Форма № 24

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій
(повне найменування факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)
бакалавр
(ступінь вищої освіти)

на тему СТВОРЕННЯ ПРОГРАМНОЇ СИСТЕМИ ОПРАЦЮВАННЯ ДАНИХ ДЛЯ РОЗПІЗНАВАННЯ АВТОМОБІЛЬНИХ НОМЕРІВ
DESIGN OF A DATA PROCESSING SYSTEM FOR CAR NUMBER RECOGNITION

Виконав(ла): студент(ка) 4 курсу, групи КНТ-221
Спеціальності 122 Комп'ютерні науки
(код і найменування спеціальності)

Освітня програма (спеціалізація)
Комп'ютерні науки

САМОРОДІН І.Р.
(ПРІЗВИЩЕ та ініціали)
Керівник КОЦУР М.І.
(ПРІЗВИЩЕ та ініціали)
Рецензент СКРУПСЬКИЙ С.Ю.
(ПРІЗВИЩЕ та ініціали)

2026

Форма № 25

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 122 Комп'ютерні науки
(код і найменування)
Освітня програма (спеціалізація) Комп'ютерні науки
(назва освітньої програми (спеціалізації)

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
_____Сергій СУББОТІН
"____" _____2026 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

САМОРОДІНА Іллі Романовича
(ПРІЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Створення програмної системи опрацювання даних для розпізнавання автомобільних номерів. Design of a Data Processing System for Car Number Recognition

керівник проєкту (роботи) к.т.н., доцент, КОЦУР Михайло Ігорович ,
(науковий ступінь, вчене звання, ПРІЗВИЩЕ, ім'я, по батькові)
затверджені наказом закладу вищої освіти від " 7 " квітня 2026 року № 139
2. Строк подання студентом проєкту (роботи) 03 червня 2026 року
3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

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

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

6. Консультанти розділів проєкту (роботи)

Розділ	ПРІЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	КОЦУР М.І., доцент		
Нормоконтроль	ДЕЙНЕГА Л.Ю., ст. викладач		

7. Дата видачі завдання “ 7 ” квітня 2026 року.

КАЛЕНДАРНИЙ ПЛАН

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

Студент(ка) _____ Ілля САМОРОДІН
(підпис) (Імя ПРІЗВИЩЕ)

Керівник проєкту (роботи) _____ Михайло КОЦУР
(підпис) (Імя ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 83 с., 3 табл., 26 рис., 3 дод., 19 джерел.

С++, VISUAL STUDIO, АВТОМОБІЛЬНИЙ НОМЕР, ЗОБРАЖЕННЯ, МОВА ПРОГРАМУВАННЯ, РОЗПІЗНАВАННЯ, СИСТЕМА ОПРАЦЮВАННЯ ДАНИХ.

Об’єкт дослідження – процеси обчислень, пов’язані з організацією та опрацюванням даних для розпізнавання автомобільних номерів.

Предмет дослідження – програмні системи опрацювання даних для розпізнавання автомобільних номерів.

Мета роботи – розробка програмної системи опрацювання даних для розпізнавання автомобільних номерів для зменшення часу ідентифікації номерних знаків.

Матеріали, методи та технічні засоби: мова програмування С++, середовище розробки Visual Studio.

Результати. Розроблено проєктні рішення для створення системи опрацювання даних для розпізнавання автомобільних номерів. Створено систему опрацювання даних для розпізнавання автомобільних номерів. Проведено дослідження розробленої системи опрацювання даних для розпізнавання автомобільних номерів.

Висновки. Мету роботи досягнуто. Розроблено систему опрацювання даних для розпізнавання автомобільних номерів за допомогою мови програмування С++ та середовища розробки Visual Studio.

Галузь використання – системи контролю дорожнього руху.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 83 pages, 3 tables, 26 figures, 3 appendixes, 19 sources.

C++, VISUAL STUDIO, LICENSE PLATE, IMAGE, PROGRAMMING LANGUAGE, RECOGNITION, DATA PROCESSING SYSTEM.

The object of research is computing processes related to the organization and processing of data for recognizing license plates.

The subject of the research is data processing systems for recognizing license plates.

The purpose of this work is to develop the data processing system for recognizing license plates to reduce the time for identifying license plates.

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

Results. Project solutions for the creation of data processing system for recognizing license plates has been designed. The data processing system for recognizing license plates has been developed. The study of the developed data processing system for recognizing license plates has been conducted.

Conclusions. The goal of the work has been achieved. The data processing system for recognizing license plates using the C++ programming language and the Visual Studio development environment has been developed.

Scope of use – traffic control systems.

ЗМІСТ

С.

