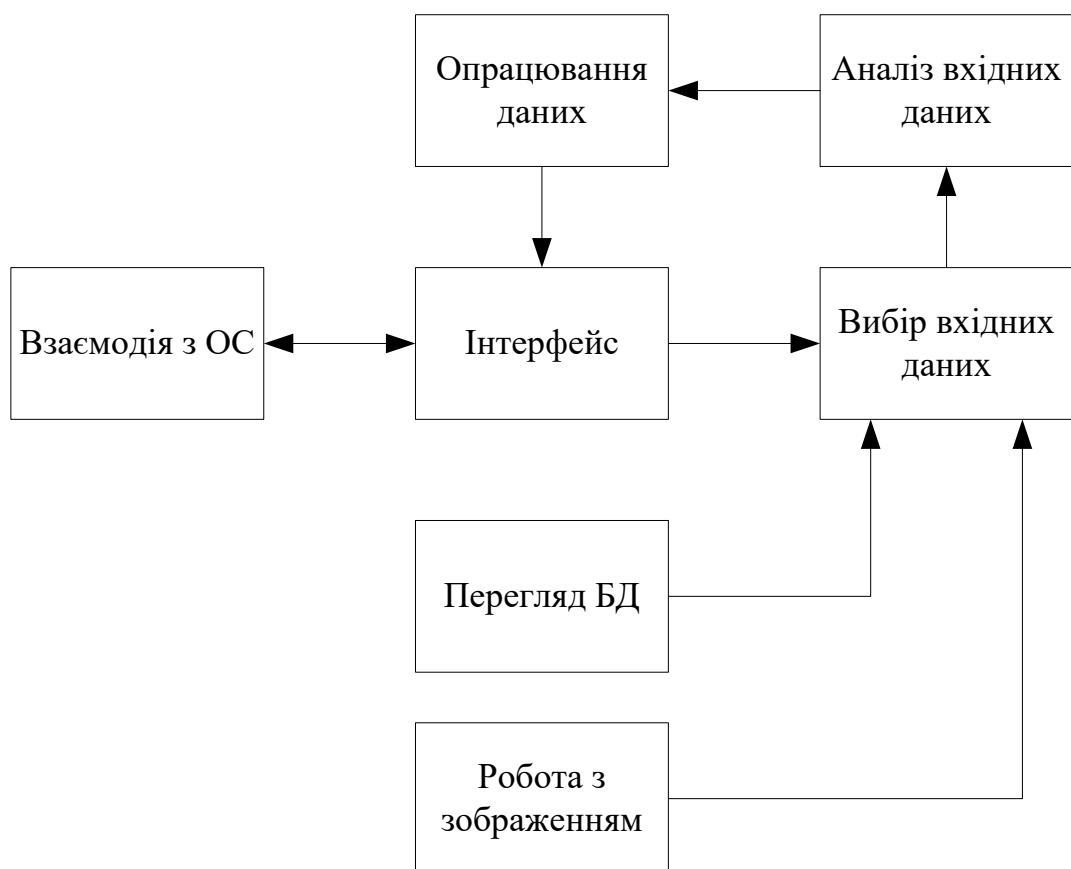


Рисунок 3.1 – Структурна схема ПЗ ідентифікації транспортних засобів.

Принцип функціонування структурної схеми, зображена на рисунку 3.1 полягає в тому, що користувач після запуску програми обирає задачу, яку йому необхідно виконати: чи це буде робота з базою даних (пошук і перегляд попередніх результатів), чи це буде ідентифікація чергового транспортного засобу. Після цього проводиться вибір вхідних даних (фільтри для пошуку по БД, чи зображення

відповідно). В подальшому вхідні дані перевіряються на коректність і передаються на обробку. Після цього отримані результати виводяться на інтерфейс, де користувач зможе їх бачити, аналізувати та проводити подальші дії [14].

Принцип функціонування структурної схеми, зображеної на рисунку 3.1, полягає в тому, що користувач після запуску програми обирає задачу, яку необхідно виконати: це може бути робота з базою даних (пошук і перегляд попередніх результатів), або ідентифікація нового транспортного засобу. Після цього здійснюється вибір вхідних даних — це можуть бути або параметри для пошуку в БД, або файл зображення автомобіля для розпізнавання номерного знака.

На етапі обробки зображення виконується кілька ключових операцій:

- **Попередня обробка зображення:** конвертація у відтінки сірого, фільтрація шумів, покращення контрасту та нормалізація розміру.
- **Виділення області номерного знака:** використання методів виявлення контурів, фільтрація за співвідношенням сторін і положенням прямокутних об'єктів.
- **Сегментація символів:** розділення зображення номерного знака на окремі символи.
- **Розпізнавання символів:** застосування Tesseract OCR або нейронної мережі для ідентифікації літер і цифр.
- **Аналіз та виведення результатів:** виведення розпізнаного номерного знака у графічному інтерфейсі або збереження у базу даних.

У випадку, якщо користувач працює з базою даних, система здійснює запит згідно з вказаними фільтрами (наприклад, дата, номер, регіон тощо), після чого результати виводяться на екран у вигляді списку або таблиці з можливістю перегляду повної інформації.

Уся логіка програми реалізована з урахуванням можливості масштабування та розширення функціоналу. Алгоритм функціонування розділений на окремі модулі, що спрощує його підтримку та тестування.

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

Застосування бібліотеки **OpenCV** для обробки зображень дозволило досягти високої швидкості виконання і сумісності з сучасними форматами зображень, а використання **Tesseract OCR** дало можливість ефективно розпізнавати символи номерних знаків навіть у складних умовах (нахил, тінь, пошкодження тощо).

Таким чином, побудований алгоритм дозволяє ефективно реалізувати повний цикл роботи системи: від завантаження вхідних даних до виведення результату в зручному для користувача вигляді.

3.2. Розробка алгоритму функціонування програми ідентифікації транспортних засобів.

Після запуску програми, користувачеві надається вибір подальших його дій. Він має можливість переглядати наявну базу даних, або обрати нове зображення для ідентифікації наявного на ньому транспортного засобу.

Якщо користувач обирає перегляд бази даних, йому пропонується обрати один з варіантів: пошук за реєстраційним номером та пошук за датою розпізнавання. При пошуку за номером одразу формується SQL-запит з метою отримання наявних у базі номерних знаків. Після вибору потрібного користувачеві номеру формується список усіх зображень, де фігурує вказаний номер. Обравши зображення зі списку, воно виводиться в окремому вікні. При пошуку за датою користувачеві пропонується обрати довільну дату, після чого формується SQL-запит і виводяться дані ідентифікації проведеної в цей день, а саме: точний час ідентифікації та реєстраційний номер.

За умови, що користувач вибере варіант ідентифікації чергового транспортного засобу – зображення завантажується відображається у вікні. У фоновому режимі проводиться процедура бінаризації, тобто переведення зображення у чорно-біле. Це проводиться в зв'язку з тим, що більшість алгоритмів комп'ютерного зору адаптовані для роботи з чорно-білими зображеннями. Після цього проводиться пошук прямокутних областей [15].

Процес пошуку прямокутних областей включає в себе адаптування

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

(полягає у масштабуванні) зображення за нижнім порогом, знаходження областей у кожній кольоровій площині зображення, застосування оператора Кенні на нульовому пороговому рівні, що дозволить захопити області з градієнтом затінення. Згодом проводиться перевірка обраної області на предмет співпадіння пропорції сторін області з пропорцією сторін номерного знаку.

Після того, як буде знайдено підходящу за критеріями область, вона передається для аналізу в бібліотеку Tesseract OCR. На цьому етапі виникає проблема з наявністю шумів на картинці, так як дана бібліотека чутлива до шумів. Для того, щоб позбавитись від шумів необхідно бінаризувати зображення і застосувати операції ерозії та дилатації, які відносяться до морфологічних операцій. Ерозія робить об'єкти більш тонкими, а дилатація – навпаки робить їх більш товстими. Після цього шум забирається і залишаються лише вагомі деталі. Згодом проводиться цикл по контурах і вибір букв з них, з подальшим їх розпізнаванням і формуванням рядка символів [16].

Результат розпізнавання виводиться у відведене для нього місце у головному вікні.

Після запуску програми користувачеві надається можливість обрати подальші дії. Інтерфейс реалізовано таким чином, щоб забезпечити зручний вибір між переглядом бази даних та проведенням нової ідентифікації транспортного засобу за допомогою завантаженого зображення.

1. Робота з базою даних

Якщо користувач обирає перегляд бази даних, система пропонує два варіанти пошуку:

- Пошук за реєстраційним номером.
- Пошук за датою розпізнавання.

