

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій**

**Індивідуальний дослідницький проєкт
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Інформаційне забезпечення
робототехнічних систем»
спеціальності 126 «Інформаційні системи та технології»
на тему: «Система контролю доступу для автотранспорту»**

Виконав:

студент IV курсу, групи ІК-82

Рахуба Євгеній Русланович _____

Керівник:

доцент кафедри інформаційних
систем та технологій ФІОТ

Олійник Володимир Валентинович _____

Засвідчую, що у цьому проєкті немає
запозичень з праць інших авторів без
відповідних посилань.

Студент _____

Київ – 2022 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційне забезпечення робототехнічних систем»

ЗАВДАННЯ

на індивідуальний дослідницький проєкт студенту

Рахубі Євгенію Руслановичу

1. Тема проєкту «Система контролю доступу для автотранспорту», керівник проєкту доцент кафедри інформаційних систем та технологій ФІОТ Олійник Володимир Валентинович.
2. Термін подання студентом проєкту: 15 червня 2022 року
3. Вихідні дані до проєкту: наукова література та інші інформаційні джерела з теми, існуючі програмні рішення, технічне завдання для створення власного програмного забезпечення для ведення відомостей і звітності.
4. Зміст пояснювальної записки:
 - Вступ
 - 1. Аналіз предметної області і постановка задачі.
 - 2. Опис стеку технологій.
 - 3. Розробка та тестування програмного забезпечення.
 - 4. Висновки.
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо): інтерфейс додатку, алгоритм роботи програми.
6. Дата видачі завдання 1 грудня 2021 року

Календарний план

№ з/п	Назва етапів виконання проєкту	Термін виконання етапів проєкту	Примітка
1	Загальна характеристика задачі контролю доступу для автотранспорту	15.01.22 – 25.01.22	
2	Пошук та порівняння технологій для створення системи контролю доступу	01.02.22 – 23.02.22	
3	Архітектурне проектування системи	25.04.22 – 02.05.22	
4	Створення програмного забезпечення	03.05.22 – 12.05.22	
5	Тестування додатку	13.05.22 – 30.05.22	
6	Оформлення дипломної роботи	01.06.22 – 14.06.22	
7	Норм. контроль	16.06.22 – 19.06.22	
8	Перевірка на співпадіння	16.06.22 – 19.06.22	
9	Захист	20.06.22 – 24.06.22	

Студент

Рахуба Євгеній

Керівник

Володимир Олійник

АНОТАЦІЯ

Рахуба Є.Р. Система контролю доступу для автотранспорту. КПІ ім. Ігоря Сікорського, Київ, 2022.

Проект містить 58 с. тексту, 32 рисунки, 3 таблиці, посилання на 10 літературних джерел та 3 додатки.

Ключові слова: Python, CV, розпізнавання, контроль, додаток, система.

Об'єктом розробки є система контролю доступу для автотранспорту.

Мета розробки – підвищення ефективності вирішення задачі керування контрольно пропускним пунктом в напів-автоматичному режимі, зокрема розширення доступності таких систем за рахунок розробки програмного забезпечення.

У роботі було розглянуто проблеми контролю доступу для транспортних засобів. Визначено основні недоліки існуючих рішень та у відповідності до потреб було поставлено задачу на розробку власної системи контролю доступу для автотранспорту.

На основі обраних сучасних технологій було розроблено спеціальний настільний додаток, що дає можливість керування системою контролю доступу. Створеним програмним забезпеченням може користуватися будь-яка людина яка не має спеціальних знань. Програма дозволяє додавати та видаляти транспортні засоби до списку тих які мають можливість проїзду. В іншому програма працює в автоматичному режимі й виводить оброблену інформацію у додаток для можливості моніторингу.

ABSTRACT

Rakhuba Yevhenii Access control system for vehicles. KPI Igor Sikorsky, Kyiv, 2022.

The project contains 58 pages. text, 32 figures, 3 tables, references to 10 references and 3 appendices.

Keywords: Python, CV, recognition, control, application, system.

The object of development is an access control system for vehicles.

The purpose of development is to increase the efficiency of solving the problem of checkpoint management in a semi-automatic mode, in particular to increase the availability of such systems through software development.

The problems of access control for vehicles were considered in the work. Appropriate shortcomings of existing solutions and compliance have been identified in order to set the task of developing our own access control system for vehicles.

Based on selected modern technologies, a special desktop application was developed that allows you to control the access control system. The created software can be used by anyone who has no special knowledge. The program allows you to add and remove vehicles to the list of those who have the opportunity to travel. In another program it works in automatic mode and displays the processed information in the application for monitoring capabilities.

Пояснювальна записка
до дипломного проекту
на тему:
«Система контролю доступу для автотранспорту»

Київ – 2022 року

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ.....	9
ВСТУП	10
1 ЗАГАЛЬНА ХАРАКТЕРИСТИКА ЗАДАЧІ КОНТРОЛЮ ДОСТУПУ ДЛЯ АВТОТРАНСПОРТУ.....	12
1.1 Важливість використання систем контролю доступом.....	12
1.2 Аналіз підходів вирішення питання.....	14
1.2.1 Контурний аналіз.....	14
1.2.2 Аналіз частин контурів	15
1.2.3 Гістограмний аналіз регіонів.....	15
1.2.4 Класифікатори	16
1.2.5 Нейронні мережі	17
1.2.6 K-nearest	18
1.2.7 Tesseract OCR	18
1.2.8 East OCR	19
1.3 Аналіз існуючих рішень	19
1.4 Постановка реальної задачі	20
Висновки до розділу	21
2 АРХІТЕКТУРНЕ МОДЕЛЮВАННЯ СИСТЕМИ.....	22
2.1 Модель розпізнавання області номеру	22
2.2 Модель зчитування тексту з номерного знаку.....	27
2.3 Схема розробки програмного забезпечення.....	28
Висновки до розділу	31

					ІК82.190БАК.003 ПЗ				
Зм.	Аркуш	№ докум.	Підп.	Дата					
Розроб.	Рахуба Є.Р,				Система контролю доступу для автотранспорту Пояснювальна записка		Лім.	Аркуш	Аркушів
Керівн.	Олійник В.В.						у	7	58
							КПІ ім. Ігоря Сікорського ФІОТ група ІК-82		
Н. Контр.									
Затв.									

3 ТЕХНОЛОГІЇ ДЛЯ СТВОРЕННЯ ДОДАТКУ. РОЗРОБКА ТА	
ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	32
3.1 Вибір мови програмування.....	32
3.1.1 JavaScript.....	33
3.1.2 Java.....	34
3.1.3 Python.....	35
3.2 Вибір середовища розробки. PyCharm чи Sublime Text.....	37
3.3 База даних. SQLite	40
3.4 Короткий опис створення ПЗ.....	43
3.5 Тестування програмного забезпечення.....	46
3.5.1 Тестування підсистеми розпізнавання та зчитування номерного	
знаку.....	47
3.5.2 Тестування додатку.....	51
Висновки до розділу	55
ВИСНОВКИ.....	56
ПЕРЕЛІК ПОСИЛАНЬ.....	57
ДОДАТКИ.....	59

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

API — application programming interface

XML — extensible markup language

ООП — об'єктно-орієнтоване програмування

ОС — операційна система

CV – computer vision

БД – база даних

ПЗ – програмне забезпечення

					ІК82.190БАК.003 ПЗ	Лист
Зм.	Аркуш	№ докум.	Підп.	Дата		9

Вступ

Система контролю доступу для автотранспорту - це комплекс технічних та програмних засобів, що здійснює регулювання проїзду транспортних засобів на територіях з обмеженим доступом, для моніторингу та попередження несанкціонованого проникнення.

Кожен день такі системи заощаджують час проходження зони контролю завдяки автоматичним/напіваавтоматичним принципам роботи. Виключають людський фактор з багатьох процесів. Потребують мінімальну кількість співробітників, тому економлять кошти на заробітну плату. Окрім цього сприяють зниженню злочинності.

Такі системи почали появлятися достатньо давно та мали різні принципи дії, інструменти адміністрування, способи взаємодії з користувачем.

В нас час найактуальнішими є системи з автоматичним принципом дії за допомогою використання підсистеми розпізнавання номерних знаків авто. Ці системи зазвичай використовують на парковках торгівельно-розважальних центрів, великих підприємствах, житлових комплексах преміум сегменту тощо. В той час як найбільшу поширеність мають більш прості системи контролю доступу, наприклад за допомогою rfid міток, або подібних засобів. Це пов'язано з ціною програмного забезпечення, обладнання, підтримки роботи здатності.

Системи які використовують розпізнавання номерних знаків є ефективнішими та мають розширений функціонал, а найголовніше є більш захищеними від несанкціонованого доступу, та надають перевагу в разі здійснення неправомірних дій.

Саме тому, було прийнято рішення створити систему контролю доступу з використанням підсистеми розпізнавання та зчитування

					ІК82.190БАК.003 ПЗ	Лист
						10
Зм.	Аркуш	№ докум.	Підп.	Дата		

номерних знаків. Система має бути максимально простою, але мати весь необхідний функціонал для безвідмовної роботи в реальних умовах. Одна з головних цілей – дешевизна обладнання та його утримання, щоб забезпечити доступність для більшості дворів, стоянок та житлових кварталів.

Метою роботи є підвищення ефективності вирішення задачі керування контрольно пропускним пунктом в напів-автоматичному режимі, зокрема розширення доступності таких систем за рахунок розробки програмного забезпечення.

Для реалізації поставленої мети необхідно:

- проаналізувати підходи та існуючі рішення, визначити переваги та недоліки;
- проектування підсистеми розпізнавання та зчитування номерного знаку;
- проектування додатку з графічним інтерфейсом для користувача;
- тестування програмного забезпечення.

					ІК82.190БАК.003 ПЗ	Лист
Зм.	Аркуш	№ докум.	Підп.	Дата		11

1 ЗАГАЛЬНА ХАРАКТЕРИСТИКА ЗАДАЧІ КОНТРОЛЮ ДОСТУПУ ДЛЯ АВТОТРАНСПОРТУ

1.1 Важливість використання систем контролю доступом

Система контролю доступу автотранспорту з використанням підсистеми розпізнавання номерних знаків – сучасне, багатофункціональне та безпечне рішення. На рис. 1.1 можна побачити приклад роботи системи:

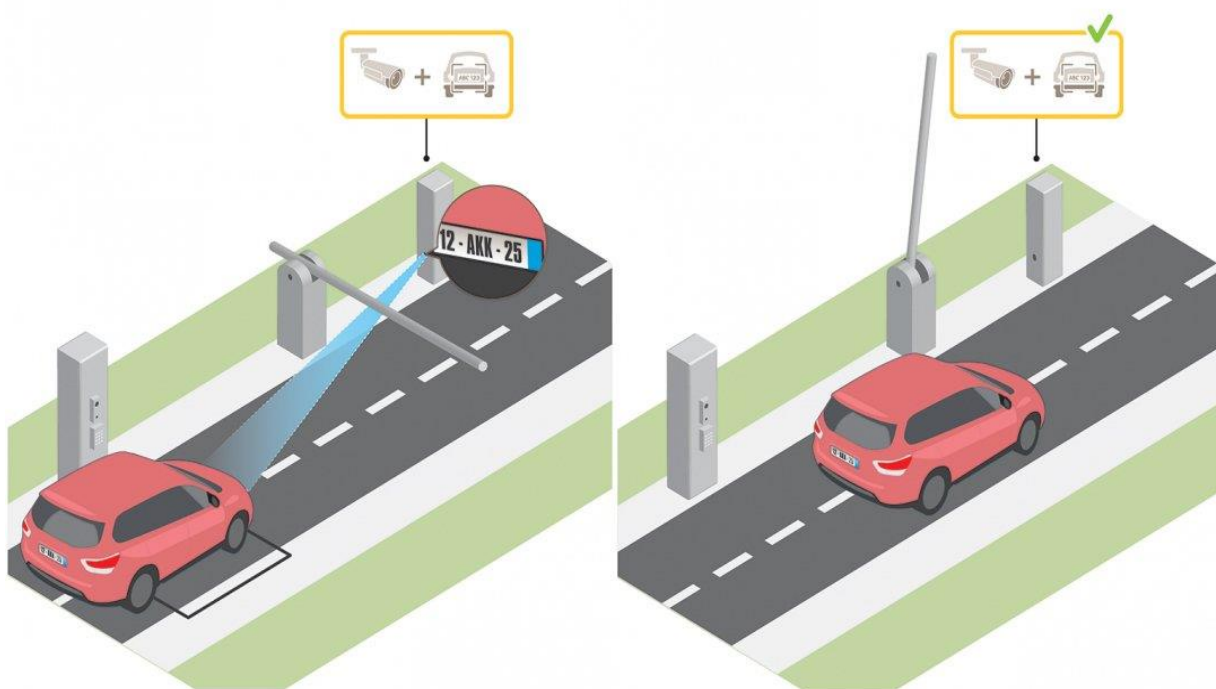


Рис 1.1 Приклад роботи системи [11]

При використанні таких систем для прибудинкових територій, житлових комплексів, дворів, парковок очікується отримати, по-перше порядок на цій території, а по-друге контроль паркувальних місць, що дуже актуально для таких міст як Київ, а також отримати зниження злочинності, так як транспортні засоби використовуються при 70 відсотках тяжких злочинів.