Перелік скорочень та умовних познак 8
Вступ 9
1 Аналіз предметної області 11
1.1 Системи опрацювання даних для розпізнавання автомобільних номерів 11
1.2 Аналіз систем опрацювання даних для розпізнавання автомобільних номерів 19
1.3 Висновки за розділом 1 29
2 Матеріали і методи 31
2.1 Вибір мови програмування 31
2.2 Вибір середовища розробки для створення системи опрацювання даних для розпізнавання автомобільних номерів 34
2.3 Висновки за розділом 2 38
3 Опис системи опрацювання даних 39
3.1 Структура системи опрацювання даних для розпізнавання автомобільних номерів 39
3.2 Функціонування системи опрацювання даних для розпізнавання автомобільних номерів 41
3.3 Розробка бази даних системи опрацювання даних для розпізнавання автомобільних номерів 44
3.4 Проєктування інтерфейсу системи опрацювання даних для розпізнавання автомобільних номерів 46
3.5 Висновки за розділом 3 49
4 Експлуатація, тестування та експериментальне дослідження системи опрацювання даних 50
4.1 Призначення й умови застосування системи опрацювання даних 50
4.2 Характеристики системи опрацювання даних для розпізнавання автомобільних номерів 51

4.3 Інструкція по експлуатації програми 52
4.3.1 Звернення до програми 52
4.3.2 Вхідні й вихідні дані 53
4.3.3 Повідомлення 53
4.4 Виконання системи опрацювання даних для розпізнавання автомобільних номерів 53
4.5 Тестування системи опрацювання даних для розпізнавання автомобільних номерів 56
4.6 Висновки за розділом 4 57
Висновки 58
Перелік джерел посилання 61
Додаток А Технічне завдання 63
Додаток Б Фрагмент тексту програми 67
Додаток В Слайди презентації 76

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

ANPR – Automatic Number Plate Recognition;

API – Application Programming Interface;

АН – автомобільний номер;

НМ – нейронна мережа;

РО – розпізнавання образів;

СОД – система опрацювання даних.

ВСТУП

Актуальність створення системи опрацювання даних для розпізнавання автомобільних номерів зумовлюється стрімким зростанням кількості транспортних засобів, ускладненням транспортної інфраструктури та підвищенням вимог до безпеки дорожнього руху. В умовах інтенсивної урбанізації та цифрової трансформації суспільства виникає об’єктивна потреба в автоматизованих засобах ідентифікації транспортних засобів, здатних функціонувати безперервно, швидко й з високим рівнем точності. Ручні методи фіксації номерних знаків втрачають ефективність через обмеженість людських ресурсів, ризик помилок та неможливість оперативного реагування на події в реальному часі [1], [2].

Сфера застосування подібних систем охоплює контроль доступу до закритих територій, адміністрування платних паркувальних зон, моніторинг транспортних потоків, автоматичну фіксацію порушень правил дорожнього руху, а також інтеграцію з міськими інформаційними платформами. Впровадження технологій розпізнавання номерних знаків сприяє підвищенню дисципліни водіїв, оптимізації процесів обліку та скороченню часу обробки інформації. У межах концепцій розумного міста, що активно реалізуються у багатьох країнах, такі рішення стають невід’ємною складовою цифрової інфраструктури [3], [4].

Крім того, зростає потреба в аналітичній обробці транспортних даних для прогнозування навантаження на дорожню мережу, планування інфраструктурних змін та підвищення екологічної ефективності міського середовища. Системи розпізнавання номерних знаків здатні виступати джерелом структурованих даних, що використовуються для статистичного аналізу, моделювання транспортних потоків та оцінювання ефективності регуляторних заходів. Таким чином, створення системи опрацювання даних для розпізнавання автомобільних номерів відповідає сучасним технологічним тенденціям, соціальним запитам і вимогам до безпеки, забезпечуючи

комплексний ефект у сфері управління транспортом, автоматизації процесів і цифрової трансформації інфраструктури [4]-[6].

Проте деякі системи опрацювання даних для розпізнавання автомобільних номерів характеризуються вразливістю до варіативності погодних умов, форматів номерних знаків різних країн або регіонів у межах однієї держави. Необхідність постійного оновлення баз даних, адаптації алгоритмів до нових стандартів та перенавчання моделей створює додаткове навантаження на розробників і операторів таких програмних систем. Високі вимоги до обчислювальних ресурсів, особливо при обробці потокового відео в режимі реального часу, зумовлюють потребу у продуктивному обладнанні та масштабованій інфраструктурі, що підвищує загальну вартість впровадження й експлуатації.

Тому актуальною є розробка системи опрацювання даних для розпізнавання автомобільних номерів. У дипломній кваліфікаційній роботі бакалавра розв’язується актуальне завдання розробки системи опрацювання даних для розпізнавання автомобільних номерів для зменшення часу ідентифікації номерних знаків.

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

– виконати аналіз предметної області та систем опрацювання даних для розпізнавання автомобільних номерів;

– здійснити проєктування системи опрацювання даних для розпізнавання автомобільних номерів;

– створити систему опрацювання даних для розпізнавання автомобільних номерів;

– дослідити розроблену систему опрацювання даних для розпізнавання автомобільних номерів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Системи опрацювання даних для розпізнавання автомобільних номерів

Система опрацювання даних для розпізнавання автомобільних номерів являє собою програмно-апаратний комплекс, призначений для автоматичного виявлення, зчитування та інтерпретації символів номерних знаків транспортних засобів. У її основі лежать алгоритми комп’ютерного зору, машинного навчання та аналізу зображень, за допомогою яких здійснюється обробка відеопотоку або статичних кадрів, виділення області номерного знака та перетворення графічної інформації на текстові дані. Отримані результати передаються до інформаційних систем для подальшого зберігання, пошуку або аналітичної обробки [1].