У першому випадку система формує SQL-запит до бази з метою отримання усіх записів, пов'язаних із введеним номером. Після цього формується список зображень, на яких фігурує вказаний номерний знак. Користувач може обрати конкретне зображення зі списку — воно відкривається у окремому вікні для перегляду.

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

У другому випадку система пропонує обрати дату, після чого також формується SQL-запит, за результатами якого виводяться записи, що містять:

- точний час розпізнавання;
- номерний знак;
- відповідне зображення (за наявності).

Це забезпечує гнучкий підхід до перегляду архівних даних і спрощує аналіз.

2. Ідентифікація нового транспортного засобу

У випадку, якщо користувач обирає ідентифікацію нового транспортного засобу:

1. **Завантажується зображення автомобіля** та відображається у головному вікні.

2. У фоновому режимі виконується **бінаризація зображення** (перетворення у чорно-біле), оскільки більшість методів комп'ютерного зору працюють ефективніше саме з такими зображеннями.

3. **Пошук прямокутних областей**, які можуть містити номерні знаки:

- масштабування зображення;
- обробка кожної кольорової площини окремо;
- застосування оператора Кенні для виявлення контурів;
- перевірка знайдених контурів на відповідність типовим співвідношенням сторін номерних знаків.

Після того, як знайдено найбільш ймовірну область з номером, вона передається для розпізнавання у бібліотеку **Tesseract OCR**.

Оскільки Tesseract чутливий до наявності шумів, попередньо виконується:

- **морфологічна обробка зображення**, зокрема:
 - **ерозія** (стиснення елементів);
 - **дилатація** (розширення елементів);
- ці операції дозволяють прибрати дрібні артефакти та залишити лише значущі символи.

Далі виконується:

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46

- виявлення контурів символів;
- сегментація та розпізнавання кожного символу;
- формування фінального рядка номерного знаку.

Результат розпізнавання (рядок із символами номерного знака) виводиться у відведене місце в головному вікні інтерфейсу. За потреби він може бути збережений у базу даних разом із зображенням і міткою часу.

3.3 Розробка інтерфейсу користувача.

Інтерфейс користувача повинен бути зручним та інтуїтивно зрозумілим для користувача. При запуску застосунку користувачу показується головне вікно(рис.3.2).

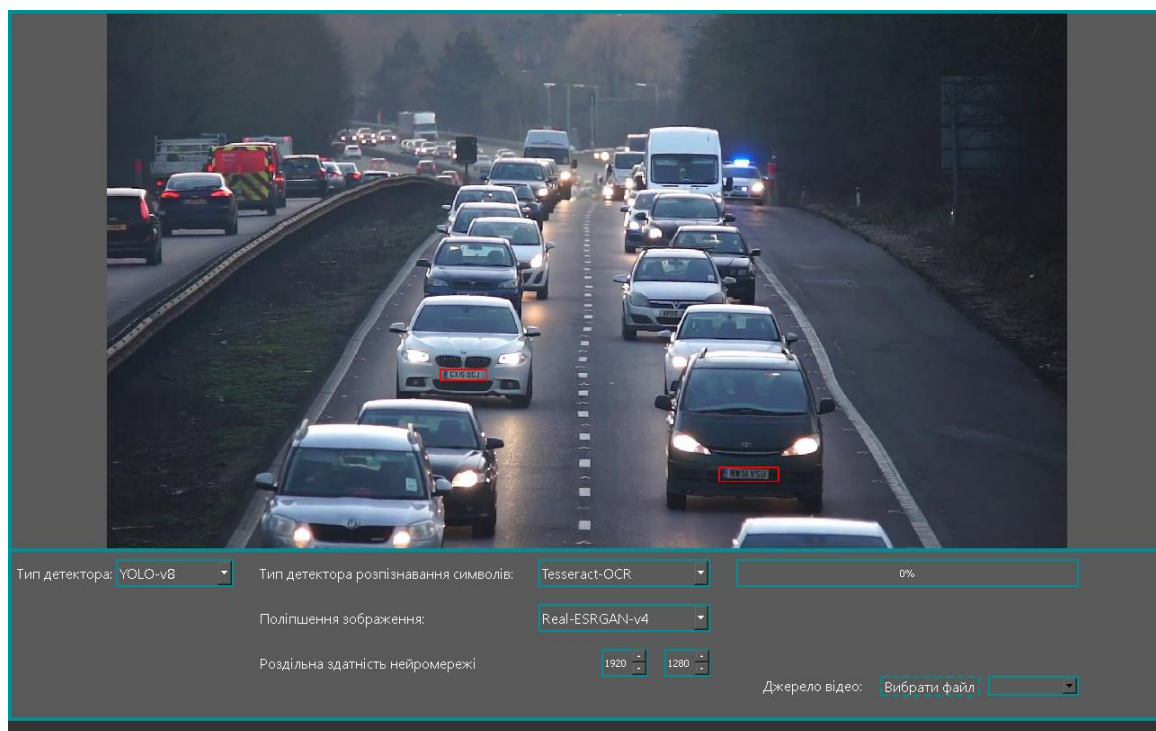


Рисунок 3.2 – Головне вікно застосунку.

У цьому вікні користувач має можливість обрати параметри за якими буде проводиться процес розпізнавання номерних знаків. Йому доступні такі параметри:

– Тип детектора.

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		47

- Тип детектора розпізнавання символів.
- Поліпшення зображення.
- Роздільна здатність нейромережі.

Також в цьому вікні користувач може прослідкувати за прогресом виконанням розпізнавання, який позначено відсотками, чи обрати файл з відео для розпізнавання.

На рисунках 3.3 та 3.4 представлено список параметрів, що доступні користувачу.

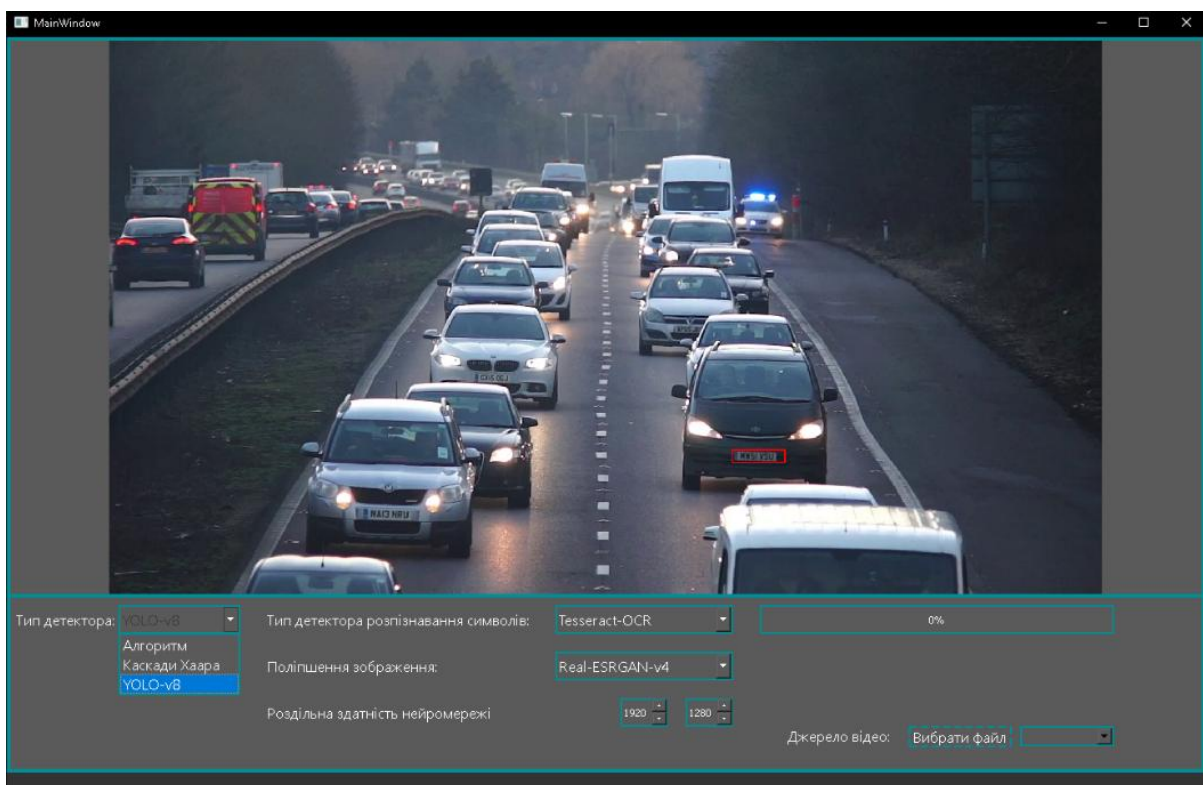


Рисунок 3.3 – Параметри детектора.