Системи такого типу в основному не доцільно ставити на невеликі території з маленьким потоком транспорту, так як програмне забезпечення,

його підтримка та обладнання мають значну вартість. Тому в таких випадках люди вимушені користуватися більш простими та менш ефективними системами.

Якщо вирішити проблему вартості, то автоматичні системи з розпізнаванням номеру набудуть більшої вживаності. Це дозволить використовувати їх для проектів з невеликим бюджетом.

Також такі системи можуть мати досить розширений функціонал (рис. 1.2), окрім прямої задачі (розпізнавання номеру та контролю доступу), наприклад:

- розпізнавання марки та моделі авто;
- розпізнавання типу транспортного засобу;
- розпізнавання кольору;
- фіксація часу заїзду/виїзду;
- перевірка номера по базі розшуку та ін.

Date/Time	Zone/Direction	Plate	Photo	Country	Brand	Model	Type
2019-11-08 08:44:21	1 ↓	5AN0212				superb	
2019-11-08 08:44:19	1 ↓	TM41TVO					
2019-11-08 08:44:18	1 ↓	3SU0138				corolla	
2019-11-08 08:44:17	1 ↓	6AM3062				fabia	
2019-11-08 08:44:16	1 ↑↓	4SE0604				ducato	
2019-11-08 08:44:16	1 ↓	3SI1501					
2019-11-08 08:44:14	1 ↓	5AS3868				sprinter	
2019-11-08 08:44:13	1 ↓	4AB2920				3	
2019-11-08 08:44:12	1 ↓	3AF2827				x3,x5,x6	
2019-11-08 08:44:11	1 ↓	4SK2231				berlingo	
2019-11-08 08:44:11	1 ↓	5Z68917					
2019-11-08 08:44:10	1 ↓	KHH5570				golf	

Рис 1.2 Приклад багатофункціональності програмного забезпечення [12]

1.2 Аналіз підходів вирішення питання

В основу системи контролю доступу лягає задача розпізнавання номерного знаку й подальша обробка для зчитування тексту з нього. Проаналізуємо різні алгоритми пошуку номерного знаку на зображенні. Але перед цим слід зазначити, що яким би точним не був алгоритм, потрібно мати високу якість зображення. Чим краще система зйомки – тим точніше результат.

Для пошуку та розпізнавання номерного знаку можуть бути використані різні алгоритми. Треба визначити їх основні переваги та недоліки.

1.2.1 Контурний аналіз

Один з самих очевидних та легких способів виділення номеру є пошук прямокутного контуру. Працює лише в ідеальних умовах, тобто номер має бути чітко видно та мати рівні контури.

Метод може мати багато хибних спрацювань. Якщо номер розташований на білому фоні його виявлення даним методом майже неможливе. Зазвичай використовується в студентських роботах для навчання обробки зображення. Приклад роботи методу контурного аналізу наведений на рис. 1.3.

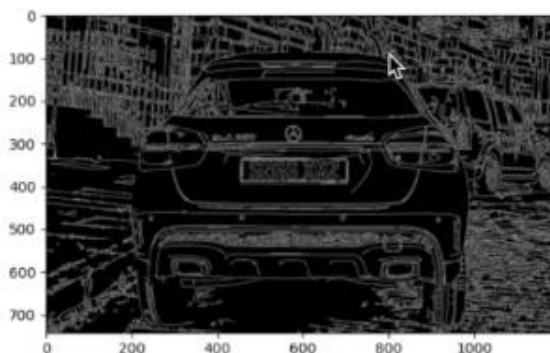


Рис. 1.3 Пошук контурів на зображенні [1]

1.2.2 Аналіз частини контурів

Цей метод також спочатку виділяє всі контури на зображенні, після чого відбувається пошук вертикальних прямих. Після чого для двох прямих, які знаходяться на невеликій відстані одна від одної з правильним відношенням сторін між вертикаллю та горизонталлю та відносно рівними сторонами будується припущення, що номер знаходиться в утвореному прямокутнику. Метод допускає невеликі відхилення по осям x, y .

Метод має більшу точність від попереднього, але має прив'язку до формату номерного знаку, а саме до відношення сторін. Це унеможливорює його швидку роботу в наших умовах. Приклад роботи методу аналізу частин контурів наведений на рис. 1.4.

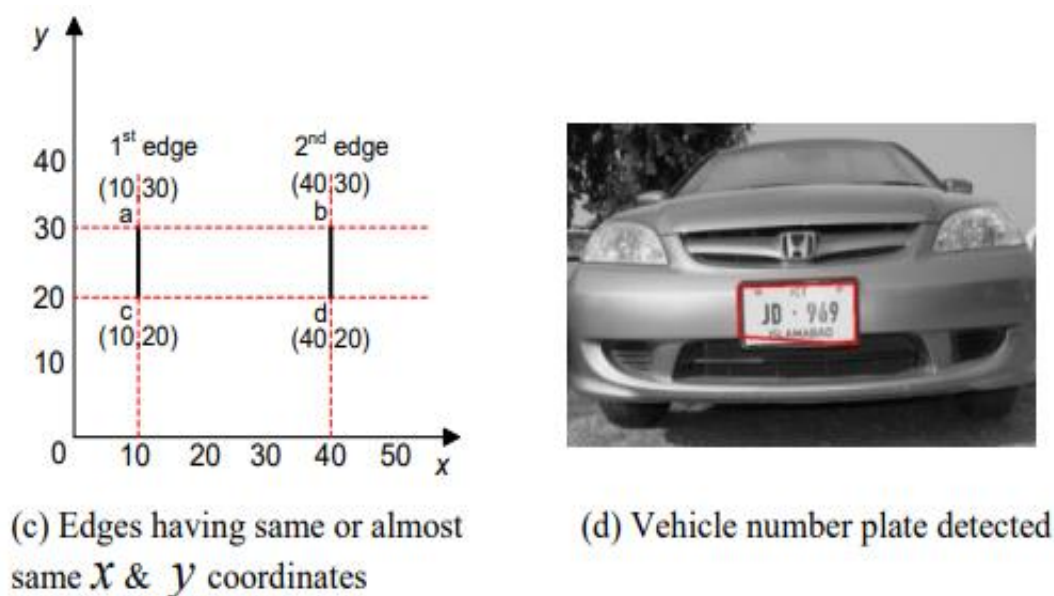


Рис. 1.4 Результат роботи алгоритму [9]

1.2.3 Гістограмний аналіз регіонів

Одним з популярних підходів у минулому був гістограмний аналіз зображення. Цей метод базується на припущенні, що частотна

характеристика місця з державним номерним знаком відрізняється від характеристики навколо нього.

На зображенні виділяються контури на основі яких будується проекція картинки на вісь у. Тоді максимум з отриманої проекції має відповідати положенню номера. Для того щоб цей метод мав високий показник результативності машина має бути на фоні без інших написів. Приклад роботи методу гістаграмного аналізу регіонів наведений на рис. 1.5.

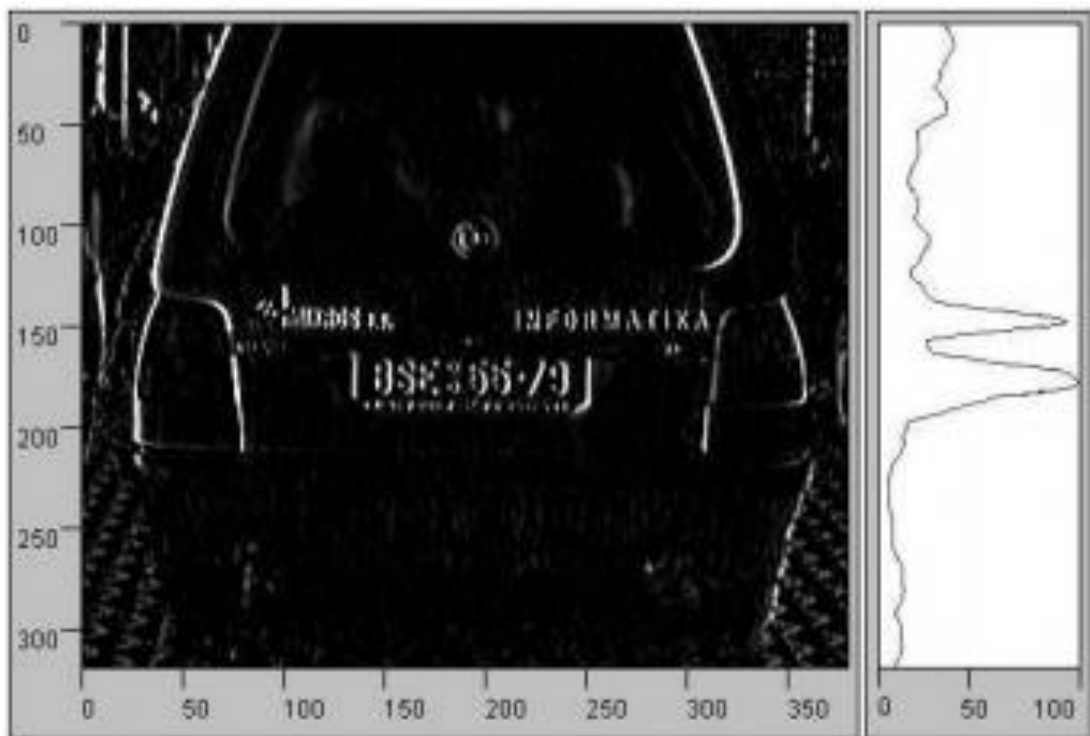


Рис. 1.5 Вертикальна проекція зображення на вісь у [1]

1.2.4 Класифікатори

Одними з найкращих вважаються методи які спираються на різні класифікатори. Гарним прикладом є навчений класифікатор – каскад Хаара. Класифікатори дозволяють аналізувати область на наявність в ній характерних ознак номерного знаку. Такі методи дозволяють знаходити

номер в більш складних умовах, під різними кутами, не чіткому зображенні. Приклад роботи каскадного класифікатора наведений на рис. 1.6.



Рис. 1.6 Відображення роботи класифікатора Хаара [2]

Використання класифікаторів дуже добре підходить для реальних умов, коли номер забруднено, або є ще якісь фактори які заважають розпізнаванню більш простими методами

1.2.5 Нейронні мережі

Нейронні мережі є найсучаснішим та найефективнішим методом розпізнавання номерного знаку. Вони також спираються на різні ознаки подібно до каскадних класифікаторів, але при правильному підході в проектуванні та навчанні на великій та різноманітній вибірці показують дуже гарні результати в реальних умовах.

Найбільше застосовуються в великих проектах з великими бюджетами, так як потребують великого та складного об'єму роботи. Приклад роботи нейронної мережі наведений на рис. 1.7.



Рис. 1.7 Результат роботи нейронної мережі в складних умовах [3]

1.2.6 K-nearest

Після знаходження номерного знаку перед нами постає задача розпізнавання символів які на ньому знаходяться. Одним з таких алгоритмів є K-nearest.