Функціонування таких програмних рішень зазвичай пов’язується з інтеграцією камер спостереження, серверного програмного забезпечення та баз даних. Обробка може виконуватися як централізовано, так і безпосередньо на периферійних пристроях, що дозволяє адаптувати архітектуру до конкретних умов експлуатації. Передбачається можливість роботи в режимі реального часу, що забезпечує оперативне реагування на події та автоматичне формування сповіщень у разі виявлення визначених номерних знаків [1].

До особливостей таких програмних систем відносять здатність функціонувати за різних погодних умов і рівнів освітлення, підтримку різних форматів номерних знаків, а також можливість масштабування від невеликих об’єктів до розподілених інфраструктур. Характерною рисою вважається інтеграція з іншими інформаційними платформами, зокрема системами безпеки, контролю доступу або управління транспортними потоками. Передбачається також ведення архіву з можливістю пошуку за номером, часом або місцем фіксації [1].

Переваги використання подібних систем пов’язуються з підвищенням ефективності контролю та автоматизацією рутинних процесів. Завдяки

автоматичному зчитуванню номерних знаків зменшується вплив людського фактора, підвищується швидкість обробки інформації та забезпечується більша точність порівняно з ручним введенням даних. Забезпечується сприяння оптимізації роботи служб безпеки, муніципальних структур і комерційних підприємств, оскільки скорочується час реагування на інциденти та спрощується моніторинг транспортних засобів. У результаті впровадження таких рішень досягається більш системний підхід до управління транспортними потоками та контролю доступу на об’єктах різного призначення [1].

Програмні комплекси для автоматичного розпізнавання номерних знаків змінюють підходи до контролю, ідентифікації та управління транспортними засобами. Вони застосовуються в різних сферах, від організації паркування до роботи правоохоронних структур і систем доступу, забезпечуючи автоматичне визначення номерних знаків за допомогою технологій комп’ютерного зору та штучного інтелекту. У роботі таких систем поєднуються технології захоплення зображень, аналізу відео, оптичного розпізнавання символів та алгоритми штучного інтелекту для виявлення транспортних засобів і витягання інформації з номерних знаків. Дані можуть отримуватися як із статичних фото, так і з потокового відео, що надходить з камер, розташованих на дорогах, в’їздах або паркувальних майданчиках [1].

Системи розпізнавання номерних знаків здатні автоматично ідентифікувати транспортний засіб та його державний номер, перетворювати буквено-цифрові символи на машинозчитуваний формат, фіксувати додаткові метадані, включно з часом, датою та місцем зйомки, а також запускати автоматичні дії, такі як відкриття воріт, відправка сповіщень або оновлення баз даних. Сучасні рішення часто виходять за межі самої інформації про номерний знак і дозволяють отримувати додаткову аналітику транспортного засобу, включаючи марку, модель, колір, клас та загальний стан [1].

Функціонування системи відбувається за кілька основних етапів. Спершу здійснюється захоплення зображень або відеопотоку за допомогою

стаціонарних камер, встановлених на воротах, стовпах або будівлях, або мобільних камер, розміщених на патрульних автомобілях чи портативних пристроях. Під час цього враховуються умови освітлення та використовуються денні і нічні камери або інфрачервона підсвітка. Наступним кроком програмне забезпечення аналізує зображення для визначення регіону інтересу, що містить номерний знак, із застосуванням алгоритмів комп'ютерного зору та машинного навчання для відокремлення його від фону. Потім засоби оптичного розпізнавання символів перетворюють графічні дані номерного знака у текстовий формат, причому сучасні системи враховують регіональні особливості номерних знаків, їх шрифти та макети для підвищення точності. На фінальному етапі розпізнані дані поєднуються з метаданими і передаються на інформаційні панелі, в бази даних або інтегруються зі сторонніми системами через API, що забезпечує автоматизацію, звітність і підтримку робочих процесів для ефективного контролю та дотримання правил [1].

Програмні комплекси для автоматизованого розпізнавання номерних знаків складаються з декількох взаємопов'язаних елементів, які разом забезпечують ефективне виявлення і обробку транспортних засобів. Центральним компонентом виступає апаратна частина, представлена стаціонарними або мобільними камерами, здатними працювати у складних умовах освітлення, часто з використанням інфрачервоного підсвічування для нічної зйомки. Основним програмним механізмом є модуль розпізнавання номерних знаків, що працює на основі алгоритмів штучного інтелекту і включає технології перетворення графічних даних у текстову форму. Цей модуль може функціонувати безпосередньо на пристрої, у локальній мережі або у хмарній/гібридній архітектурі, забезпечуючи гнучке розгортання в різних середовищах. Для взаємодії з користувачами та сторонніми системами передбачені панелі керування та API, які дозволяють переглядати результати, управляти журналами та інтегрувати інформацію у ширші системи моніторингу та контролю [1], [2].