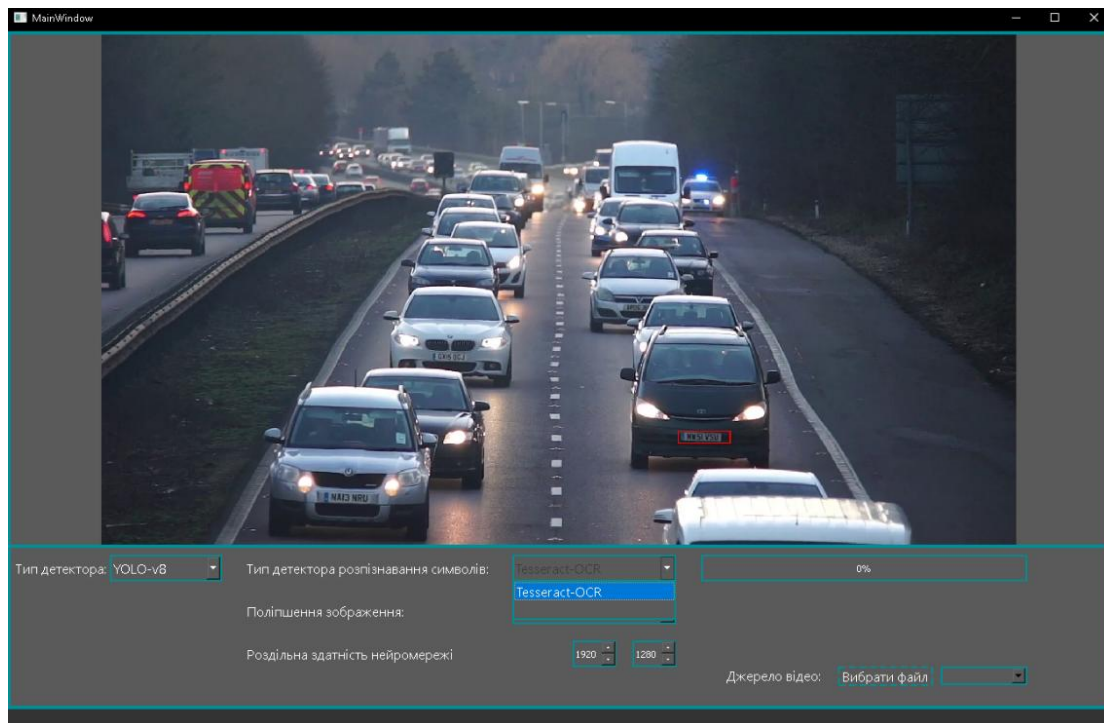


Рисунок 3.4 – Параметри детектора розпізнавання символів.

В залежності від обраних параметрів застосунок запускає певний алгоритм дій, детальний опис яких наведено у відповідному підпункті.

Для початку роботи з програмою необхідно обрати файл з відео. Для цього користувач повинен

3.4 Реалізація алгоритму роботи програмного забезпечення.

Для роботи системи існують 3 алгоритми, що були описані вище: на основі нейромережі, на основі каскадів Хаара та функція.

Каскади Хаара - це метод комп'ютерного зору, що використовується для виявлення облич, об'єктів або певних характеристик в зображеннях. Вперше запропоновані В'ячеславом Хааром у 1909 році, вони отримали свою популярність завдяки праці Віоли та Джонса у 2001 році, де їх було застосовано для швидкого виявлення облич на зображеннях [18].

Каскади Хаара використовуються в алгоритмі виявлення облича з методом Віоли-Джонса, який включає кілька етапів обробки.

Побудова інтегрального зображення:

					БР.ІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

– Спочатку зображення перетворюється у інтегральне зображення, яке представляє суму пікселів усередині прямокутника, що розташований вгорі та зліва від поточного пікселя. Це дозволяє ефективно обчислювати суму пікселів у довільному прямокутнику за константний час.

Використання каскаду класифікаторів:

– Каскади Хаара включають набір класифікаторів, які по черзі оцінюють регіони зображення. Кожен класифікатор є деревом прийняття рішень, яке базується на попередньому класифікаторі і використовує набір признаков Хаара.

Використання признаков Хаара:

– Признаки Хаара - це шаблони різного розміру та форми, які застосовуються до регіонів зображення для виявлення певних характеристик. Кожен признак Хаара представляє собою суму яскравості пікселів на одній області зображення, відняту від суми яскравості пікселів на іншій області. Ці признаки дозволяють виявляти зміни яскравості на зображенні.

Адаптивне порогове значення:

– Кожен класифікатор має адаптивне порогове значення, яке використовується для прийняття рішення про те, чи належить регіон об'єкту, який виявляється. Порогове значення обчислюється шляхом порівняння суми признаков Хаара з попередньо встановленим порогом.

Каскадний підхід:

– Каскад Хаара складається з кількох етапів, в кожному з яких використовуються класифікатори зі зростаючою складністю. Перші класифікатори швидко відкидають неінтересні регіони, а лише потенційно цікаві регіони проходять наступні етапи. Це дозволяє прискорити процес виявлення обличчя [19].

Оцінювання та фільтрація:

– На останньому етапі, якщо регіон успішно пройшов усі класифікатори каскаду, він вважається знайденим об'єктом. Додатково можуть використовуватися методи фільтрації, наприклад, коригування прямокутної

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		50

рамки об'єкта для покращення точності знаходження.

Каскади Хаара є ефективним методом виявлення облич та об'єктів у реальному часі, і їх успішно використовують у системах відеоспостереження, автоматичному розпізнаванні облич та інших застосуваннях комп'ютерного зору.

Відповідні алгоритми були реалізовані програмним кодом та були додані в інструкцію користувача.

Алгоритм опис і його робота:

1. Розмиття зображення для видалення зайвого шуму;
2. знаходження країв на зображенні ;
3. знаходження замкнених контурів у списку країв ;
4. сортування контурів за площею ;
5. знаходження зі списку замкнених контурів чогось схожого на номерний знак (апроксимація замкненого контуру);
6. якщо контур має чотири точки (тобто він прямокутник) ;
7. визначення співвідношення сторін прямокутника ;
8. якщо співвідношення сторін більше ніж 2 та менше ніж 4, то відмітити як номерний знак [20].

Робота алгоритму покроково представлена на рисунках 3.5 – 3.9.



Рисунок 3.5 – Картинка з покращеною якістю.

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

Початком роботи алгоритму розпізнавання знаків є покращення зображення, тобто видалення зайвих шумів та недоліків. Результат покращення зображено на рисунку вище.



Рисунок 3.6 – Інвертація зображення в сірий колір.

Після покращення якості зображення його колір було змінено на сірий. Ця маніпуляція дозволяє програмі краще проаналізувати номерний знак машини. Результат відповідної маніпуляції зображено вище.



Рисунок 3.7 – Заблюрене зображення.

Ця маніпуляція призначене для того, щоб краще виділявся номерний знак автомобіля. Що допоможе програмі ефективніше зчитувати номерний знак і коректніше отримувати результат.



Рисунок 3.8 – Затемнене зображення.

Ця маніпуляція призначене для того, щоб алгоритм виділив відповідний номерний знак, та виділив його, після чого цей номерний знак буде повернуто до нормального вигляду.



Рисунок 3.9 – Визначений номерний знак.

Результатом роботи алгоритму є зображення номерного знаку, що розпізнається з картинки автомобіля. Алгоритм роботи працює коректно, номерний знак співпадає з номерним знаком на картинці.

3.5. Перевірка функціоналу програмної системи.

					БР.ІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		53

Для початку роботи з програмою потрібно обрати файл з відео, для якого буде проводитися зчитування знаків. Для цього потрібно натиснути кнопку «Вибрати файл» та обрати його (рис.3.10).

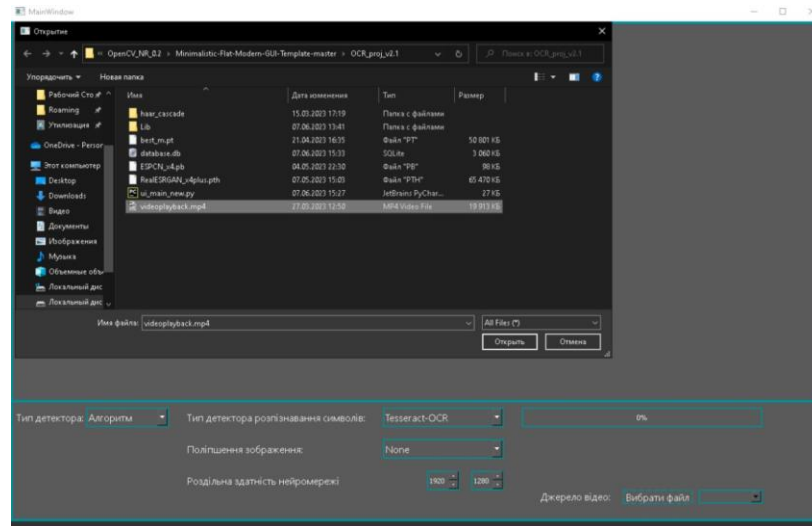


Рисунок 3.10 – Вікно з вибором файлу.