Це простий алгоритм для розпізнавання символів, але не дивлячись на свою примітивність працює доволі непогано. В теорії, при умові створення великої бази з прикладами символів в різних умовах видимості цей алгоритм буде давати непогані результати. Але метод матиме проблеми з розпізнаванням забруднених або потертих номерних знаків.

1.2.7 Tesseract OCR

Відкрите програмне забезпечення, яке вміє розпізнавати як одиничні символи так і текст. Tesseract стабільно працює та легко навчається. Але має ще гірші результати при тестуванні в реальних умовах. Розпізнає тільки ідеально чисті та правильно розташовані знаки.

					ІК82.190БАК.003 ПЗ	Лист
Зм.	Аркуш	№ докум.	Підп.	Дата		18

1.2.8 Easy OCR

За допомогою даного метода можна розпізнавати текст з достатньо великою точністю, в деяких випадках Easy OCR може позмагатись з комерційними рішеннями. Має зручні інструменти взаємодії, та має вбудований функціонал в python. Easy OCR на відміну від Tesseract OCR не потребує первинної підготовки зображення та підготує зображення самостійно для подальшого розпізнавання текстових блоків та тексту на них. Також має неточності при забрудненому номерному знаку.

1.3 Аналіз існуючих рішень

В сучасному світі ми можемо спостерігати використання систем контролю доступу на основі розпізнавання номерного знаку на парковках торгівельно-розважальних центрів, великих підприємствах та нерухомості преміум класу. Це пов'язано з тим що існуючі рішення потребують великих грошових затрат на купівлю софту, апаратної частини та обслуговування.

Наприклад програмно-апаратний комплекс «НомерОк» призначений для розпізнавання номерних знаків та управлінням виконуючими пристроями. Має досить високий процент розпізнавання, до 97 відсотків, Українських номерів та номерів більшості Європейських країн. Але головна проблема залишається – це ціна. Ціна на програмне забезпечення починається від 60 тисяч гривень на базову збірку і закінчується 270 тисячами гривень за повноцінний софт з розширеним функціоналом.

Подібні існуючі рішення мають схожу вхідну ціну, або високі вимоги до обладнання, що нівелює невисоку ціну програмного забезпечення.

					ІК82.190БАК.003 ПЗ	Лист
Зм.	Аркуш	№ докум.	Підп.	Дата		19

1.4 Постановка реальної задачі

Проаналізувавши існуючі рішення, потреби ринку та існуючі методи обробки зображень з автомобільними номерами можна зробити висновки для подальшого планування розробки програмного забезпечення для системи розпізнавання автомобільних номерів.

Мета дипломного проекту – підвищення ефективності вирішення задачі керування контрольно пропускним пунктом в напів-автоматичному режимі, зокрема розширення доступності таких систем за рахунок розробки програмного забезпечення.

Необхідні кроки для реалізації мети проекту:

- обрати метод розпізнавання номерного знаку;
- обрати метод зчитування державного номеру;
- обрати мову програмування;
- обрати середовище розробки;
- визначитись з основним стеком технологій;
- створення додатку для керування та моніторингу транспортних засобів на контрольно пропускному пункті;
- програмне забезпечення має містити весь необхідний функціонал, при цьому не мати нічого зайвого;
- ПЗ має бути легким в освоєнні, щоб не потребувати спеціальних навичок у персоналу;
- система не має потребувати високих показників від технічного обладнання;
- всі використані методи для вирішення задач у проекті мають бути безкоштовними.

					ІК82.190БАК.003 ПЗ	Лист
Зм.	Аркуш	№ докум.	Підп.	Дата		20

Висновки до розділу

У даному розділі було розкрито важливість питання потреби у безкоштовному програмному забезпеченні для системи контролю доступом. Проаналізувавши потреби ринку було прийнято рішення про необхідність підвищення ефективності вирішення задачі керування контрольно пропускним пунктом в напів-автоматичному режимі, зокрема розширення доступності таких систем за рахунок розробки програмного забезпечення. На основі проаналізованих існуючих рішень, а також методів та підходів до задач які потрібно буде вирішити в процесі створення системи контролю доступом для автотранспорту було визначено необхідні кроки в вирішенні поставленої задачі.

					ІК82.190БАК.003 ПЗ	Лист
						21
Зм.	Аркуш	№ докум.	Підп.	Дата		

2 АРХІТЕКТУРНЕ МОДЕЛЮВАННЯ СИСТЕМИ

2.1 Модель розпізнавання області номеру

Для того щоб вибрати модель розпізнавання номерних знаків маємо визначитися з нашими фаворитами. Це будуть каскадні класифікатори та нейронні мережі, так як вони мають найбільшу швидкість роботи та високу точність.

В наш час нейронні мережі стали дуже ефективними та майже повністю витіснили каскадні класифікатори з вживання. Але для нашої задачі дуже важливо мати мінімальні трудовитрати на реалізацію проекту. Тому ми зупинимось на такому рішенні, як каскадний класифікатор для розпізнавання номерних знаків. З офіційної Git-репозиторії OpenCV візьмемо xml файл з розміткою для класифікатору. По суті ми отримаємо вже готові ваги, які були отриманні завдяки навчанню на великій вибірці даних. Це значно спростить реалізацію програмного забезпечення та дозволить зробити систему більш доступною.

Для розпізнавання наш каскадний класифікатор використовує метод Віоли-Джонса, який в своїй основі має примітиви Хаара, вони представляють собою розбивку заданої прямокутної області на набори різних прямокутних підобластей (рис. 2.1).

На етапі виявлення певних ознак в методі Віоли-Джонса об'єкт встановленого розміру рухається по зображенню і для кожної області яку приходить об'єкт розраховується ознака Хаара. Визначення можливої присутності шуканого предмета визначається різницею між фактичним значенням та пороговим, отриманим у результаті навчання. Оскільки самі по собі ознаки Хаара не підходять для навчання чи класифікації, так як їх точність трошки вища ніж у випадкової нормально розподіленої величини, тому у методі Віоли-Джонса примітиви Хаара організовані у каскадний

					ІК82.190БАК.003 ПЗ	Лист
Зм.	Аркуш	№ докум.	Підп.	Дата		22

класифікатор. Головною перевагою ознак Хаара є найбільша швидкість в порівнянні з іншими ознаками.

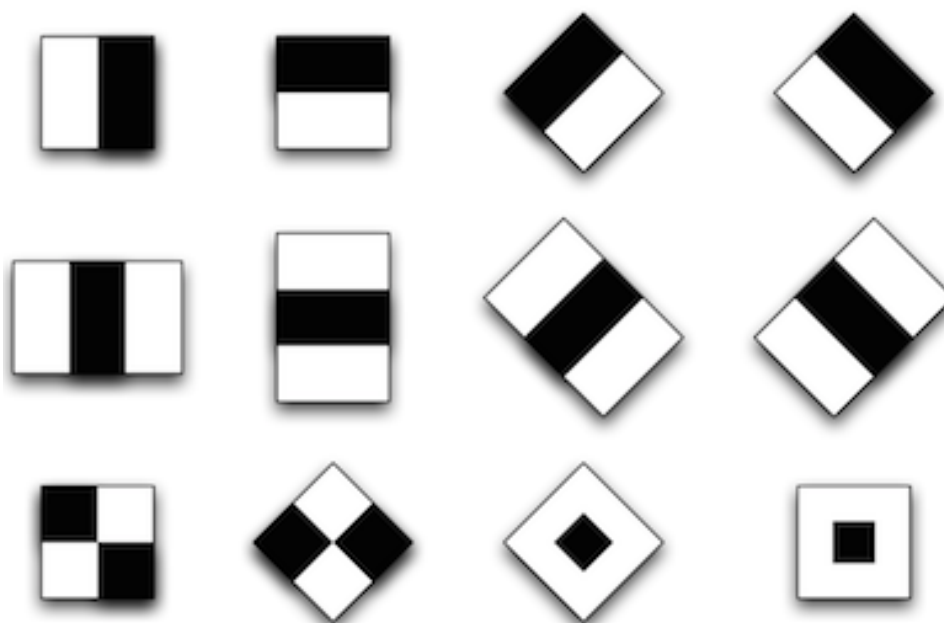


Рис. 2.1 Ознаки Хаара [2]

В першій версії алгоритму Віюлі-Джонса використовувались тільки примітиви Хаара без поворотів, а для визначення значення ознаки сума яскравостей пікселів однієї підобласті віднімали від суми яскравостей іншої підобласті. З розвитком методи були запропоновані ознаки з кутом нахилу 45 градусів, та несиметричних конфігурацій (рис. 2.2).

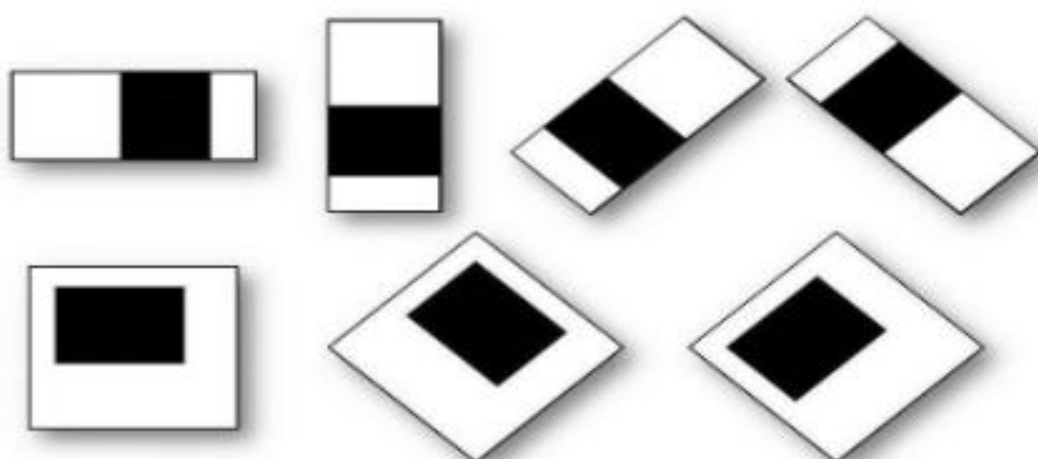


Рис. 2.2 Додаткові ознаки Хаара [2]

Також замість того щоб рахувати звичайну різницю, було запропоновано присвоювати кожній підобласті певні ваги та вираховувати значення ознаки як зважену суму пікселів різних підобластей. Формула наведена на рис. 2.3.

$$feature = \sum_{i \in I=1, \dots, N} w_i \cdot RectSum(r_i)$$

Рис. 2.3 Формула розрахунку ознаки [2]

Для визначення належності до певного класу в кожному каскаді знаходиться сума значень слабких класифікаторів для цього каскаду. Кожен слабкий класифікатор повертає два значення в залежності від того більше, або менше заданого порогового значення, що належить до цього класифікатора. Після чого сума значень слабких класифікаторів порівнюється з порогом каскаду й отримується рішення чи даний каскад знайшов предмет. Якщо поглянути на перший каскад нашого класифікатора (рис. 2.4), то на перший погляд здається, що це просто набір незрозумілих цифр та інформації. Але на ділі все просто:

- weakClassifiers — набір слабких класифікаторів, на основі яких приймаються рішення про присутність предмету на зображенні,
- internalNodes та leafValues — це параметри конкретного слабого класифікатора.