Традиційні архітектури систем розпізнавання автомобільних номерів спираються на локальні сервери, що обробляють дані поблизу камер, однак сучасні платформи пропонують хмарні та гібридні варіанти розгортання, що забезпечує більшу гнучкість, масштабованість і менші витрати на інфраструктуру. Хмарні рішення дозволяють інтегрувати API для сторонніх додатків, централізовано управляти декількома майданчиками та автоматично оновлювати моделі штучного інтелекту без ручного втручання. В гібридній моделі первинне розпізнавання здійснюється на периферійних пристроях або безпосередньо на камерах, а додатковий аналіз та перевірка здійснюються у хмарі, що забезпечує повну аналітику та підвищує точність системи навіть у випадках низької достовірності даних [1], [2].

Переваги впровадження таких програмних систем включають підвищення ефективності робочих процесів за рахунок автоматизації рутинних операцій, високу точність розпізнавання номерних знаків у різних умовах експлуатації, швидку обробку подій у реальному часі та покращення безпеки контролю периметра. Вони знижують витрати, пов’язані з ручною обробкою даних і складною локальною інфраструктурою. При виборі відповідного програмного рішення звертається увага на підтримку номерних знаків різних країн, високу точність розпізнавання навіть за слабкого освітлення та руху, наявність API і зручної платформи для розробників, сумісність з хмарними або гібридними архітектурами, роботу з типовими IP-камерами та можливість отримання додаткової аналітики транспортних засобів, що виходить за межі самого номерного знака [1], [2].

Методи автоматичного розпізнавання номерних знаків (Automatic Number Plate Recognition, ANPR) ґрунтуються на комплексі алгоритмів комп’ютерного зору та машинного навчання, які послідовно перетворюють відео- або зображення транспортних засобів на структуровану інформацію про реєстраційні номери. Основою процесу є конвеєр обробки даних, де вихідні кадри з камер проходять крізь низку етапів для отримання точних результатів [3]-[8].

Перший етап включає ідентифікацію транспортних засобів та їхніх класів на зображенні. Використовуються нейронні мережі глибокого навчання, натреновані на великих наборах даних, здатні визначати автомобілі, вантажівки, мотоцикли та інші категорії в реальному часі. Такі алгоритми дозволяють одночасно розпізнавати кілька об’єктів на кадрі, забезпечуючи точність навіть при швидкому русі або складних умовах освітлення [3], [4].

На наступному етапі відбувається обробка зображень, спрямована на підвищення якості кадрів перед застосуванням оптичного розпізнавання символів. Методи обробки включають корекцію кольору, підвищення різкості, обрізання та нормалізацію кадрів, що дозволяє алгоритмам точно визначати контури номерних знаків навіть при затіненні, частковій оклюзії або поганій погоді. Застосовуються як класичні бібліотеки комп’ютерного зору, так і сучасні фреймворки для підготовки даних до розпізнавання [3], [4].

Оптичне розпізнавання символів служить для перетворення зображення номерного знака у текстовий формат. Використовуються алгоритми, навчені на локальних наборах шрифтів та форматів номерних знаків, здатні ідентифікувати букви та цифри з високою точністю. Крім того, застосовуються методи зіставлення шаблонів, коли отриманий номер порівнюється з бібліотекою відомих знаків для підвищення достовірності розпізнавання [3], [4].

Процес ANPR охоплює кілька ключових етапів: спочатку камера фіксує транспортний засіб у відеопотоці або на фотографії, часто з використанням інфрачервоного підсвічування для нічного часу. Потім на зображенні локалізується номерний знак за допомогою алгоритмів виявлення об’єктів, виділяється його область і проводиться нормалізація для підвищення якості. Далі програмне забезпечення OCR перетворює виявлену область у машинозчитуваний текст, враховуючи регіональні та національні відмінності у форматах номерів [5], [6].

Отримана інформація використовується для зберігання у базах даних, інтеграції з іншими інформаційними системами та автоматизації робочих

процесів. Розпізнані номерні знаки порівнюються з реєстраційними записами або чорними та білими списками для визначення подальших дій, таких як сповіщення, обмеження доступу або формування звітів. Сучасні системи ANPR постійно вдосконалюються, щоб підвищувати точність, швидкість обробки та надійність навіть у складних і динамічних середовищах [5], [6].

Системи автоматичного розпізнавання номерних знаків забезпечують широкий спектр переваг, які роблять їх ключовими для сучасного управління транспортними потоками та безпекою. Вони дозволяють значно скоротити обсяг ручної праці, забезпечують швидке і точне виявлення транспортних засобів та покращують ефективність обслуговування. Автоматизація процесів дає змогу керувати сповіщеннями, доступом та контролем руху без втручання людини, що особливо важливо у великих містах або на платних дорогах, де потрібна безперервна обробка даних [5], [6].

Системи ANPR відрізняються високою точністю, здатні швидко ідентифікувати автомобілі навіть у складних умовах освітлення та при високій швидкості руху. Зібрані дані використовуються для аналітики транспортних потоків, що дозволяє оптимізувати мобільність, зменшувати затори та підвищувати ефективність реагування на інциденти. Розпізнавання номерних знаків також забезпечує швидку ідентифікацію транспортних засобів для контролю доступу на об’єктах, відстеження конкретних автомобілів або організації автоматизованих процедур безпеки [5], [6].