Після обрання файлу необхідно в параметрах програми вказати тип детектора та тип детектора розпізнавання символів. Після цього програма розпочне свою роботу, а прогрес виконання буде відслідковуватись користувачем у вигляді відсотків (рис.3.11).

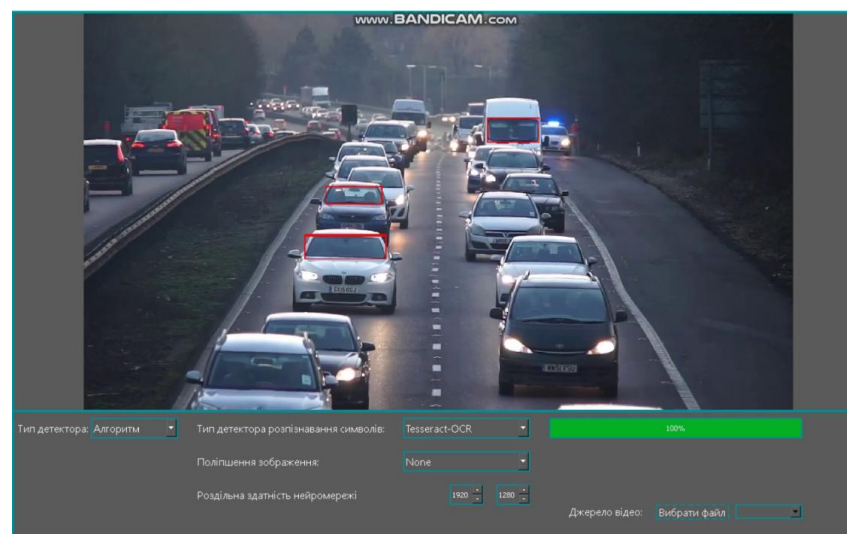


Рисунок 3.11 – Результат виконання програми.

Якщо прослідкувати за картинкою, то можна побачити, що номерні знаки, які записуються в програму визначені коректно, тому програма адекватно виконує свою роботу, а алгоритм розпізнання номерних знаків свій функціонал.

Перед зчитуванням даних з відео, картинку потрібно підготувати, відповідний процес представлено на рисунках 3.12 – 3.15.



Рисунок 3.12 – Початкове зображення.



Рисунок 3.13 – Зображення в сірому кольорі.



Рисунок 3.14 – Розмите зображення.



Рисунок 3.15 – Чорно-біле зображення.

Процес підготовки зображення включає в себе конвертацію зображення в сірий колір та його подальше заблюрення для виділення основних елементів картини. Після чого її перетворюють в чорно-білий формат та передають Tesseract. Там на кінцевому зображенні розпізнаються символи номерного знаку та відповідна інформація виводиться на екран.

Після розпізнавання номерного знаку він заноситься до бази даних (рис.3.16).

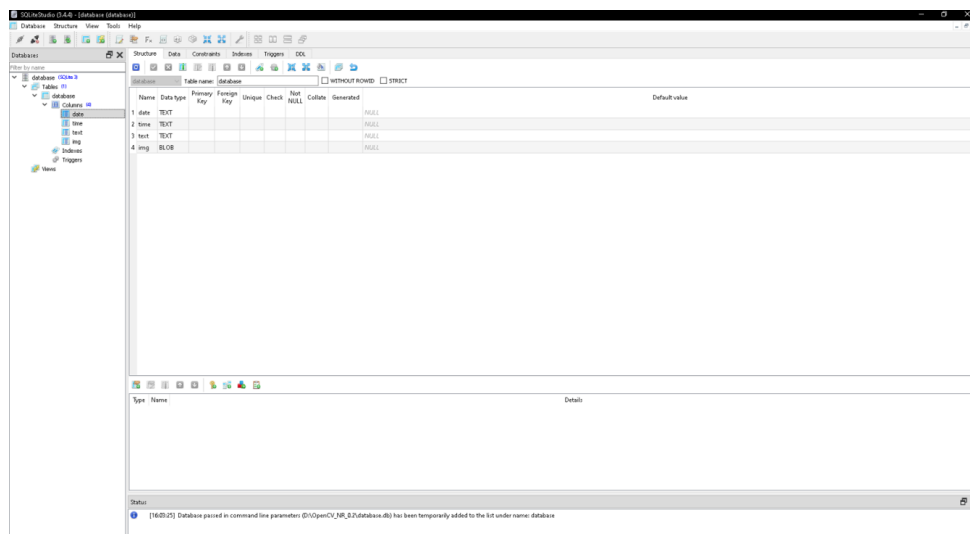


Рисунок 3.16 – База даних номерних знаків.

В цій базі даних буде міститись інформація про номерні знаки, а саме: картинка номерного знаку, текст, дата та час розпізнавання.

У результаті тестування програми було зроблено висновок про те, що вона працює коректно. Номерні знаки розпізнаються програмою шляхом

підготовки зображення з відео та його подальшою обробкою з метою передачі в Tesseract. Там відповідне зображення обробляється та виводить користувачу на екран текст, який є на ньому. Вся інформація після цього переноситься в базу даних. В ній, завдяки наданій інформації, користувач зможе проаналізувати коректність роботи системи.

3.6 Висновки по розділу

У третьому розділі спроектовано алгоритми функціонування системи розпізнавання номерних знаків та ідентифікації транспортних засобів, які забезпечують швидке виявлення та обробку зображень. Розроблено зручний користувацький інтерфейс, який спрощує взаємодію з системою. Реалізація програмного забезпечення на базі OpenCV дозволила досягти високої точності розпізнавання. Проведене модульне тестування підтвердило стабільність роботи системи в різних умовах, включаючи зміни освітлення та кути зйомки, а також її відповідність поставленим вимогам щодо швидкості та достовірності обробки.

					БР.ІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		57

РОЗДІЛ 4. РОЗРОБКА СИСТЕМИ РОЗПІЗНАВАННЯ

4.1 Виклад основного матеріалу дослідження

Розглянемо розробку алгоритму розпізнавання номерних знаків. Віртуальне середовище є важливою складовою для розробки та тестування сучасних програмних комплексів, зокрема для розпізнавання номерних знаків та задач комп'ютерного зору. В організації такого середовища особливу роль відіграє підбір інструментів і бібліотек, а також налаштування програмного забезпечення для досягнення максимального результату при реалізації алгоритмів. Ось деякі ключові аспекти налаштування віртуального середовища для завдань розпізнавання номерних знаків.

Вибір операційної системи та програмного забезпечення. Перше, що потрібно зробити при організації віртуального середовища розпізнавання номерних знаків, це визначити операційну систему та набір інструментів, які будуть використовуватися. Найбільш популярними операційними системами для таких завдань є Linux і Windows, оскільки вони підтримують більшість необхідних бібліотек і інструментів. У більшості випадків віртуальне середовище розробки налаштовується за допомогою програмних засобів, таких як Python і OpenCV.

Python — це універсальний інструмент для розробки та тестування систем машинного навчання завдяки численним бібліотекам для роботи з комп'ютерним зором і обробкою зображень.

OpenCV — це бібліотека для роботи з зображеннями та відео, яка включає широкий набір функцій для обробки зображень, таких як фільтрація, визначення контурів, виділення об'єктів, розпізнавання тексту тощо. Tesseract OCR це оптичне розпізнавання тексту, яке дозволяє витягувати текст із зображень (наприклад, номерних знаків автомобілів).

NumPy і Matplotlib це бібліотеки для наукових обчислень і візуалізації даних, які широко використовуються в задачах обробки зображень.

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		58

Імутилс — це службова бібліотека, призначена для полегшення роботи з OpenCV. Він містить функції для швидкого виконання типових операцій, таких як зміна розміру, обертання, автоматична зміна орієнтації зображень або видалення об'єктів у певній області кадру. Ця бібліотека робить код більш компактним і легким для читання. EasyOCR це сучасна бібліотека для розпізнавання тексту на основі глибокого навчання. EasyOCR має вбудовані моделі нейронних мереж, які дозволяють розпізнавати текст навіть на складних зображеннях із низькою якістю, складним фоном або тексті, написаному різними мовами. Завдяки своїй простоті та потужності ця бібліотека є чудовим вибором для систем, які вимагають високої точності розпізнавання.