Розшифровка internalNodes: перші два значення в нашому випадку не використовуються, третє – номер ознаки в загальній таблиці ознак (вона розташовується далі в XML файлі під тегом features), четверте – граничне значення слабого класифікатора. Так як у нас використовується класифікатор, заснований на однорівневих вирішальних деревах (decision stump), то якщо значення ознаки Хаара менше за поріг слабого

класифікатора (четверте значення в `internalNodes`), вибирається перше значення `leafValues`, якщо більше – друге.

```
<maxWeakCount>6</maxWeakCount>
<stageThreshold>-1.3110191822052002e+000</stageThreshold>
<weakClassifiers>
  <_>
    <internalNodes>
      0 -1 193 1.0079263709485531e-002</internalNodes>
    <leafValues>
      -8.1339186429977417e-001 5.0277775526046753e-001</leafValues></_>
  <_>
    <internalNodes>
      0 -1 94 -2.2060684859752655e-002</internalNodes>
    <leafValues>
      7.9418992996215820e-001 -5.0896102190017700e-001</leafValues></_>
  <_>
    <internalNodes>
      0 -1 18 -4.8777908086776733e-002</internalNodes>
    <leafValues>
      7.1656656265258789e-001 -4.1640335321426392e-001</leafValues></_>
  <_>
    <internalNodes>
      0 -1 35 1.0387318208813667e-002</internalNodes>
    <leafValues>
      3.7618312239646912e-001 -8.5504144430160522e-001</leafValues></_>
  <_>
    <internalNodes>
      0 -1 191 -9.4083719886839390e-004</internalNodes>
    <leafValues>
      4.2658549547195435e-001 -5.7729166746139526e-001</leafValues></_>
  <_>
    <internalNodes>
      0 -1 48 -8.2391249015927315e-003</internalNodes>
    <leafValues>
      8.2346975803375244e-001 -3.7503159046173096e-001</leafValues></_></weakClassifiers>
```

Рис. 2.4 Перший каскад класифікатору [2]

Всі ці ознаки в певній мірі являються звичайними детекторами границь. На основі цього будується рішення з приводу того чи був розпізнаний предмет на зображенні.

Також другим по важливості моментом в методі Віоли-Джонса є використання каскадної моделі або виродженого дерева прийняття рішень: у кожному вузлі дерева за допомогою каскаду приймається рішення, чи може на зображенні містяться об'єкт чи ні. Якщо об'єкт не міститься, то алгоритм закінчує свою роботу, якщо він може утримуватися, ми переходимо до наступного вузла. Навчання побудовано таким чином, щоб на початкових рівнях з найменшими витратами відкидати більшу частину вікон, у яких не може бути об'єкт. У разі розпізнавання автомобільних номерів – перший рівень містить шість слабких класифікаторів, при тому що останні рівні можуть бути до 15-ти класифікаторів. Для того щоб краще зрозуміти принцип роботи наведемо декілька рівнів розпізнавання номерних знаків на рис 2.5. Більш насичений колір відповідає більшій вазі вікна відносно рівня класифікації.



Рис. 2.5 Відображення роботи класифікатору
на основі методу Віоли-Джонса [2]

2.2 Модель зчитування тексту з номерного знаку

Перед тим як перейти до зчитування тексту з зображення, його треба попередньо обробити. В нашому випадку потрібні два методи попередньої обробки:

- De-skew — вирівнювання тексту належним чином для застосування повної горизонтальності або вертикальності.

- Бінаризація — перетворення зображення з кольорового в чорно-біле задля того, щоб відокремити текст від фону. Ця операція є необхідною, оскільки більшість алгоритмів працюють лише з бінарними зображеннями, а якість бінаризації напряду впливає на якість розпізнавання символів, в ідеалі використовувати певний набір параметрів бінаризації для кожного типу вхідного зображення.

Для підвищення точності розпізнавання можна обмежити можливий перелік символів, які можуть бути на зображенні. В нашому випадку підходить обмеження розпізнавання цифр та букв англійського алфавіту.

В нашому випадку буде доцільно використати пакет EasyOCR, щоб легко виконувати оптичне розпізнавання символів з номерного знаку. EasyOCR є найпростішим способом застосування оптичного розпізнавання символів, що в умах поставленої задачі є головним:

- Пакет EasyOCR можна встановити за допомогою одної команди
- Залежності від пакета EasyOCR мінімальні, що дозволяє легко налаштувати середовище розробки OCR
- Після встановлення EasyOCR потрібен тільки одна команда для того щоб додати пакет у проект

Пакет EasyOCR створюється та підтримується Jaided AI, компанією, яка спеціалізується на послугах оптичного розпізнавання символів.

EasyOCR реалізовано за допомогою Python і бібліотеки PyTorch. Якщо у вас є графічний процесор із підтримкою CUDA, базова бібліотека

					ІК82.190БАК.003 ПЗ	Лист
Зм.	Аркуш	№ докум.	Підп.	Дата		27

глибокого навчання PyTorch може надзвичайно прискорити виявлення тексту та швидкість OCR.

На даний момент EasyOCR може розпізнавати текст більш як 58 мовами, включаючи англійську, німецьку, російську тощо! Підтримувачі EasyOCR планують додати додаткові мови в майбутньому.

Після того як EasyOCR був доданий до проекту ми маємо вказати мову яку хочемо зчитувати. EasyOCR може читати кількома мовами одночасно, але вони повинні бути сумісні одна з одною. Англійська мова сумісна з усіма мовами. Мови, які мають спільні символи (наприклад, латинський шрифт) одна з одною, сумісні.

2.3 Схема розробки програмного забезпечення

Визначившись з інструментами для обробки зображення, розпізнавання номерного знаку та способом зчитування тексту ми маємо з'єднати між собою весь необхідний функціонал за допомогою програмних засобів. Після того як буде розроблений робочий прототип, потрібно буде створити простий інтерфейс користувача за допомогою десктопного додатку та додати остаточну логіку роботи програмного забезпечення з використанням бази даних.

Ледве не найважливішою частиною розробки кожної системи є проектування архітектури, тому що архітектура має бути ефективною, гнучкою, гарно масштабованою, здатною легко розширюватися без подальших радикальних змін у основі структури системи і тому якісно протестованою.

Отже, при виборі інструментів та технологій для підтримки компонентів системи потрібно дотримуватися, перш за все, принципів масштабованості та уникнення конфліктів між компонентами архітектури.

Тобто розробку нашого програмного забезпечення можна поділити на такі компоненти: підсистема розпізнавання номерного знаку, підсистема

зчитування символів з номерного знаку, підсистема взаємодії з базою даних, модуль інтерфейсу для взаємодії з користувачем. Це відображає функціональна схема системи на рис. 2.6 та додатку ІК82.190БАК.003 ДЗ.

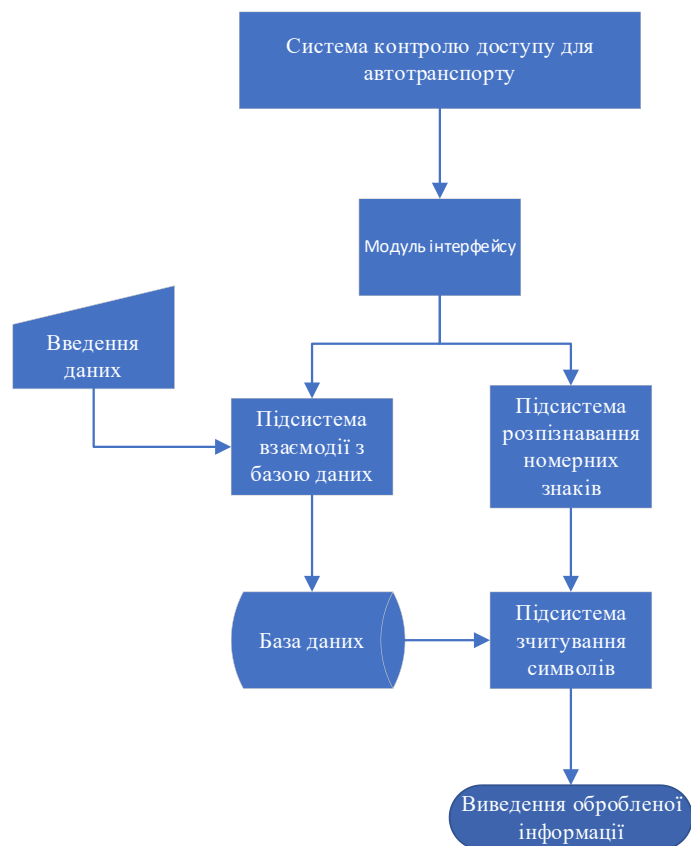


Рис. 2.6 Функціональна схема системи

Розглянемо роботу кожної з підсистем окремо, а також їх взаємодію між собою:

- Користувач взаємодіє з програмою через графічний інтерфейс. Модуль інтерфейсу має відображати всю необхідну інформацію (картинку, номерний знак, дозвіл на проїзд). А також надавати користувачу можливість взаємодіяти з базою даних.

- Підсистема взаємодії з базою даних має на собі обов'язки коригувати дані в базі, відображати їх для користувача, а також передавати в підсистему зчитування символів.

- Підсистема розпізнавання отримує на вхід зображення з камери спостереження, попередньо обробляє його для більш точної роботи

класифікатору, та отримує за допомогою нього координати розпізнаного номерного знаку. Для того щоб розуміти яку область класифікатор вважає областю з номером ми відображаємо її на вхідному зображенні виділяючи цю область рамкою. Також ці координати будуть передані в підсистему зчитування символів, для подальшої роботи системи.

– Підсистема зчитування символів маючи на вході область з номерним знаком, за допомогою інструменту EasyOCR, розпізнає символи в указаній області. Проводить перевірку в базі даних на право доступу, та передає інформацію користувачу, а також при наявності права на проїзд подає сигнал на автоматичний шлагбаум.

Також була створена схема алгоритму роботи програми, її наведено в додатку ІК82.190БАК.003 Д2 та її частина на рис. 2.7.

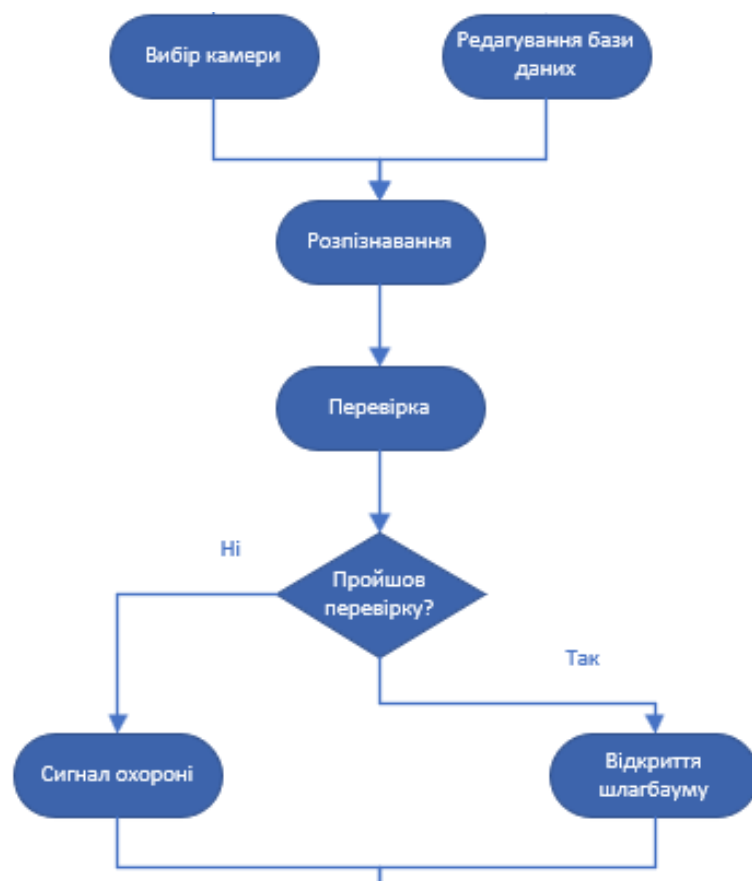


Рис. 2.7 Основна частина алгоритму програми

Та для кращого розуміння технологічних процесів в системі наведена діаграма активності на рис. 2.8 (частково), а також у додатку ІК82.190БАК.003 Д4.