Економічна ефективність ANPR проявляється у здатності обробляти великі обсяги даних без людського втручання, що дозволяє скоротити витрати на управління та зменшити час очікування користувачів. Витрати на встановлення та експлуатацію систем відносно низькі, а технологія сумісна з різними типами камер і обладнання. Масштабованість систем забезпечується за рахунок інтеграції передових технологій, таких як периферійні обчислення та штучний інтелект на пристроях, що дозволяє реалізовувати автономну обробку відеопотоків і розширювати функціонал системи без необхідності великих серверних ресурсів [5], [6].

Зручність використання ANPR полягає у легкій інтеграції з іншими IT-системами, що забезпечує безперебійний досвід для користувачів та відкриває можливості для нових сервісів, таких як автоматизовані платежі за паркування. Універсальність технології дозволяє застосовувати її в різних сферах, включаючи управління паркуванням, контроль доступу, моніторинг дорожнього руху, оптимізацію логістики та підвищення безпеки на об’єктах [7], [8].

Використання ANPR також має позитивний ефект для навколишнього середовища, оскільки сприяє зменшенню заторів і скороченню непотрібних поїздок у пошуку місця для паркування. Це допомагає зменшити викиди шкідливих речовин і підвищує загальну екологічну ефективність міської інфраструктури, сприяючи більш організованому та безпечному руху транспорту. Системи ANPR, поєднуючи автоматизацію, аналітику та високоточне розпізнавання, стають невід’ємним інструментом сучасного управління транспортом, безпеки та оптимізації операцій у міських і комерційних середовищах [7], [8].

Системи автоматичного розпізнавання номерних знаків широко використовуються у різних сферах, де критично важливо ідентифікувати, відстежувати та локалізувати транспортні засоби. У правоохоронних органах такі системи дозволяють перевіряти реєстрацію автомобілів, виявляти транспортні засоби, пов’язані з порушеннями чи злочинами, та оперативно визначати їхнє місцезнаходження. Завдяки можливості обробки даних у реальному часі, органи влади можуть ефективно реагувати на інциденти, відстежувати підозрілі автомобілі та проводити розслідування без затримок [7], [8].

У сфері управління паркуванням ANPR забезпечує повну автоматизацію контролю за автомобілями на об’єктах. Кожен транспортний засіб враховується за номерним знаком, що дозволяє паркувальним системам вести точний облік, відстежувати заповненість паркомісць та забезпечувати користувачам безперервний досвід без необхідності контролювати квитки чи

оплачувати додатково час перебування. Інтелектуальна аналітика дозволяє не лише керувати наявними паркувальними місцями, а й оптимізувати процес розподілу автомобілів, зменшуючи стрес та ризик помилок у користувачів [7], [8].

Аналітичні можливості ANPR охоплюють і дорожній рух. Вимірювання часу подорожі між вузлами дозволяє органам влади планувати маршрути та оцінювати ефективність транспортної інфраструктури. Автоматичне розпізнавання номерних знаків сприяє виявленню високошвидкісних або небезпечних водіїв, а також аналізу ситуацій у місті для оптимізації руху та зменшення заторів. У масштабі міських систем ANPR надає дані для моделювання потоків транспорту, виявлення вузьких місць та оперативного реагування на надзвичайні ситуації [7], [8].

У комерційних зонах та торговельних парках система допомагає контролювати доступ транспортних засобів, запобігаючи незаконному паркуванню та підвищуючи безпеку об’єктів. У сфері платних доріг ANPR дозволяє автоматизувати збір плати за проїзд, зменшуючи потребу у ручному контролі та скорочуючи час зупинок для водіїв. Така автоматизація забезпечує ефективність пунктів стягнення плати, дозволяє впроваджувати динамічне ціноутворення та підвищує загальну продуктивність дорожньої системи. Усі ці випадки демонструють, як ANPR стає невід’ємним інструментом для управління транспортом, підвищення безпеки та оптимізації міських і комерційних процесів [7], [8].

Визначено, що система опрацювання даних для розпізнавання автомобільних номерів являє собою програмно-апаратний комплекс, призначений для автоматичного виявлення, зчитування та інтерпретації символів номерних знаків транспортних засобів. У її основі лежать алгоритми комп’ютерного зору, машинного навчання та аналізу зображень, за допомогою яких здійснюється обробка відеопотоку або статичних кадрів, виділення області номерного знака та перетворення графічної інформації на текстові дані.

1.2 Аналіз систем опрацювання даних для розпізнавання автомобільних номерів

Проаналізуємо системи опрацювання даних для розпізнавання автомобільних номерів [9]-[15].

Програмне забезпечення Neurotechnology SentiVeillance (рис. 1.1) є багатофункціональним програмним середовищем для організації інтелектуального відеоспостереження з підтримкою автоматизованого зчитування автомобільних номерних знаків. У межах цієї програмної системи реалізовано механізми аналізу відеопотоків, що надходять з мережевих камер, із подальшим виділенням транспортних засобів у кадрі та ідентифікацією їхніх реєстраційних позначень. Обробка здійснюється без необхідності попереднього збереження відео, що дає змогу виконувати розпізнавання безпосередньо під час надходження зображення [9], [10], [13].