Для розробки програми ми використовуємо інтегроване середовище розробки (IDE) PyCharm. Це середовище підтримує Python, надає інструменти автозавершення, вбудовані налагоджувачі та полегшує роботу з бібліотеками для машинного навчання та обробки зображень. Щоб встановити PyCharm, ми завантажуюмо його з офіційного веб-сайту за посиланням для завантаження PyCharm. Після встановлення відкрийте PyCharm і створіть новий проект, вибравши потрібну версію Python.

Використані бібліотеки для роботи з зображеннями та текстом, такі як OpenCV, imutils, pytesseract та інші, можна встановити через термінал PyCharm. Для цього потрібно виконати наступні команди. Для встановлення бібліотеки OpenCV виконуємо команду:

```
pip встановити opencv-python.
```

Для додаткових утиліт, які полегшують роботу з OpenCV (imutils), ми запускаємо команду `pip встановити imutils`.

Щоб налаштувати оптичне розпізнавання тексту (Tesseract OCR) за допомогою бібліотеки pytesseract, ми використовуємо цю команду

```
pip встановити pytesseract.
```

OpenCV — це основна бібліотека для обробки зображень, яка включає функції фільтрації, визначення контурів, виділення об'єктів тощо;

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		59

imutils – допоміжна бібліотека для зручного масштабування та повороту зображень;

пїтессеракт – інтерфейс до Tesseract OCR для розпізнавання тексту на зображеннях;

numpy — бібліотека для роботи з масивами, часто використовується для обробки пікселів зображення;

winsound - бібліотека для відтворення звуків у середовищі Windows, яка використовується для інформування про знайдений номерний знак.

Щоб використовувати Tesseract OCR, ви також повинні завантажити та встановити його з офіційного репозиторію на GitHub, а потім вказати шлях до виконуваного файлу в коді:

```
pytesseract.pytesseract.tesseract_cmd = r"C:\Program Файли\Tesseract-OCR\tesseract.exe".
```

Для обробки та розпізнавання номерних знаків необхідно підготувати набір зображень, що містять номерні знаки автомобіля. Зображення можуть бути з різних ракурсів, з різним освітленням і якістю, що вимагає відповідних налаштувань для оптимальної роботи алгоритмів виявлення контурів і розпізнавання тексту. Для спрощення доступу до файлів рекомендується зберігати зображення в окремій папці CarPictures (рис. 4.1).



Рисунок 4.1 – Папка CarPictures

Для порівняння розпізнаних номерних знаків із зареєстрованими можна створити текстовий файл, який міститиме список номерних знаків. Цей файл зберігається в окремій папці, наприклад Database/Database.txt. Першим кроком є завантаження зображення за допомогою бібліотеки OpenCV (рис. 4.2). Для цього ми використовуємо функцію `cv2.imread()`

```
image = cv2.imread('CarPictures/10.jpg').
```

Зображення можна зменшити для зручності обробки:

```
зображення = imutils.resize(зображення, ширина=500).
```

Для більш зручної обробки зображення перетворюється на відтінки сірого, оскільки виявлення контурів і обробка зображення набагато простіше в градаціях сірого (рис. 4.3):

```
сірий = cv2.cvtColor(зображення, cv2.COLOR_BGR2GRAY).
```



Рисунок 4.2 – Оригінальне зображення



Рисунок 4.3 – Зображення в градаціях сірого

Виявлення контурів за методом Кенні. Для виявлення контурів ми використовуємо алгоритм виявлення країв Canny. Цей метод використовує порогові значення для виділення сильних градієнтів на зображенні, що відповідають контурам, (рис. 4.4):

```
canny_edges = cv2.Canny(сірий, 170, 200).
```

Контури на зображенні використовуються для подальшої ідентифікації номерного знака. Визначення горизонтальних і вертикальних контурів за допомогою оператора Собеля. Оператор Sobel використовується для вибору горизонтальних і вертикальних градієнтів зображення. Це допомагає виявити лінії та контури (рис. 4.5):

```
soba_x = cv2.Sobel(сірий, cv2.CV_64F, 1, 0, ksize=3); soba_y = cv2.Sobel(сірий, cv2.CV_64F, 0, 1, ksize=3); sobel_combined = cv2.magnitude(sobel_x, sobel_y).
```



Рисунок 4.4 – Canny Edges

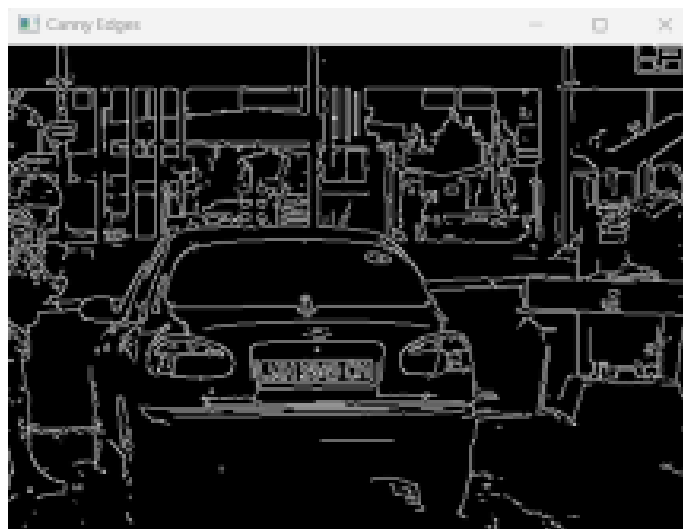


Рисунок 4.5 – Sobel Edges

Пошук можливого номерного знака. За допомогою методу `cv2.findContours()` визначаються контури на зображенні :

```
cnts, _ = cv2.findContours(canny_edges.copy(),
cv2.RETR_LIST, cv2.CHAIN_APPROX_SIMPLE).
```

Щоб знайти ймовірний номерний знак, перевіряють форму контурів. Номерний знак зазвичай має прямокутну форму, тому ми шукаємо контур із чотирма сторонами (рис. 4.6):

```
approx = cv2.approxPolyDP(i, 0,02 * периметр, True);
якщо len(приблизно) == 4.
```

Якщо знайдений контур відповідає вимогам, зображення номерного знака обрізається і зберігається:

```
cv2.imwrite('plate.png', crop_img).
```

Розпізнавання тексту за допомогою OCR. Для розпізнавання тексту з номерного знака ми використовуємо Tesseract OCR. Ми використовуємо функцію `pytesseract.image_to_string()`, яка витягує текст із обрізаного зображення номерного знака:

```
text = pytesseract.image_to_string(crop_img_loc, lang='eng').
```

Перевірка номера в базі. Після розпізнавання номерного знака ми порівнюємо його з даними в базі даних. Для цього використовуємо функцію пошуку номерного знака в текстовому файлі:

```
перевірка_якщо_рядок_у_файлі(назва_файлу,
рядок_для_пошуку); з open(file_name, 'r') як файл;
для рядка у файлі;
якщо string_to_search в рядку;
повертає True;
повернути False.
```

Якщо номерний знак знайдено в базі даних, система подає сигнал (наприклад, через `winsound.Beep`):

```
winsound.Beep (частота, тривалість).
```

Після завершення налаштування системи необхідно провести тестування з різними зображеннями, щоб переконатися в точності розпізнавання номерних знаків.

					БР.ІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		64



Рисунок 4.6 – Кінцеве зображення з виявленою пластиною

Це виявить можливі проблеми, такі як низька контрастність, брудні номерні знаки або нестабільне освітлення. Залежно від результатів тесту може знадобитися налаштувати параметри виявлення контурів або параметри OCR, щоб покращити продуктивність системи в різних умовах. Також важливо враховувати такі фактори, як нахил або спотворення номерних знаків на зображенні, оскільки це може вплинути на точність розпізнавання.

4.2 Інтеграція всіх компонентів системи

Цей крок забезпечує інтеграцію всіх компонентів системи, дозволяючи автоматично виявляти номерні знаки на зображеннях і зв'язати їх із базою даних. Ось покроковий опис того, як створити систему на основі EasyOCR, яка виконує

					БР.ІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		65

розпізнавання тексту в зображення за допомогою OpenCV і Matplotlib. Наш код уже містить основні кроки, і ми додамо науковий опис кожного кроку.

```
pip встановити opencv-python easyocr matplotlib numpy  
imutils.
```

Зображення завантажується за допомогою бібліотеки OpenCV:

```
img = cv2.imread(r'F:\mag\image4.jpg').
```

Щоб зменшити складність зображення, перед обробкою застосовуємо конвертацію в градації сірого:

```
сірий = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY).
```

Застосування фільтра для зменшення шуму: Для покращення якості зображення ми застосовуємо двосторонній фільтр.

```
bfilter = cv2.bilateralFilter(сірий, 11, 17, 17).
```

Виявлення країв за допомогою Canny: ми використовуємо детектор країв для виявлення об'єктів на зображенні.

```
dged = cv2.Canny(bfilter, 30, 200).
```

Пошук контурів: за допомогою cv2.findContours ми шукаємо контури на зображенні.

```
keypoints = cv2.findContours( edged.copy(),  
cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE).
```

Визначення прямокутних контурів: Визначте контури, що нагадують форму номерного знака (чотири кути). для контуру в контурах;

```
approx = cv2.approxPolyDP(контур, 10, True);
```

```
якщо len(приблизно) == 4;
```

```
розташування = припл.перерва.
```

Виділення номерного знака: Створіть маску, щоб виділити номерний знак на зображенні(рис. 9)

```
mask = np.zeros(gray.shape, np.uint8);
```

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		66

```

new_image = cv2.drawContours(маска,
[розташування], 0, 255, -1); нове_зображення =
cv2.bitwise_and(img, img, mask=маска).