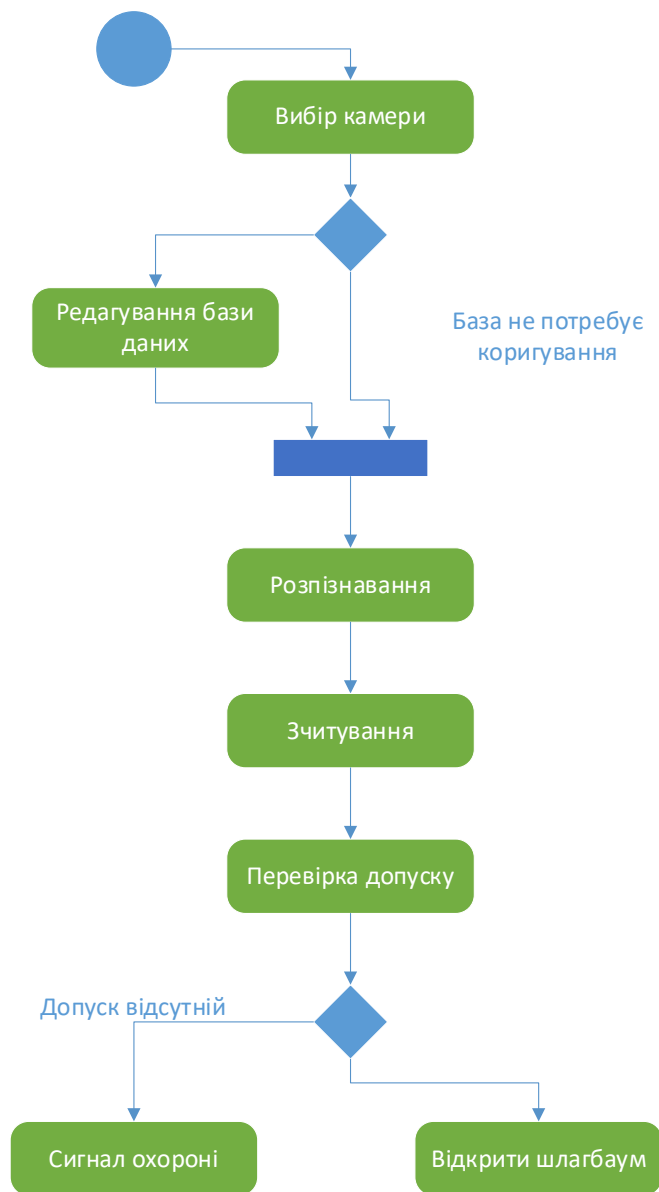


Рис. 2.8 Часткове відображення діаграми активності

Висновки до розділу

У даному розділі було визначено модель для розпізнавання області з номерним знаком та моделі зчитування символів з нього. Також було визначено схему розробки програмного забезпечення. Розробка ПЗ розділена на окремі компоненти та визначено порядок взаємодії між ними.

З ТЕХНОЛОГІЇ ДЛЯ СТВОРЕННЯ ДОДАТКУ. РОЗРОБКА ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Вибір мови програмування

В 2022 році людям при створенні будь яких програмних засобів доступні різні ресурси. Такі як:

- мови програмування;
- середовища розробки;
- бібліотеки та фреймворки;
- та багато іншого.

В цьому підрозділі треба провести глибинний аналіз ринку, для розуміння актуальності одних або інших засобів та дізнатись їх особливості. Також слід врахувати власні навички та знання. Згодом ми матимемо змогу вибрати конкретні засоби для початку розробки власного проекту.

Найголовніше це вибір мови програмування. Для того щоб отримати певну статистику про мови програмування можемо скористатись сервісом Google Trends.

Проведемо аналіз популярності різних мов програмування за період з травня 2021 року по травень 2022 року. За статистикою взятою з ресурсу Google Trends маємо наступну картину (рис. 3.1). З цього можемо зробити висновок що найбільшу популярність мають наступні мови програмування: Python, Java, JavaScript. Тому проведемо ретельний аналіз наведених мов програмування.

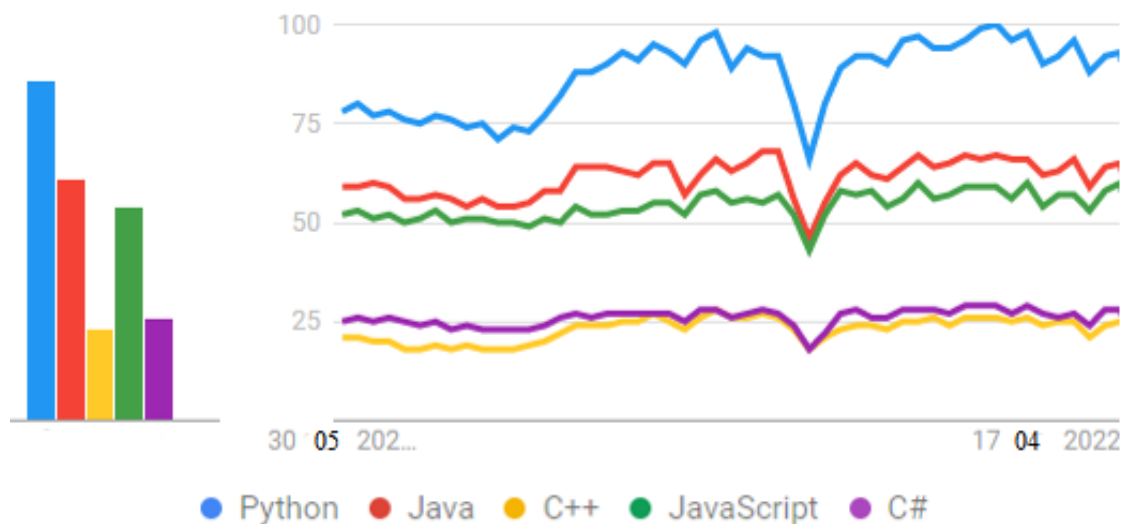


Рис. 3.1 Популярність мов програмування за останній рік

3.1.1 JavaScript

JavaScript - мова програмування, що широко використовується і де-факто є стандартом в браузерах та різних додатках. Одна з головних задач JavaScript є забезпечення інтерактивності сторінки. В той час коли на сторінці відбувається щось, окрім статичного відображення, а саме: анімація, відображення вмісту що оновлюється, то швидше за все в цьому приймає участь JavaScript.

Інтерфейси прикладного програмування (API) надають додаткові можливості для використання JavaScript. API — це готові набори блоків коду, які дозволяють розробнику реалізовувати програми, які в іншому випадку було б важко реалізувати. Це можна порівняти з дерев'яним конструктором. Набагато простіше зібрати певну модель конструктора з заготовлених деталей виробником, ніж починати збірку з суцільних дерев'яних листів.

Мовою програмування JavaScript можуть оволодіти навіть новачки, з необхідних навичок маємо: знання об'єктно орієнтованого програмування

та теоретична підготовка. Переваги та недоліки даної мови програмування відображає таблиця 3.1.

Таблиця 3.1. Переваги та недоліки JavaScript

Переваги	Недоліки
Швидка та проста	Немає типізації
Сумісна та універсальна	Безпека на стороні клієнта
Різноманітні інтерфейси	Зупинка рендеру під час помилки

3.1.2 Java

На 2022 рік Java є однією з найпоширеніших та найпопулярніших мов програмування, котра була створена організацією Oracle і в результаті стала її об'єктно-орієнтованою структурою і хорошим прикладом, за те що мала властивість під назвою WORA (write once, run anywhere). Даний лозунг означає, що будь-яка програма, написана лише один раз, може працювати будь-де. Тобто, програма може бути створена на будь-якому пристрої, скомпільованому у стандартний байт-код, і розраховується, що вона буде запускатися на будь-якому пристрої, забезпеченим віртуальною машиною (JVM). З огляду на це, Java підтримує: Mac, Windows, Android, IOS і т.д. Як наслідок згадана мова програмування визнана як крос-платформна. На сьогоднішній день в світі наявні близько 3 мільярди пристроїв, на котрих запущено програми, створені на основі Java.

Вона використовується при створенні веб додатків, десктопних програм, для обробки великих даних та для програмування бізнес логіки на стороні серверів великих корпорацій, таких як: Google, Amazon, Twitter і YouTube. За допомогою нових фреймворкам, як: Spring, Hibernate та Struts

— Java не збувається свого світового визнання та лідируючих позицій в численних рейтингах. До того ж, програмісти Java формують чимале Інтернет-співтовариство та підтримують одне одного, допомагають у вирішенні професійних питань, спілкуючись на безліч форумах.

Мовою програмування Java можуть оволодіти досвідчені програмісти, з необхідних навичок маємо: знання об'єктно орієнтованого програмування та теоретична підготовка. Переваги та недоліки даної мови програмування відображає таблиця 3.2.

Таблиця 3.2. Переваги та недоліки Java

Переваги	Недоліки
Крос-платформенна	Мова вищого рівня
Автоматичне виділення пам'яті та збір сміття	Не підходить для створення хмарних додатків
Багатопоточна та захищена	

3.1.3 Python

Python на 2022 рік є беззаперечним лідером поміж решти мов програмування. Це зумовлено простотою у період навчання для новачків, завдяки хорошій, легкій читабельності коду. Поза тим, він не вимагає ніякої оплати та його вихідний код міститься у вільному доступі на просторах всесвітньої павутини. Дана мова програмування має розгалужені модулі підтримки, постійно оновлюється та підтримується за допомогою розробників та власних користувачів. Без труднощів інтегрується з веб сервісами та іншими структурами даних. У теперішній час – це найліпший вибір для програм машинного навчання.

Python використовується для підтримки та створення різних 2D зображень та 3D-анімацій, відомих відеоігор, наукових та обчислювальних програм, крім того, і для популярних сайтів, таких як: Blender, Inkscape та Autodesk, Civilization IV, FreeCAD та Abacus, YouTube, Pinterest та Instagram і т.п.

Мовою програмування Python можуть оволодіти навіть новачки, з необхідних навичок маємо: теоретичні вміння та абстрактне мислення. Переваги та недоліки даної мови програмування відображає таблиця 3.3.

Таблиця 3.3. Переваги та недоліки Python

Переваги	Недоліки
Гнучка, легко вивчити	Абстрагує занадто багато важливих базових понять
Інтуїтивно зрозуміла	Не підходить для створення мобільних додатків
Відмінні офіційні навчальні посібники та документація	
Має структуру скрипта, непотрібно компілювати	

Все ж таки була обрана мова програмування Python, беручи до уваги три аргументи:

- через свої значні переваги, які складають досить великий список;
- через те що обрані нами методи розпізнавання та зчитування номерного знаку мають легку реалізацію за допомогою Python;

— через власний досвід в використуванні даної мови та власні навички що допоможуть створити програмне забезпечення відповідно темі проекту.

3.2 Вибір середовища розробки. PyCharm чи Sublime Text

Sublime Text (рис. 3.2) — це комерційний текстовий редактор для програміста, який розробляється з 2008 року. Його основні переваги — швидкість, потужні функції редагування тексту коду (мультикурсор), підтримка між платформою, налаштування та екосистема плагінів. Наразі Sublime Text версія 3 знаходиться в бета-версії. Хоча розробка сповільнилася в один момент, оскільки Sublime Text був переважно одним шоу, нові збірки Sublime Text виходять тепер регулярно. Sublime Text коштує 70 доларів США. Якщо ви не придбаєте ліцензію, ви отримаєте сповіщення в нудному діалоговому вікні.