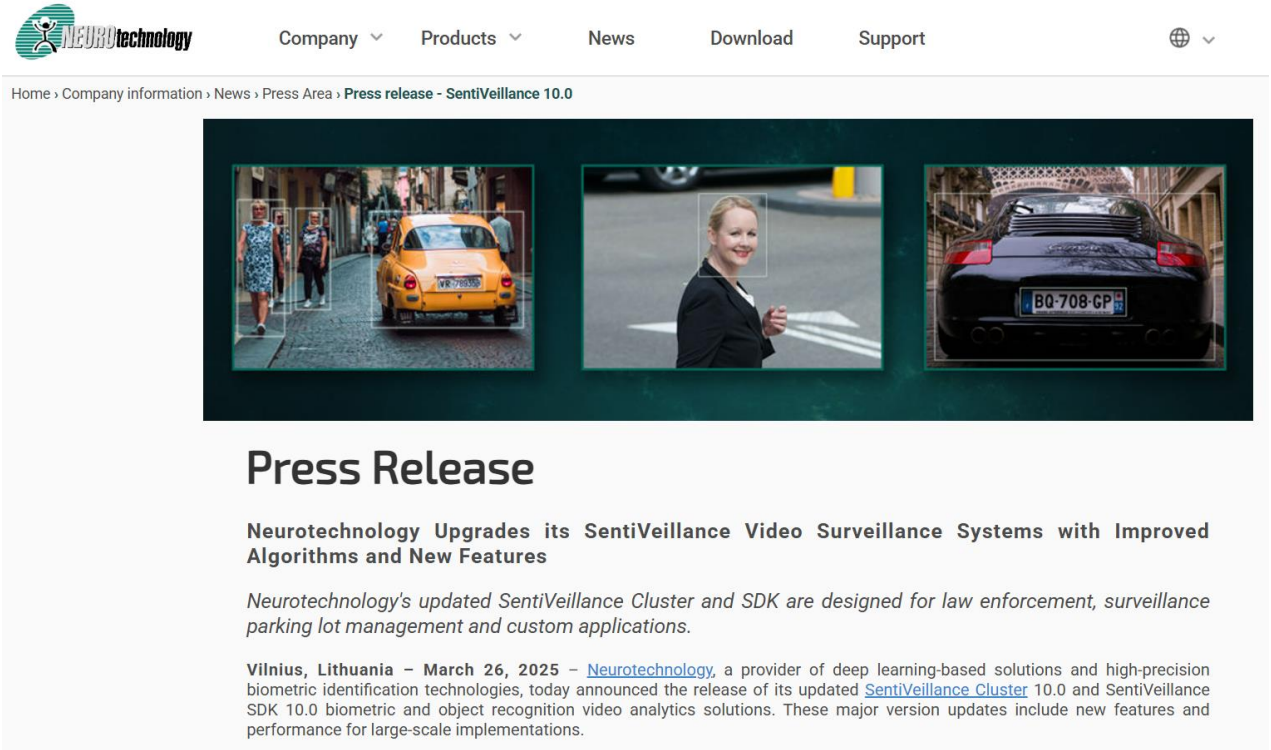


Рисунок 1.1 – Програмна система Neurotechnology SentiVeillance [13]

Платформа Neurotechnology SentiVeillance орієнтована на роботу з номерними знаками різних країн і регіонів, враховуючи відмінності у форматах, шрифтах та структурі символів. Алгоритмічні модулі здатні зберігати стабільність функціонування за умов змінного освітлення, руху транспортних засобів із підвищеною швидкістю, часткових перекриттів або зйомки під непрямим кутом. Передбачається використання адаптивних методів обробки зображень, що дозволяють мінімізувати вплив шумів і зовнішніх факторів на якість результату [9], [10], [13].

У межах єдиної архітектури Neurotechnology SentiVeillance реалізовано поєднання функцій автоматичного розпізнавання номерних знаків із засобами біометричної ідентифікації та аналітики об’єктів. Це створює основу для формування комплексної системи контролю доступу та моніторингу подій. Програмне середовище може інтегруватися з системами керування відеоспостереженням і сторонніми інформаційними ресурсами, що забезпечує гнучкість під час побудови розподіленої інфраструктури безпеки [9], [10], [13].

Система Neurotechnology SentiVeillance характеризується високим рівнем точності розпізнавання за різних сценаріїв експлуатації та підтримує масштабування відповідно до потреб об’єкта. Передбачено можливість розгортання у конфігураціях із великою кількістю камер із централізованим управлінням та оперативним сповіщенням про виявлені події. Разом із тим налаштування та оптимізація параметрів вимагають належної технічної підготовки, оскільки для досягнення максимальної ефективності необхідно коректно конфігурувати серверне середовище, мережеву інфраструктуру та параметри аналітики [9], [10], [13].