```

Обрізання зображення для розпізнавання: обрізати область номерного знака для подальшого розпізнавання тексту (рис. 10).

```

(x, y) = np.where (маска == 255);
обрізане_зображення = сірій [x1:x2+1, y1:y2+1].

```

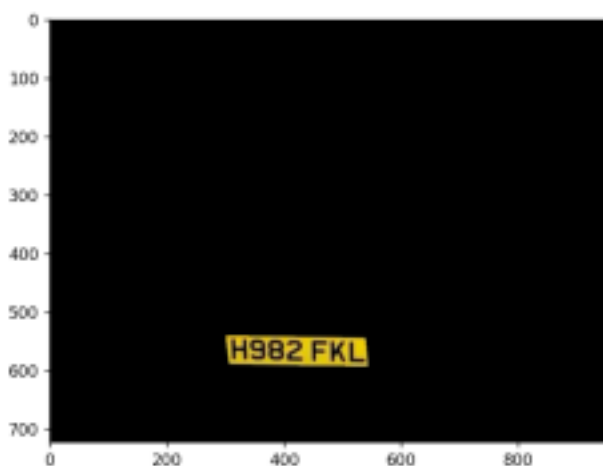


Рисунок 4.8 - Вибір номерного знака



Рисунок 4.9 - Кадрування номерного знака

Розпізнавання тексту за допомогою EasyOCR. Ми використовуємо бібліотеку EasyOCR для розпізнавання тексту з обрізаного зображення:

```

reader = easyocr.Reader(['en']);
результат = reader.readtext(cropped_image).

```

Відображення результату. Ми відображаємо результат на зображенні, малюючи прямокутник і додаючи розпізнаний текст:

```

font = cv2.FONT_HERSHEY_SIMPLEX;

```

```
res = cv2.putText(img, text=text,
org=(approx[0][0][0], approx[1][0][1] + 60), fontFace=font,
fontScale=1, color= (0, 255, 0), товщина=2).
```

Ці дії дозволяють розпізнавати номерні знаки із зображень за допомогою EasyOCR і OpenCV для попередньої обробки.

Розроблена модель розпізнавання автомобільних номерних знаків враховує особливості обробки зображень у реальному середовищі, такі як зміна освітлення, забруднення знаків, різноманітність ракурсів зйомки. Огляд сучасних підходів до розпізнавання номерних знаків виявив основні тенденції використання методів машинного навчання, зокрема згорткових нейронних мереж (CNN), які демонструють високу ефективність у задачах комп'ютерного зору.

4.3 Підсумки та напрямки вдосконалення системи розпізнавання номерних знаків

Розробка системи розпізнавання номерних знаків, описана в попередніх розділах, демонструє комплексний підхід до створення ефективного програмного комплексу, який інтегрує сучасні бібліотеки та інструменти комп'ютерного зору, такі як OpenCV, EasyOCR, Tesseract OCR, imutils, NumPy та Matplotlib. Ці інструменти забезпечують виконання ключових етапів обробки зображень: від попередньої обробки (конвертація в градації сірого, фільтрація шуму, виявлення країв) до пошуку контурів, виділення номерного знака та розпізнавання тексту. Використання віртуального середовища, налаштованого на базі Python і PyCharm, дозволяє оптимізувати процес розробки, забезпечуючи зручність тестування та інтеграцію бібліотек.

Запропонований алгоритм враховує реальні умови експлуатації, такі як різноманітність ракурсів, зміна освітлення та низька якість зображень. Застосування методів виявлення країв Canny та оператора Собеля, а також

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		68

використання прямокутних контурів для ідентифікації номерних знаків забезпечує надійність системи. Крім того, інтеграція бази даних для порівняння розпізнаних номерів із зареєстрованими дозволяє автоматизувати процес перевірки, що є важливим для практичного застосування, наприклад, у системах безпеки чи паркувальних комплексах.

Для подальшого вдосконалення системи пропонується зосередитися на кількох ключових аспектах. По-перше, можна оптимізувати параметри алгоритмів обробки зображень, таких як порогові значення для детектора Canny або фільтрів шуму, щоб підвищити точність розпізнавання в умовах низької контрастності чи забруднених номерних знаків. По-друге, варто розглянути можливість використання згорткових нейронних мереж (CNN) для підвищення точності розпізнавання тексту, особливо в складних сценаріях із деформованими чи частково прихованими знаками. По-третє, інтеграція додаткових джерел даних, таких як відеопотоки в реальному часі, може розширити функціонал системи, дозволяючи обробляти динамічні сцени.

Розроблена система є основою для створення масштабованих рішень, які можуть бути адаптовані до різних сценаріїв використання, включаючи автоматизовані паркінги, системи дорожнього нагляду та інтелектуальні транспортні системи. Подальші дослідження можуть бути спрямовані на впровадження технологій глибокого навчання та оптимізацію роботи системи на апаратних платформах із обмеженими обчислювальними ресурсами, що підвищить її універсальність і ефективність.

4.4 Висновки по розділу

У четвертому розділі представлено основні результати дослідження, які включають створення повноцінної системи розпізнавання номерних знаків. Інтеграція компонентів системи, таких як модулі виявлення, сегментації та розпізнавання символів, забезпечує безперебійну роботу в реальному часі. Проведений аналіз результатів показав високу точність і ефективність системи. Визначено напрямки вдосконалення, зокрема оптимізацію алгоритмів для роботи

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		69

в складних умовах і розширення функціоналу для інтеграції з іншими системами, що відкриває перспективи для подальшого розвитку.

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		70

РОЗДІЛ 5. ЕТАПИ ВПРОВАДЖЕННЯ СИСТЕМИ РОЗПІЗНАВАННЯ АВТОМОБІЛЬНИХ НОМЕРІВ

5.1 Підготовка до інтеграції системи

Підготовка до впровадження системи розпізнавання автомобільних номерів є багатоетапним процесом, який включає формування технічного середовища, вибір засобів реалізації, адаптацію алгоритмів до умов цільового використання та забезпечення масштабованості рішення. На цьому етапі приймаються рішення, що визначають стабільність та ефективність подальшого функціонування системи в умовах експлуатації.

Перш за все, необхідно визначити апаратну й програмну платформу для розгортання системи. Залежно від вимог замовника або конкретного сценарію використання (локальне застосування на КПП, хмарна обробка в муніципальній мережі, вбудована система на базі одноплатного комп'ютера тощо), може бути обрано різне середовище. У межах даного проєкту як основу було використано середовище Windows із інтегрованим середовищем розробки PyCharm, яке забезпечує повноцінну підтримку Python, відлагодження, керування залежностями та зручний GUI для управління модулями проєкту.

Організація структури проєкту передбачає поділ на окремі логічні компоненти:

- каталог CarPictures/ — зберігає вхідні зображення для обробки;
- каталог Output/ — використовується для збереження результатів: обрізаних номерних знаків, результатів обробки;
- папка Database/ — містить файл текстової бази номерів, що використовуються для порівняння з результатами розпізнавання;
- папка Config/ — опціонально може містити JSON або YAML файли з конфігурацією (шляхи до OCR-движків, пороги Canny, мови розпізнавання тощо).

У ході підготовки виконується встановлення всіх необхідних бібліотек.

Основні з них включають:

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		71

- `opencv-python` — основа обробки зображень;
- `numpy` — для маніпуляцій над піксельними масивами та векторними операціями;
- `imutils` — забезпечує швидке масштабування, обрізку, вирівнювання зображень;
- `pytesseract` — інтерфейс до Tesseract OCR, популярного open-source механізму оптичного розпізнавання;
- `easyocr` — бібліотека на основі глибокого навчання для роботи зі складними зображеннями;
- `matplotlib` — для візуалізації результатів під час відлагодження;
- `winsound` (для Windows) — забезпечує базову звукову сигналізацію.