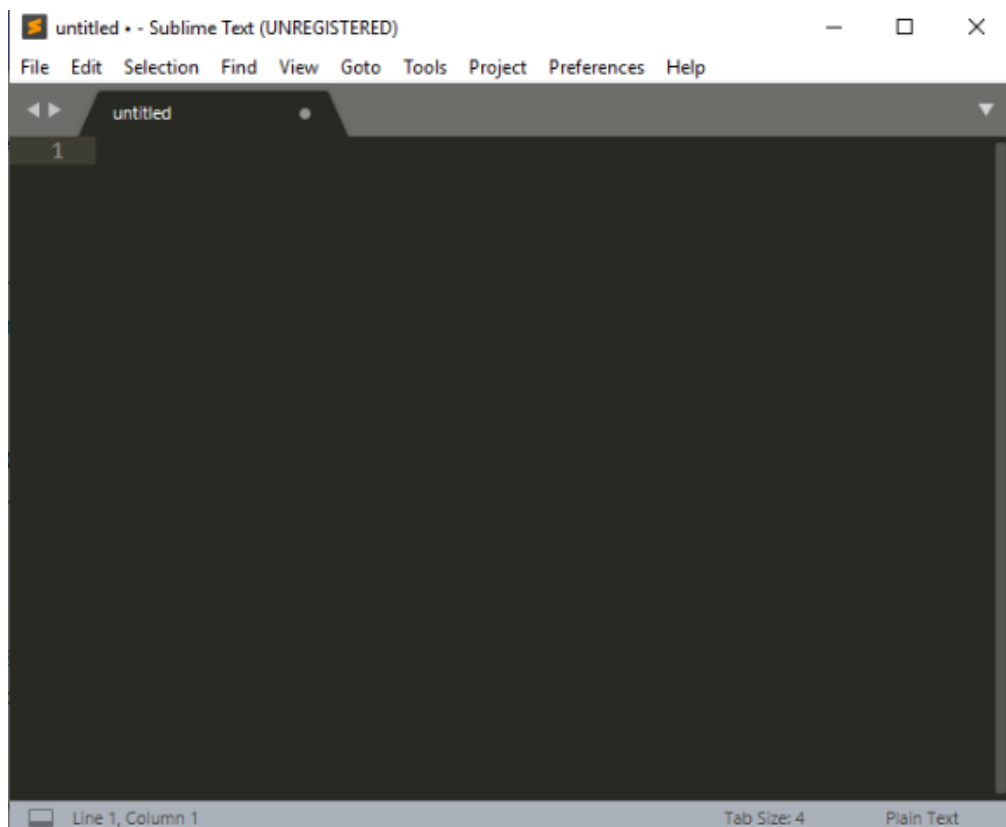


Рис. 3.2 Інтерфейс Sublime Text

PyCharm (рис. 3.3) — це дитина сімейства редакторів JetBrains IntelliJ IDEA. Перший PyCharm був випущений у 2010 році, але кодова база IDE сходить до IntelliJ IDEA, яка була випущена ще в 2001 році – я пам’ятаю, як розробляв Java на IntelliJ у 2004 році. PyCharm розроблено чеською компанією JetBrains, у якій працює понад 400 співробітників. PyCharm поділяє більшість функцій з іншими IDE сімейства IDEA, що означає, що він має надійну підтримку HTML, JavaScript і CSS. Ліцензія PyCharm коштує 199 євро / рік (професійна), 99 євро / рік (індивідуальна), а також є безкоштовне видання для спільноти. Видання для спільноти на 100% є відкритим вихідним кодом.

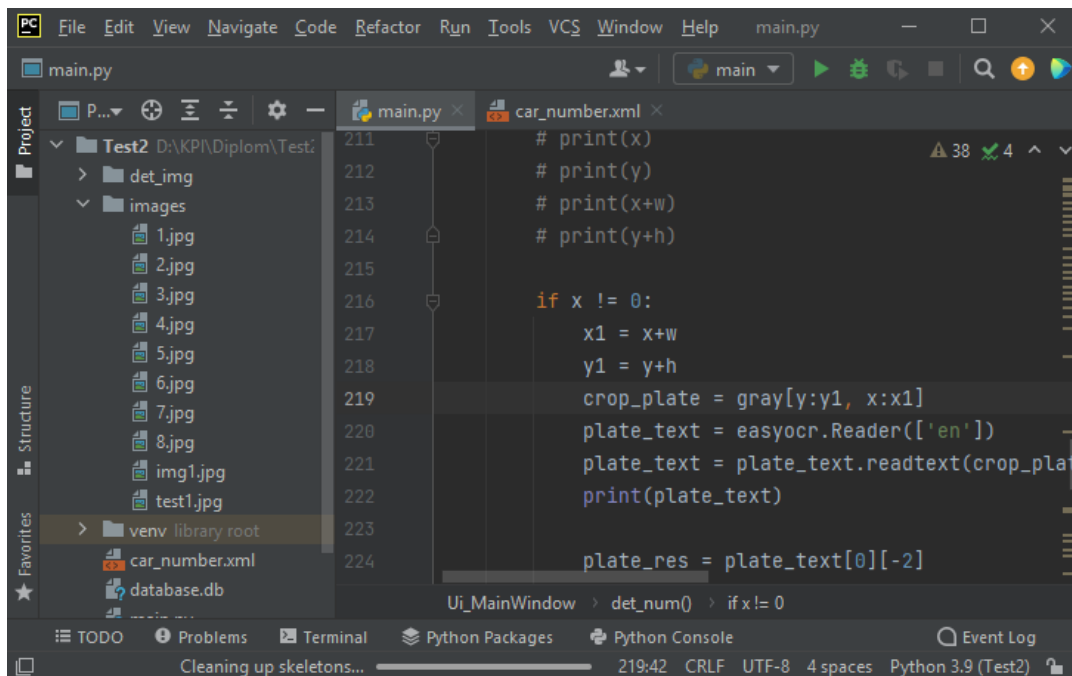


Рис. 3.3 Інтерфейс PyCharm

Sublime Text і PyCharm мають інтегрований менеджер плагінів. Sublime Text Package Control не вбудований, що робить початкове впровадження більш клопоту. З іншого боку інсталятор плагінів PyCharm є більш громіздким у використанні, але для початкового користувача є простішим.

Обидва текстові редактори хороші для редагування Python і мають, наприклад, інструкції щодо відступів і швидке перемикання параметрів м'якого перенесення тексту.

PyCharm має історію з 2001 року, він має безліч функцій із коробки. Він дуже відшліфований і дуже добре виконує більшість функцій – адже продаж IDE є основним бізнесом для JetBrains – наприклад, порівнюючи IBM Eclipse, тоді як основним бізнесом IBM є продаж послуг IBM. З PyCharm вам потрібно витратити небагато часу на налаштування середовища програмування або пошук плагінів для ваших основних потреб розробки (Python, JavaScript, HTML, CSS).

PyCharm постачається з інтегрованим налагоджувачем. Ви можете двічі клацнути, щоб встановити точки зупину у своєму редакторі, а потім запустити програму, щоб зупинитися на лінії. Але ви все одно не втрачаєте можливість перекидатися в інтерактивну оболонку IPython, коли потрапляєте до точки зупину.

Автозаповнення, автоімпорт та інші інструменти розробки коду та рефакторингу краще працюють у PyCharm. З Sublime Text вам потрібно пограти з великою кількістю плагінів, щоб отримати гідне автозаповнення. Плагіни Sublime Text мають власні, несумісні налаштування та потребують великої кількості ручної установки пакетів (pip install flake8 тощо). У PyCharm я просто натиснув alt+enter на відсутній символ, і він був доданий як імпорт на початку файлу. Хоча я не міг змінити його, щоб відформатувати імпорт, як я хочу – один оператор імпорту на один рядок.

PyCharm виконує фонову перевірку орфографії написаного тексту та рядків документів Python. Це дуже зручно, якщо ви хочете писати високоякісне програмне забезпечення зі змістовними коментарями та описами API.

PyCharm має більш надійну підтримку інтегрованого контролю версій (Git, SVN). Хоча Sublime Text має плагіни для цього, API плагіна

Sublime Text пропонує лише дуже обмежену взаємодію з інтерфейсом користувача, і ви не можете, наприклад, розфарбовувати файли в провіднику проекту на основі їх статусу редагування.

PyCharm має режим енергозбереження. Він вимикає фонові завдання, такі як код Intel, які є осушенням ЦП для великих проектів. Це робить цифрове кочування набагато веселішим, коли ви боретеся за єдину доступну розетку живлення в хостелі на віддаленому острові.

PyCharm має інтегрований термінал і вікно запуску, хоча він має недоліки і, здається, не поводить себе як справжній термінал. Також PyCharm має інтегрований модуль тестування.

З усього вище описаного напрошується висновок, що для моєї задачі та рівня підготовки більше підійде PyCharm. Так як він має нижчий поріг знань для входу, має різноманітний вбудований функціонал та зможе виконати всі поставлені задачі.

3.3 База даних. SQLite

База даних — це спеціально організований набір інформації або даних, що часто зберігаються в електронному вигляді на сервері чи комп'ютері. Керується база даних - системою управління базами даних (СУБД).

Системою бази даних називається - інформація сукупно з СУБД, плюс пов'язані з ними програми. У найпоширеніших типах новітніх баз даних, дані лежать у вигляді рядків і стовпців, що сформовані в таблицю. Дана інформація піддається управлінню, редагуванню, змінюванню, оновленню, контролю та організуванню. У найпопулярніших базах даних для запису і запитів інформації застосовується мова структурованих запитів (SQL), що є мовою програмування, яка використовується в великій кількості реляційних баз даних для запитів, обробки і визначення даних, плюс контролю доступу.

					ІК82.190БАК.003 ПЗ	Лист
Зм.	Аркуш	№ докум.	Підп.	Дата		40

Існує велика кількість різноманітних видів баз даних. Вибір найоптимальнішою бази даних для певної організації залежить від наміру взаємодії з конкретними даними.

– Реляційні бази даних. Реляційні бази даних отримали свою популярність ще в 80-х роках. Дані в реляційній базі структуровані у вигляді таблиць, що формуються за допомогою рядків і стовпців. Реляційна СУБД забезпечує швидкий і ефективний доступ до структурованих даних.

– Об'єктно-орієнтовані бази даних. Інформація в об'єктно-орієнтованій базі даних зберігається у вигляді об'єкта, як в об'єктно-орієнтованому програмуванні.

– Розподілені бази даних. Дана база даних формується за допомогою декількох частин, розміщених на інших серверах. До того ж, вона може зберігатися на декількох комп'ютерах.

– Сховища даних. Будучи централізованим репозиторієм для даних, сховище представляє собою вид бази даних, що спеціально призначена для прискореного виконання запитів і аналізу.

– Бази даних NoSQL. База даних NoSQL, або, по інакшому, нереляційних база даних, надає змогу зберігати та обробляти неструктуровані дані. Популярність зазначених баз даних збільшується в міру розповсюдження і ускладнення веб-додатків.

– Графові бази даних. Зазначені бази даних, у вигляді графа зберігають дані в контексті сутностей і зв'язків між сутностями.

– Бази даних OLTP. База даних, що слугує для вчинення бізнес-транзакцій, що виконуються значною кількістю користувачів.

– Бази даних з відкритим вихідним кодом. Відповідні бази даних мають відкритий вихідний код і об'єднують у собі засоби управління SQL та NoSQL.

– Хмарні бази даних. База даних у хмарі являє собою набір структурованих або неструктурованих даних, що знаходяться на приватній,

публічній або гібридній платформі хмарних обчислень. Наявно два типи моделей хмарних баз даних: традиційна база даних і база даних як послуга. У моделі послуги обслуговування та адміністративні задачі здійснюються постачальником хмарних послуг.

– Багатомодельна бази даних. Ця база даних поєднує різні типи моделей баз даних в одну інтегровану серверну СУБД, що означає, її можливість вміщення різних типів даних.

– Документальні бази даних / JSON. Бази даних, що призначені для зберігання, вилучення та обробки документальної інформації та надають новітній спосіб збереження даних в форматі JSON, на відміну звичайного способу, в форматі рядків і стовпців.

– Автономні бази даних. Зазначені бази даних являють собою - сучасні і революційні хмарні бази даних, що користуються машинним навчання для автоматизації налаштування, захисту, резервного копіювання, оновлення та безліч інших стандартних можливостей обслуговування, переважно виконуваних адміністраторами баз даних.

При розробці додатків, нерідко з'являється необхідність в збереженні деяких налаштувань або тимчасових даних. Здебільшого, для виконання даної мети, використовуються файли або бази даних, про що було зауважено вище. З нашими потребами буде доцільно скористатись SQLite.

SQLite - компактна СУБД, що вбудовується. Вихідний код бібліотеки передано на громадські надбання. 2005 року проект отримав нагороду Google-O'Reilly Open Source Awards.