Економічна модель Neurotechnology SentiVeillance використання передбачає ліцензування, що формується залежно від конфігурації та кількості підключених каналів. Вартість визначається індивідуально з урахуванням обраного функціоналу та масштабу впровадження, що може бути відчутним для невеликих інсталяцій, проте є обґрунтованим у випадку створення комплексних систем безпеки. Для забезпечення стабільної роботи

рекомендується застосування продуктивного апаратного забезпечення, здатного обробляти інтенсивні відеопотоки та виконувати аналітичні операції без затримок [9], [10], [13].

Загалом платформа Neurotechnology SentiVeillance орієнтована на великі організації, операторів інфраструктурних об’єктів та спеціалізовані служби безпеки, яким необхідна інтегрована система відеоаналітики з розширеними можливостями розпізнавання номерних знаків у складі єдиної екосистеми спостереження [9], [10], [13].

Основними характеристиками програмної системи Neurotechnology SentiVeillance є такі [9], [10], [13]:

– інтелектуальна відеоаналітика: у програмній системі Neurotechnology SentiVeillance забезпечується автоматичний аналіз потокового відео з мережевих камер із виявленням транспортних засобів, виділенням номерних знаків та їх подальшим розпізнаванням у режимі безперервної обробки даних [9], [10], [13];

– підтримка різних форматів номерів: у програмній системі Neurotechnology SentiVeillance реалізовано алгоритми, здатні коректно працювати з номерними знаками автомобілів різних країн і типів, з урахуванням особливостей шрифтів, структури символів і способів кріплення табличок [9], [10], [13];

– інтеграція з біометрією: програмна система Neurotechnology SentiVeillance може поєднувати функції розпізнавання номерних знаків із модулями ідентифікації облич та іншими біометричними технологіями в межах єдиної системи безпеки [9], [10], [13];

– масштабованість архітектури: передбачено можливість розгортання як у невеликих конфігураціях, так і в розподілених інфраструктурах із централізованим керуванням великою кількістю відеоканалів [9], [10], [13];

– сумісність із системами відеокерування: підтримується інтеграція з популярними платформами керування відеоспостереженням, що спрощує впровадження у вже існуюче середовище безпеки [9], [10], [13];

– робота в складних умовах: алгоритми у програмній системі Neurotechnology SentiVeillance адаптовано до змін освітлення, руху об’єктів із високою швидкістю та нестандартних кутів огляду, що дозволяє зберігати стабільність результатів [9], [10], [13];

– гнучке ліцензування: у програмній системі Neurotechnology SentiVeillance застосовується модель ліцензування з урахуванням кількості каналів і функціональних модулів, що дає змогу адаптувати конфігурацію під потреби конкретного об’єкта [9], [10], [13];

– централізоване адміністрування: у програмній системі Neurotechnology SentiVeillance надаються інструменти керування користувачами, ролями доступу та подіями, що підвищує рівень контролю та інформаційної безпеки [9], [10], [13].

Серед недоліків програмної системи Neurotechnology SentiVeillance можна відзначити складність початкового налаштування, що потребує залучення фахівців із відповідною технічною підготовкою, а також необхідність використання продуктивного серверного обладнання для досягнення максимальної ефективності роботи системи [9], [10], [13].

Програмна система PlateSmart (рис. 1.2) позиціонується як інтелектуальне програмне рішення для автоматизованого розпізнавання номерних знаків транспортних засобів, побудоване на основі сучасних методів машинного навчання. У межах цієї системи забезпечується аналіз відеопотоків, що надходять із мережевих камер загального призначення, без потреби у використанні спеціалізованих апаратних комплексів. Обробка здійснюється програмними засобами, що дозволяє інтегрувати рішення у вже наявну інфраструктуру відеоспостереження без суттєвих змін у технічному середовищі. Алгоритмічна основа платформи PlateSmart ґрунтується на глибоких нейронних моделях, здатних стабільно працювати за різноманітних умов експлуатації. Розпізнавання номерних знаків супроводжується додатковим аналізом характеристик транспортного засобу, зокрема визначенням його типу, марки, моделі та кольору [9], [10], [14].