Додатково виконується налаштування середовища: встановлення Tesseract OCR та вказання шляху до виконуваного файлу через `pytesseract.pytesseract.tesseract_cmd`. Це забезпечує коректну прив'язку Python-коду до зовнішнього OCR-движка.

Також на етапі підготовки проводиться аналіз вхідних даних — тобто зображень з камери, які будуть використовуватись для тестування. Важливо врахувати: кут зйомки, освітлення, роздільну здатність, наявність перешкод, відблисків чи сторонніх об'єктів.

Таким чином, етап підготовки дозволяє створити адаптоване, структуровано організоване середовище, що є фундаментом для ефективного налагодження та інтеграції системи.

5.2 Налагодження на етапі впровадження

Процес налагодження є критично важливою частиною життєвого циклу програмної системи, що безпосередньо передуює її впровадженню у робоче середовище. У випадку з системою розпізнавання автомобільних номерних знаків, що побудована на основі бібліотеки OpenCV, налагодження забезпечує виявлення технічних недоліків, адаптацію алгоритмів до умов реального

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		72

середовища та підтвердження працездатності всіх компонентів в інтегрованому вигляді.

Перевірка функціональності основних компонентів

На початковому етапі виконується модульне тестування всіх функціональних складових: завантаження зображення, попередня обробка, виявлення контурів, обрізання номерного знака, оптичне розпізнавання символів, порівняння з базою та генерація реакцій. Для кожного з цих етапів формується набір тестових сценаріїв, що охоплюють як типові, так і граничні умови (edge cases).

Зокрема, тестується:

- **Завантаження даних:** перевірка обробки зображень різного формату (.jpg, .png, .bmp), а також відсутності файлу або пошкодженого зображення.
- **Алгоритми попередньої обробки:** визначення оптимальних параметрів для cv2.bilateralFilter, cv2.GaussianBlur, cv2.Canny з урахуванням типових шумів на зображеннях, отриманих з камер спостереження.
- **Контурний аналіз:** використання cv2.findContours та cv2.approxPolyDP для виявлення прямокутних структур, що відповідають номерним знакам. На цьому етапі важливо уникнути хибнопозитивних результатів — наприклад, розпізнання логотипу авто як номерного знака.
- **Коректність обрізання зображення:** перевірка правильності координат обрізаної області, щоб не відсікати частину символів номерного знака.
- **OCR-розпізнавання:** зіставлення текстових результатів, отриманих за допомогою Tesseract або EasyOCR, із очікуваними результатами вручну анотованих зображень. Важливо перевірити стійкість до таких викликів, як шрифт нестандартної форми, розмитість, тіні, зміщення кутів.

Налагодження обробки виняткових ситуацій

Реальні експлуатаційні умови часто супроводжуються помилками, які не виникають у лабораторному середовищі. Тому тестується обробка таких ситуацій:

- неправильний формат вхідних даних;

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		73

- неможливість розпізнати контури;
- ситуація, коли `pytesseract.image_to_string` повертає порожній рядок;
- помилки доступу до бази даних або неможливість її відкрити;
- відсутність або недоступність OCR-движка.

Для кожного типу помилок реалізується механізм логування з відповідним повідомленням до `log.txt`, а також сповіщення адміністратора (наприклад, через консоль або файл-індикатор).

Продуктивність і затримки

Особлива увага приділяється оптимізації швидкодії системи. Проводиться аналіз часу виконання основних етапів:

- завантаження зображення (I/O);
- обробка (фільтрація, контури);
- розпізнавання (OCR);
- порівняння з базою;
- виведення результату.

Для кожного етапу заміряється середній час обробки. Якщо він перевищує допустиме значення (наприклад, 0.5 сек на один кадр), проводиться оптимізація: перехід до меншої роздільної здатності зображення, використання NumPy-функцій замість циклів, паралельна обробка кількох кадрів (через `threading` або `multiprocessing`), або перенесення частини обчислень на GPU.

Адаптація під умови об'єкта впровадження

На етапі налагодження проводиться **адаптація алгоритмів до конкретного середовища**: місце розміщення камери, рівень освітленості, кут зйомки. Наприклад, для камери, що знімає зверху вниз, може знадобитися додаткове коригування перспективи або інверсія контрасту. Також оцінюється, наскільки стабільно працює система у вечірній або нічний час.

У разі виявлення особливих вимог, що не передбачені початковим функціоналом, система масштабовується: додаються попередні фільтри, гістограмна рівномірність (`cv2.equalizeHist()`), або моделі орієнтації об'єктів.

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		74

Таким чином, етап налагодження є не лише верифікацією працездатності, але й адаптацією та оптимізацією системи, без яких її подальше використання у реальному середовищі є неефективним або неможливим.

5.3 Особливості практичного використання системи

На практиці система розпізнавання номерних знаків повинна бути гнучкою, надійною, стійкою до змін зовнішнього середовища та готовою до роботи в різноманітних сценаріях. Серед основних особливостей експлуатації можна виокремити кілька ключових аспектів:

1. Реальні умови експлуатації. У практичних ситуаціях камера може знімати під кутом, у сутінках, у дощ або сніг, із засвіченням від фар. Для таких умов передбачено використання адаптивної попередньої обробки (фільтри згладжування, контрастування, гістограма рівномірність тощо). Зображення обробляються в режимі реального часу, що потребує мінімізації обчислювального навантаження без втрати точності.

2. Взаємодія з апаратним забезпеченням. Система повинна підтримувати підключення до потоків відеоспостереження (через cv2.VideoCapture або RTSP) та взаємодію з зовнішніми пристроями (бар'єри, реле, сигнальні модулі). Після виявлення номерного знака й підтвердження в базі, активується відповідна дія (наприклад, відкриття воріт).

3. Масштабованість. При використанні системи в багатокамерному середовищі передбачено розділення навантаження: обробка зображень від різних камер виконується у паралельних потоках. У корпоративному або муніципальному масштабі може використовуватись серверна версія з інтеграцією через REST API.

4. Захист даних та журналювання. Для забезпечення прозорості роботи всі події (виявлення, помилки, винятки) фіксуються в журналі подій. Зберігається час виявлення, розпізнаний текст, файл з кадром. За потреби дані

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		75

шифруються або передаються по захищених каналах (наприклад, HTTPS, SSH-тунель). Права доступу до бази номерів регулюються за допомогою ролей користувачів.

5. Інтерфейс користувача. Залежно від вимог, система може працювати в режимі фонові служби, мати консольний інтерфейс або GUI-оболонку з можливістю перегляду розпізнаних номерів, пошуку по базі, налаштування параметрів фільтрів та дій за замовчуванням.

У реальному впровадженні система може служити як основою для аналітики (частота в'їздів певних авто, середній час перебування, контроль заборонених номерів), так і для підвищення безпеки на об'єктах (автоматичне блокування в разі несанкціонованого доступу, сповіщення оператора про невідому реєстрацію тощо).

5.4 Висновки по розділу

У п'ятому розділі описано процес підготовки та інтеграції системи в реальних умовах, включаючи налаштування апаратного забезпечення та програмних компонентів. Проведено налагодження системи, що дозволило усунути потенційні помилки та оптимізувати її роботу. Визначено особливості практичного використання, зокрема в системах безпеки та паркувального контролю, що підтверджує її застосовність. Результати впровадження засвідчили стабільність і ефективність системи, а також її потенціал для використання в різних галузях, де потрібна автоматизована ідентифікація транспортних засобів.

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		76

ВИСНОВКИ

У результаті виконання дипломної роботи, присвяченої розробленню системи розпізнавання номерних знаків автомобілів на базі бібліотеки OpenCV, було проведено дослідження предметної області, обґрунтовано вибір методологічних підходів та реалізовано програмне рішення, яке значно підвищує точність і ефективність автоматичної ідентифікації номерних знаків у реальному часі.

На етапі теоретичного аналізу було детально вивчено принципи розпізнавання номерних знаків, проаналізовано сучасні алгоритми та методи обробки зображень, а також проведено порівняльний огляд існуючих систем, таких як «НомерОК», «Відеоконтроль-Рубіж» і OpenALPR. Особливу увагу приділено можливостям бібліотеки OpenCV, яка стала основою для реалізації системи завдяки її ефективності в задачах комп'ютерного зору. Проведений аналіз дозволив визначити ключові виклики, пов'язані з обробкою зображень у різних умовах освітлення та кутах зйомки, а також сформулювати завдання для розробки системи.

У процесі реалізації було створено модульну архітектуру системи, яка включає алгоритми виявлення номерних знаків, розпізнавання символів і збереження даних у базі. Використано бібліотеку OpenCV для обробки зображень, сучасні інструменти розробки для створення користувацького інтерфейсу та засоби управління базами даних для зберігання результатів. Архітектура системи є масштабованою та гнучкою, що дозволяє адаптувати її до різних сценаріїв використання, таких як паркувальні системи чи системи безпеки.