Слово «вбудований» (embedded) означає, що SQLite не використовує парадигми клієнт-сервер, тобто двигун SQLite не є окремо працюючим процесом, з яким взаємодіє програма, а являє собою бібліотеку, з якою програма компонується, і двигун стає складовою програми. Таким чином, як протокол обміну використовуються виклики функцій (API)

					ІК82.190БАК.003 ПЗ	Лист
Зм.	Аркуш	№ докум.	Підп.	Дата		42

бібліотеки SQLite. Даний підхід мінімізує накладні витрати, час відгуку та спрощує програму. SQLite зберігає всю базу даних (включно з визначеннями, таблицями, індексами та даними) у єдиному стандартному файлі, на тому комп'ютері, на якому виконується програма. Простота реалізації досягається рахунок того, що перед початком виконання транзакції запису весь файл, що зберігає базу даних, блокується; ACID-функції досягаються у тому числі створення створення файлу журналу.

Декілька процесів або потоків здатні одночасно без будь-яких проблем читати дані з однієї бази. Запис до бази можна здійснити лише у тому випадку, якщо жодних інших запитів на даний момент не обслуговується; інакше спроба запису закінчується невдачею, й у програму повертається код помилки. Іншим варіантом розвитку подій є автоматичне повторення спроб запису протягом заданого інтервалу часу.

3.4 Короткий опис створення ПЗ

Для створення нашого додатку, який буде задовільняти наші потреби та мати всі необхідні функції, застосуємо стек технологій який був описаний вище. Нам потрібно створити даних, написати бекенд та фронтенд для нашої програми. Фронтенд буде включати в себе графічний інтерфейс користувача, роботу кнопок та відображення інформації.

Для початку роботи ми маємо додати до проекту всі необхідні бібліотеки (рис. 3.4). На даному рисунку ми можемо побачити імпорт бібліотек для бекенду (1-2 рядки), бази даних(4 рядок) та фронтенду(6-8 рядки).

```
1 import cv2
2 import easyocr
3
4 import sqlite3
5
6 from PyQt5 import QtCore, QtWidgets
7 from PyQt5.QtWidgets import QFileDialog
8 from PyQt5.QtGui import QPixmap
```

Рис. 3.4 Додавання бібліотек

					ІК82.190БАК.003 ПЗ	Лист
Зм.	Аркуш	№ докум.	Підп.	Дата		43

За допомогою бібліотеки cv2 й навченого каскадного класифікатору було реалізовано (рис. 3.5) першу важливу функцію – розпізнавання області номерного знаку.

```
numbers = cv2.CascadeClassifier('car_number.xml')
results = numbers.detectMultiScale(gray,
                                   scaleFactor=1.1,
                                   minNeighbors=5)
```

Рис. 3.5 Команди загрузки та роботи класифікатору

Коли класифікатор виконав свою роботу та повернув 4 координати, ми можемо перейти до графічного відображення результатів. На рис. 3.6 відображено проміжкові результати роботи підсистеми.



Рис. 3.6 Результат розпізнавання області номерного знаку

Після чого за допомогою бібліотеки `easyocr` було реалізовано (рис. 3.7) зчитування символів з області номерного знаку використавши наступні рядки коду:

```
plate_text = easyocr.Reader(['en'])  
plate_text = plate_text.readtext(crop_plate)
```

Рис. 3.7 Рядки коду задля зчитування символів з номерного знаку

Якщо вивести змінну «`plate_text`» у консоль (рис. 3.8), то побачимо наступне:

```
[([[16, 10], [118, 10], [118, 38], [16, 38]], 'АН5840 СМ', 0.6451754733481558)]
```

Рис. 3.8 Результат роботи функціоналу зчитування символів

Ця функція повертає багато різних значень, та в нашому випадку нас цікавить тільки зчитані символи. Як бачимо вони потребують деякої пост обробки, треба прибрати всі знайденні пробіли.

Після реалізації двох основних функцій програми маємо створити базу даних (таблицю) для збереження номерів автомобілів. Нам буде достатньо мати одну примітивну таблицю яка буде мати три поля:

- ID
- Номерний знак
- Повноваження для проїзду

Маючи таку легку в реалізації табличку можемо порівнювати розпізнаний номерний знак з тими що є в базі даних, надавати різні статуси проїзду, це може бути одноразовий проїзд або постійний, що дуже важливо для нормального функціонування системи, так як не потрібно буде перевіряти номери транспортних засобів що мають дозвіл на одноразовий в'їзд - кур'єрів, таксистів, гостей та інших в ручному режимі.

Так як було реалізовано всю логіку роботи програми залишилось написати графічний інтерфейс який матиме всі необхідні функції для повноцінної роботи системи.

Для реалізації інтерфейсу я скористався утилітою Qt designer (рис.3.8). В ній за допомогою простого та зрозумілого інтерфейсу вдалось створити макет нашого додатку.

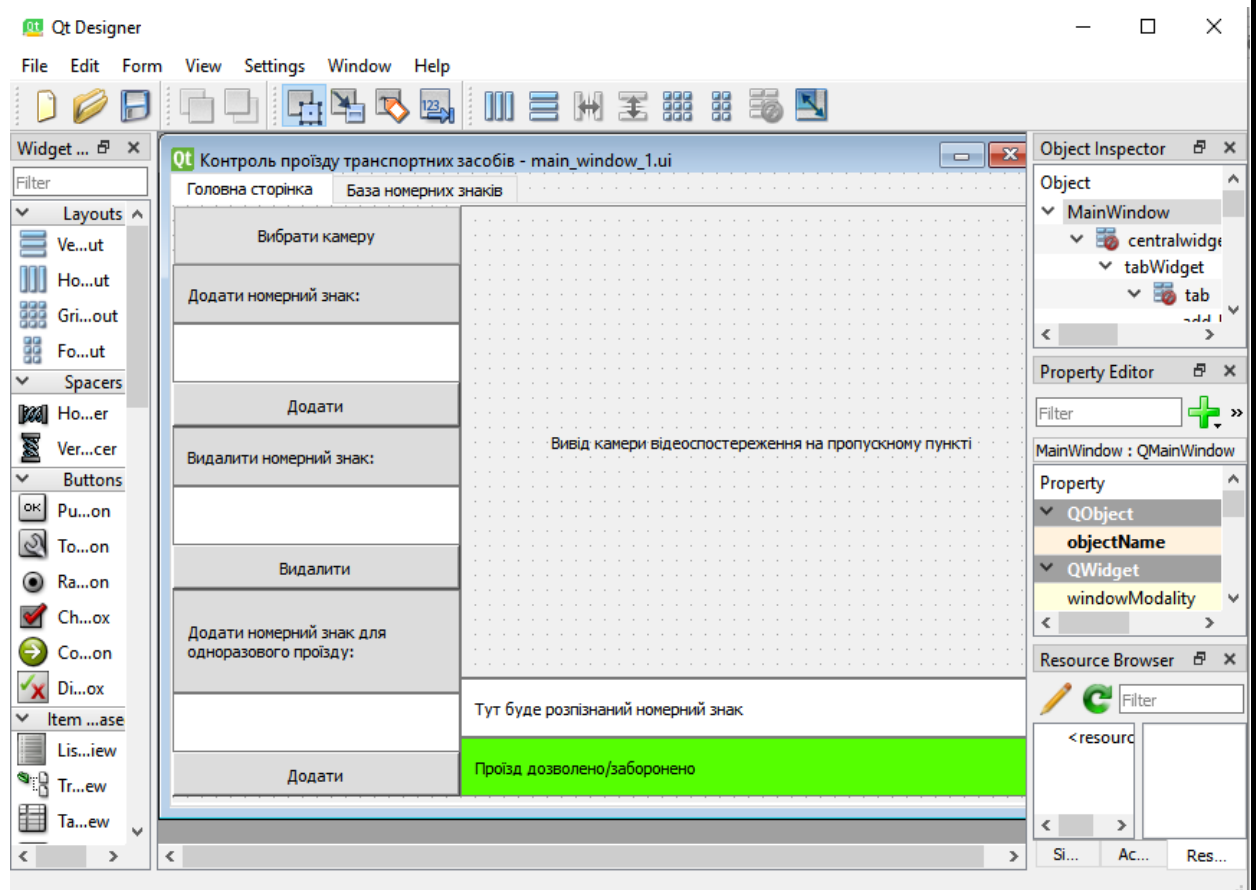


Рис. 3.8 Відображення макету додатку в Qt designer

3.5 Тестування програмного забезпечення

Оскільки маємо створене програмне забезпечення, то потрібно перейти до його наповнення даними та тестування. Але перед тим як тестувати сам додаток ми маємо провести тестування підсистеми розпізнавання номерних знаків.

3.5.1 Тестування підсистеми розпізнавання та зчитування номерного знаку

Для цього тестування нам знадобиться велика кількість зображень автомобіля з номерним знаком. Після того як була сформована така збірка, було запущено тестування точності розпізнавання. З вибірки в 100 зображень система правильно розпізнала область номерного знаку на 96 із них. При цьому вірно розпізнати символи номерного знаку вийшло на 90 автомобілях. Отже наша система має точність розпізнавання приблизно 90%. Це досить висока точність розпізнавання для даної системи.

Розглянемо випадки в яких система давала хибний результат:

– На рис. 3.9 не було розпізнано область номерного знаку, це пов’язано з низькою якістю зображення в купі з віддаленим положенням автомобіля. Тому камера має передавати чітке зображення для подальшої обробки.



Тут буде розпізнаний номерний знак

Рис. 3.9 Приклад хибної роботи системи 1

– На рис. 3.10 не було розпізнано область номерного знаку, це пов’язано з положенням камери відносно автомобіля. Камера розташована не на оптимальній висоті, має великий нахил горизонту та автомобіль розташовано надто близько до неї. При правильному положенні камери таких проблем виникати не має.



Рис. 3.10 Приклад хибної роботи системи 2

– На рис. 3.11 не було розпізнано область номерного знаку, це пов’язано з сильним забруднення номерного знаку та області навколо нього. Така проблема особливо актуальна в зимовий період, але відповідно до правил дорожнього руху державний номер має бути завжди чистий та чітко читаємий. Тому тут тільки один шлях вирішення – дотримання державного номеру в чистоті



Рис. 3.11 Приклад хибної роботи системи 3

– На рис. 3.12 було розпізнано область номерного знаку, але було хибно зчитано символи з нього. Це пов'язано з тим що номерний знак має сильні потертості. Номер повинен чітко читатись і не мати таких пошкоджень що система не може ідентифікувати символи з нього. Це також вирішується тільки одним шляхом – заміною номерного знаку на новий відповідно до діючих законів.



ВН3Z35BB

Рис. 3.12 Приклад хибної роботи системи 4

– На рис. 3.13 було розпізнано область номерного знаку, але при створених умовах перекриття одного символу система розпізнала номер не вірно. Умови створені штучно, за допомогою графічного редактору. Але існують випадки коли навмисно, або випадково закривають один чи декілька символів для уникнення фіксації номерного знаку. В такому випадку система буде давати невірний результат.



KA456HP

Рис. 3.13 Приклад хибної роботи системи 5

Як бачимо система дає результати достатньої точності. Та має похибку лише в специфічних випадках. З чого можемо робити висновок, що створена система повністю задовільняє потреби розпізнавання номерних знаків в створених для цього умовах.

3.5.2 Тестування додатку

Оскільки тестування підсистеми розпізнавання номерних знаків дало позитивний результат, можемо перейти до тестування додатку. При запуску програми ми бачимо вікно яке відповідає рис. 3.14.

					ІК82.190БАК.003 ПЗ	Лист
Зм.	Аркуш	№ докум.	Підп.	Дата		51

Контроль проїзду транспортних засобів

Головна сторінка База номерних знаків

Вибрати камеру

Додати номерний знак:

Додати

Видалити номерний знак:

Видалити

Додати номерний знак для одноразового проїзду:

Додати

Вивід камери відеоспостереження на пропускному пункті

Тут буде розпізнаний номерний знак

Проїзд дозволено/заборонено

Рис. 3.14 Початкова сторінка додатку

На цій сторінці ми можемо побачити:

- кнопку за допомогою якої є можливість додати зображення з камери;
- інтерфейс для додання номерних знаків в базу даних;
- інтерфейс для видалення номерних знаків з бази даних;
- інтерфейс для додавання номерного знаку з правом одноразового проїзду;
- область виводу зображення з камери;
- область виводу номерного знаку;
- область відображення прав доступу на територію;
- перехід на другу вкладку з відображенням існуючої інформації в базі даних.