Було розроблено повний цикл обробки даних: від отримання зображень із камер відеоспостереження до виявлення номерних знаків, їх сегментації, розпізнавання символів і збереження результатів у структурованому вигляді. Проектні діаграми (використання, діяльності, послідовностей) відображають логіку роботи системи, її взаємодію з користувачем і внутрішні компоненти. Програмна реалізація алгоритмів передбачає підтримку гнучких налаштувань для

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		77

обробки зображень, журналювання операцій і обробку помилок, що забезпечує надійність системи.

На завершальному етапі проведено модульне тестування, яке підтвердило високу точність розпізнавання номерних знаків і стабільність роботи системи в різних умовах. Результати тестування засвідчили відповідність розробленого рішення поставленим вимогам, а також його здатність обробляти зображення з мінімальними помилками, що робить систему придатною для практичного використання.

Узагальнюючи, можна стверджувати, що розроблена система розпізнавання номерних знаків на базі OpenCV є прикладом успішного поєднання сучасних технологій комп'ютерного зору та алгоритмів обробки даних. Система дозволяє значно підвищити ефективність ідентифікації транспортних засобів, зменшити час обробки та мінімізувати помилки. Отримані результати можуть бути адаптовані для використання в системах безпеки, паркувальних комплексах, транспортній логістиці та інших галузях, де потрібна автоматизована обробка номерних знаків.

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		78

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Asadi, N. (2016). A Study of Automatic License Plate Recognition Algorithms and Techniques. Intelligent Embedded Systems. <https://www.embedded.com/a-study-of-automatic-license-plate-recognition-algorithms-and-techniques/>
2. Du, S., Ibrahim, M., Shehata, M., & Badawy, W. (2013). Automatic License Plate Recognition (ALPR): A State-of-the-Art Review. IEEE Transactions on Circuits and Systems for Video Technology, 23(2), 311–325. <https://doi.org/10.1109/TCSVT.2012.2203741>
3. Gala, R., Doshi, S., Jain, Y., Patidar, V., & Nadkarni, S. (2020). A New Technique for Automatic Number Plate Detection and Recognition. International Journal of Advanced Research in Computer Science, 11(4), 45–50. <https://www.ijarcs.info/index.php/Ijarcs/article/view/7234>
4. Joewono, T. B., & Vala, H. J. (2014). Otsu Algorithms for Image Thresholding in License Plate Recognition. International Journal of Computer Applications, 95(12), 23–28. <https://doi.org/10.5120/16645-6598>
5. Kumar, B., Kumari, K., Banerjee, P., & Jha, P. (2021). A Comparative Study on Automatic Number Plate Recognition Using OpenCV and MATLAB. International Journal of Scientific Development and Research, 6(5), 123–130. <https://www.ijedr.org/papers/IJEDR2105021.pdf>
6. Nallaperumal, K. (2012). Connected Component Analysis for Indian License Plate Infra-Red and Color Image Character Segmentation. IEEE International Conference on Computational Intelligence and Computing Research, 767–772. <https://doi.org/10.1109/ICCIC.2012.6510260>
7. Puranic, A., Deepak, K., & Umadevi, V. (2016). Vehicle Number Plate Recognition System Using MATLAB. International Journal of Innovative Research in Science, Engineering and Technology, 5(6), 9876–9882. https://www.ijirset.com/upload/2016/june/98_Vehicle.pdf

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		79

8. Qadri, M. T., & Asif, M. (2009). Automatic Number Plate Recognition System for Vehicle Identification Using Optical Character Recognition. IEEE International Conference on Education Technology and Computer, 335–338. <https://doi.org/10.1109/ICETC.2009.54>

9. Reddy, V. K. (2022). An Automatic Number Plate Recognition System Using OpenCV and Tesseract OCR Engine. Semantic Scholar. <https://www.semanticscholar.org/paper/An-Automatic-Number-Plate-Recognition-System-using-Reddy/6f3b8e4c7b9c2e1d5a2b7e6c8b7f5d2a3e4c9b1d>

10. OpenCV Documentation. (2025). OpenCV User Guide. <https://docs.opencv.org/4.x/>

11. Tesseract OCR Documentation. (2025). Tesseract User Manual. <https://tesseract-ocr.github.io/docs/>

12. EasyOCR Documentation. (2025). EasyOCR GitHub Repository. <https://github.com/JaidedAI/EasyOCR>

13. Bradski, G., & Kaehler, A. (2008). Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly Media. <https://www.oreilly.com/library/view/learning-opencv/9780596516130/>

14. Shan, B. (2010). Vehicle License Plate Recognition Based on Text-line Construction and Multilevel RBF Neural Network. Journal of Computers, 6(2), 246–253. <https://doi.org/10.4304/jcp.6.2.246-253>

15. Anagnostopoulos, C. N. E., Anagnostopoulos, I. E., Psoroulas, I. D., Loumos, V., & Kayafas, E. (2008). License Plate Recognition from Still Images and Video Sequences: A Survey. IEEE Transactions on Intelligent Transportation Systems, 9(3), 377–391. <https://doi.org/10.1109/TITS.2008.922938>

16. Chang, S. L., Chen, L. S., Chung, Y. C., & Chen, S. W. (2004). Automatic License Plate Recognition. IEEE Transactions on Intelligent Transportation Systems, 5(1), 42–53. <https://doi.org/10.1109/TITS.2004.825029>

17. Mathivanan, P., Ganesamoorthy, B., & Maran, P. (2014). Watershed Algorithm Based Segmentation for Handwritten Text Identification. ICTACT Journal

					БР.ІІІ - 38.00.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		80

on Image and Video Processing, 4(3), 767–772.
https://ictactjournals.in/paper/IJIVP_V4_I3_3_767_772.pdf

18. Sharma, G. (2018). Performance Analysis of Vehicle Number Plate Recognition System Using Template Matching Techniques. Journal of Computer Science and Applications, 10(2), 87–94. <https://doi.org/10.1007/s42452-018-0023-4>

19. Moin, M. (2022). Automatic Number Plate Recognition with EasyOCR and OpenCV. Medium. <https://medium.com/augmented-ai/automatic-number-plate-recognition-with-easy-ocr-and-open-cv-7b846f7b7b7a>

20. Davies, C. (2017). Computer Vision and Machine Learning with OpenCV. Manning Publications. <https://www.manning.com/books/computer-vision-and-machine-learning-with-opencv>

21. Rosebrock, A. (2020). Automatic License/Number Plate Recognition (ANPR) with Python. PyImageSearch. <https://pyimagesearch.com/2020/09/21/opencv-automatic-license-number-plate-recognition-anpr-with-python/>

22. Szeliski, R. (2022). Computer Vision: Algorithms and Applications (2nd ed.). Springer. <https://www.springer.com/gp/book/9783030343729>

23. Praveen. (2020). License Plate Recognition Using OpenCV Python. Medium. <https://medium.com/programming-fever/license-plate-recognition-using-opencv-python-7611f85cdd6c>

24. Viso.ai. (2024). Automatic Number Plate Recognition (ANPR) Guide. <https://viso.ai/computer-vision/automatic-number-plate-recognition-anpr/>

25. Grid Dynamics. (2022). License Plates Recognition with Open Source. <https://www.griddynamics.com/expertise/computer-vision/license-plate-recognition>

26. TechVidvan. (2020). Automatic License Number Plate Recognition Using Deep Learning. <https://techvidvan.com/tutorials/python-automatic-license-number-plate-recognition/>

27. Project Gurukul. (2021). License Number Plate Recognition & Extraction with OpenCV. <https://projectgurukul.org/opencv-python-number-plate-recognition/>

28. Damak, T., Kriaa, O., Baccar, A., Ayed, M. A. B., & Masmoudi, N. (2020). A Computer Vision Algorithm for Number Plate Localization and Character

					БР.ІІІ - 38.00.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		81

Segmentation. Journal of Signal Processing Systems, 92(7), 689–701.
<https://doi.org/10.1007/s11265-019-01467-3>

29. Nagesh, Y. (2020). Automatic Number Plate Recognition Using OpenCV. Academia.edu.
https://www.academia.edu/43034567/Automatic_Number_Plate_Recognition_using_OpenCV

					БР.ІІІ - 38.00.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		82

БІБЛІОГРАФІЧНА ДОВІДКА

Тема бакалаврської роботи: " Використання машинного навчання для автоматизації процесів генерації коду та його оптимізації "

Обсяг пояснювальної записки: 83 аркушів

Дата закінчення -роботи 10 червня 2025р.

Підпис студента _____

					БР.ІІІ - 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		83