При роботі програми в режимі автоматичного контролю доступу користувач бачить наступне:

- в випадку коли ТЗ має право проїзду (рис. 3.15);

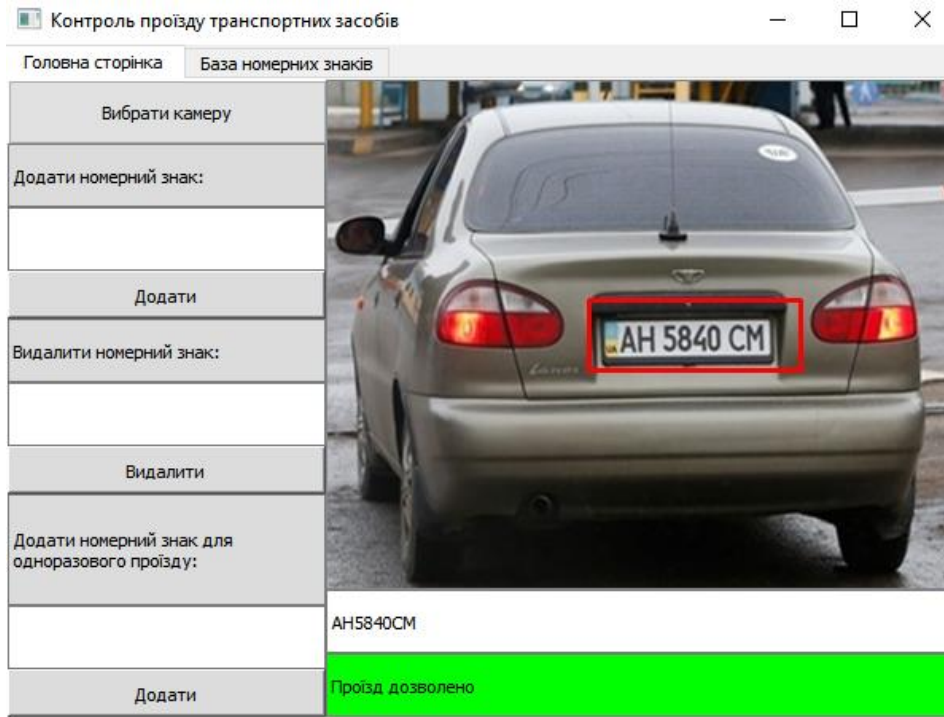


Рис. 3.15 Відображення роботи програми 1

- в випадку коли ТЗ не має права проїзду (рис. 3.16);

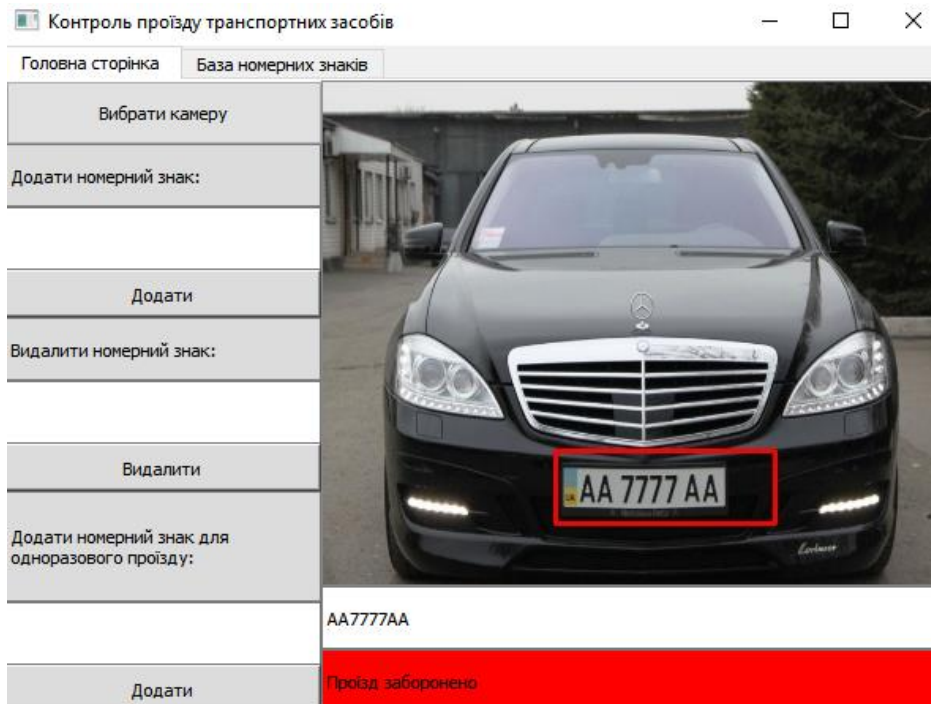


Рис. 3.16 Відображення роботи програми 2

— в випадку коли ТЗ має право одноразового проїзду (рис. 3.17).

The screenshot shows a web application window titled "Контроль проїзду транспортних засобів". It has two tabs: "Головна сторінка" and "База номерних знаків". The left sidebar contains several buttons: "Вибрати камеру", "Додати номерний знак:", "Додати", "Видалити номерний знак:", "Видалити", "Додати номерний знак для одноразового проїзду:", and another "Додати". The main area displays a photo of a green Honda Civic with license plate KA 4556 HP, which is highlighted by a red rectangle. Below the photo, the text "KA4556HP" is shown, followed by a yellow bar indicating "Одноразовий проїзд дозволено".

Рис. 3.17 Відображення роботи програми 3

При переході на вкладку «база номерних знаків» (рис. 3.18) можемо побачити відображення номерних знаків та їх рівень доступу.

The screenshot shows the same application window as Figure 3.17, but now displaying the "База номерних знаків" tab. A table lists four records:

ID	Номерний знак	Право доступу
1	AH5840CM	Постійний
2	KA4556HP	Одноразовий
3	KA2354MP	Постійний
4	BH5437MK	Одноразовий

Below the table is a button labeled "Обновити данні".

Рис. 3.18 Відображення бази номерних знаків

Зм.	Аркуш	№ докум.	Підп.	Дата	ІК82.190БАК.003 ПЗ	Лист 54
-----	-------	----------	-------	------	--------------------	------------

[illegible]

– в випадку коли ТЗ має право одноразового проїзду (рис. 3.17).

The screenshot shows a web application window titled "Контроль проїзду транспортних засобів". It has two tabs: "Головна сторінка" and "База номерних знаків". The "База номерних знаків" tab is active, displaying a form with several input fields and buttons. On the right side of the form, there is a live video feed of a green Honda Civic with a Ukrainian license plate "KA 4556 HP", which is highlighted by a red rectangle. Below the video feed, the text "КА4556HP" is displayed. At the bottom of the form, a yellow banner indicates "Одноразовий проїзд дозволено".

Рис. 3.17 Відображення роботи програми 3

При переході на вкладку «база номерних знаків» (рис. 3.18) можемо побачити відображення номерних знаків та їх рівень доступу.

This screenshot shows the same application window as Figure 3.17, but now displaying a table of license plate data under the "База номерних знаків" tab. The table has three columns: "ID", "Номерний знак", and "Право доступу". It contains four rows of data:

ID	Номерний знак	Право доступу
1	AH5840CM	Постійний
2	KA4556HP	Одноразовий
3	KA2354MP	Постійний
4	BH5437MK	Одноразовий

Below the table, there is a grey button labeled "Обновити данні".

Рис. 3.18 Відображення бази номерних знаків

					ІК82.190БАК.003 ПЗ	Лист
Зм.	Аркуш	№ докум.	Підп.	Дата		54

[illegible][illegible]

Висновки до розділу

В даному розділі було проаналізовано та визначено на основі яких технологій буде створений додаток. Ними стали: мова програмування Python, середовище розробки PyCharm, утиліта для використання БД SQLite. На їх основі було створено програмне забезпечення для роботи системи контролю доступу транспортних засобів, за допомогою технологій які були описані в попередніх розділах. Програмне забезпечення було протестовано на всіх стадіях роботи. Воно пройшло тестування та повністю готове до повноцінної роботи. А також визначено умови в яких система буде давати майже 100 відсотковий результат.

					ІК82.190БАК.003 ПЗ	Лист
						55
Зм.	Аркуш	№ докум.	Підп.	Дата		

ВИСНОВКИ

В ході написання даного дипломного проекту було створено систему контролю доступу для автотранспорту. Проведено аналіз теми та підходів до вирішення питання. Були розглянуті основні проблеми даної тематики та запропоновані шляхи вирішення. Зроблений детальний аналіз найкращих методів вирішення задачі та обрані ті що найкраще підходять в нашому випадку. Було розглянуто вже існуючі рішення та визначено їх основні проблеми: висока ціна програмного забезпечення та обладнання яке потрібне для роботи системи. Це надало поштовх для створення власного безкоштовного програмного забезпечення. Проведено збір даних щодо сучасних мов програмування та відповідного стеку технологій. Це дало змогу створити простий додаток зі зрозумілим та зручним інтерфейсом користувача. Створений додаток дозволяє керувати людині без особливих навиків контрольно-пропускним пунктом. Користувач має змогу слідкувати за роботою системи в автоматичному режимі, а також додавати та видаляти номерні знаки яким дозволено проїзд. Написане програмне забезпечення має потенціал для створення додаткових корисних функцій та можливості покращення оформлення, наприклад перевірка номеру на відповідність даному транспортному засобу тощо.

					ІК82.190БАК.003 ПЗ	Лист
Зм.	Аркуш	№ докум.	Підп.	Дата		56

ПЕРЕЛІК ПОСИЛАНЬ

1. «Розпізнавання номерів від А до 9» (2014). [Електронний ресурс].
Режим доступу: <https://habr.com/ru/company/recognitor/blog/221891/>
2. «Робота каскаду Хаара в OpenCV у картинках: теорія та практика» (2014). [Електронний ресурс]. Режим доступу: <https://habr.com/ru/company/recognitor/blog/228195/>
3. «Розпізнавання номерів. Практичний посібник. Частина 1» (2019). [Електронний ресурс]. Режим доступу: <https://habr.com/ru/post/432444/>
4. «Розпізнавання номерів. Як ми отримали 97% точності для українських номерів. Частина 2» (2019). [Електронний ресурс]. Режим доступу: <https://habr.com/ru/post/439330/>
5. «FineReader, Tesseract та EasyOCR або потрібно терміново змінювати інструмент для OCR» (2021). [Електронний ресурс]. Режим доступу: <https://habr.com/ru/post/573030/>
6. «Розпізнавання автомобільних номерів у деталях» (2014). [Електронний ресурс]. Режим доступу: <https://habr.com/ru/company/recognitor/blog/225913/>
7. «Навчання каскаду Хаара на прикладі пошуку символів автомобільного номера OpenCV» (2015). [Електронний ресурс]. Режим доступу: <https://kostyakulakov.ru/opencv-обучение-каскада-хаара/>
8. «Розпізнавання автомобільних номерів – огляд 16 виробників» (2019). [Електронний ресурс]. Режим доступу: <https://securityrussia.com/blog/avto-nomera.html>

9. «Використання автоматизованого розпізнавання номерних знаків. Лінії Хафа та відповідність шаблону» (2012). [Електронний ресурс]. Режим доступу: http://www.iaeng.org/publication/WCECS2012/WCECS2012_pp199-203.pdf
10. «PyCharm vs. Sublime Text» (2015). [Електронний ресурс]. Режим доступу: <https://opensourcehacker.com/2015/05/02/pycharm-vs-sublime-text/>
11. «Розпізнавання номерних знаків» [Електронний ресурс]. Режим доступу: <https://securtv.ru/presscenter/materials/raspoznavaniye-nomernykh-znakov.html>
12. «Розпізнавання номерів автомобілів - БЕЗКОШТОВНО» [Електронний ресурс]. Режим доступу: https://www.youtube.com/watch?v=UJ3X7SH9fJU&t=760s&ab_channel=%D0%A2%D0%95%D0%A5%D0%9D%D0%9E4.0