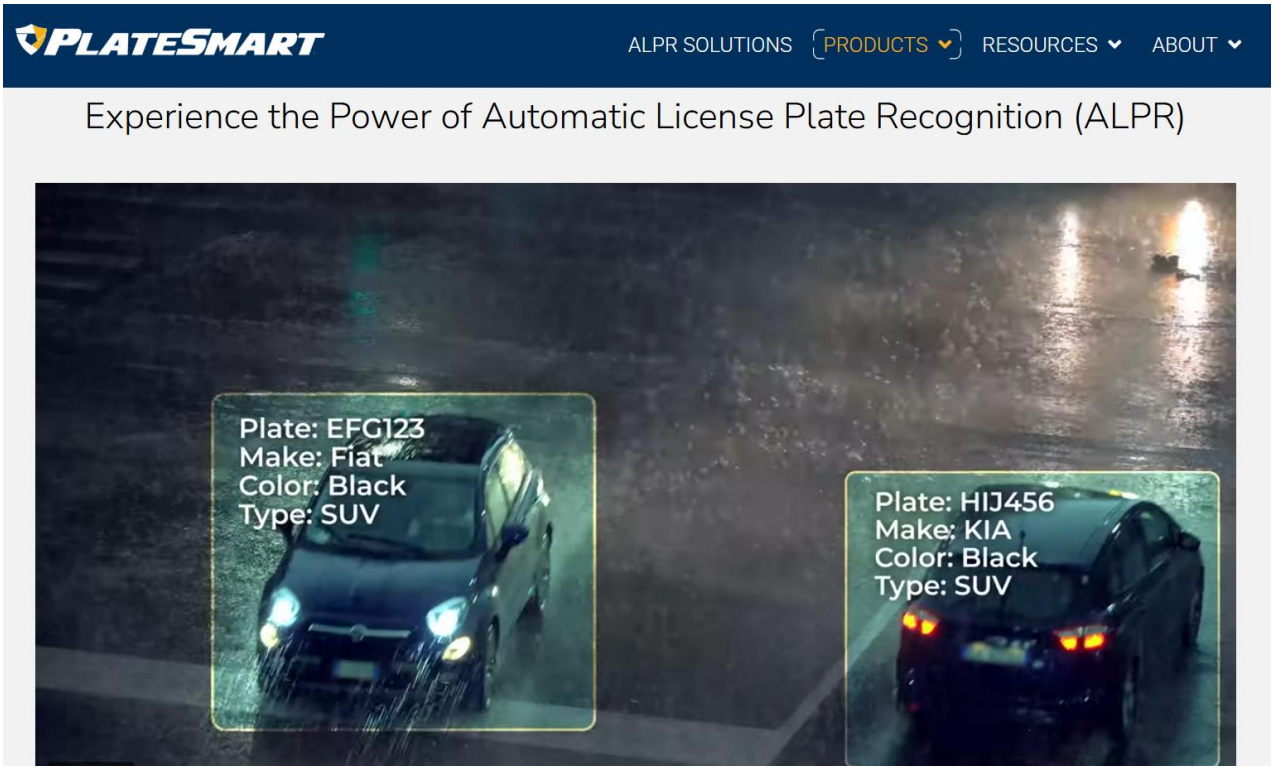


Рисунок 1.2 – Програмна система PlateSmart [14]

Архітектура рішення PlateSmart передбачає можливість виконання обчислень безпосередньо на периферійних пристроях або в межах локальної інфраструктури, що сприяє зменшенню затримок під час обробки даних та підвищенню рівня конфіденційності. Завдяки цьому інформація може аналізуватися поблизу джерела її отримання, без обов'язкової передачі до централізованих серверів. Такий підхід також полегшує масштабування системи в умовах розподілених об'єктів [9], [10], [14].

Програмна система PlateSmart орієнтоване на використання у сферах організації паркувального простору, контролю доступу, забезпечення безпеки об'єктів та автоматизації процесів стягнення плати за проїзд. У програмній системі PlateSmart Передбачено можливість інтеграції з корпоративними інформаційними системами через прикладні програмні інтерфейси, що спрощує включення платформи до складних IT-екосистем. Масштабованість дозволяє застосовувати рішення як у межах окремих майданчиків, так і в масштабі міської інфраструктури [9], [10], [14].

Водночас ефективність функціонування програмної системи PlateSmart значною мірою залежить від якості відеопотоку та правильності розміщення камер, оскільки некоректний кут огляду або недостатня чіткість зображення можуть впливати на точність результатів. Початкове впровадження та налаштування вимагають технічної компетентності, особливо під час інтеграції з існуючими системами. Модель ліцензування формується індивідуально з урахуванням масштабу використання та обраного функціоналу, що робить рішення більш орієнтованим на корпоративний сегмент, ніж на невеликі інсталяції [9], [10], [14].

Загалом платформа PlateSmart призначена для організацій та муніципальних структур, яким необхідне гнучке та масштабоване програмне рішення для розпізнавання номерних знаків без інвестування у спеціалізоване обладнання, із можливістю глибокої інтеграції та централізованого керування аналітичними процесами [9], [10], [14].

Програмна система PlateSmart має такі особливості [9], [10], [14]:

– інтелектуальне розпізнавання: у програмній системі PlateSmart застосовуються моделі глибокого навчання для високоточного визначення номерних знаків у відеопотоці з IP-камер [9], [10], [14];

– у програмній системі PlateSmart одночасно здійснюється ідентифікація марки, моделі та кольору транспортного засобу, що підвищує точність аналітики [9], [10], [14];

– робота з існуючими камерами: у програмній системі PlateSmart не вимагається спеціалізоване обладнання, система сумісна з будь-якими мережевими камерами [9], [10], [14];

– обробка на периферії: частина аналітичних обчислень виконується безпосередньо на пристроях, що зменшує затримки та підвищує конфіденційність [9], [10], [14];

– інтеграція через API: у програмній системі PlateSmart передбачено масштабовану інтеграцію з корпоративними системами та платформами управління [9], [10], [14];

– адаптивність до умов: алгоритми коригуються залежно від освітлення, погодних умов та кутів огляду камер [9], [10], [14];

– масштабованість: програмна система PlateSmart дозволяє реалізовувати рішення як для окремих об’єктів, так і для розподілених інфраструктур із великою кількістю відеоканалів [9], [10], [14];

– централізоване адміністрування: у програмній системі PlateSmart забезпечується керування користувачами, ролями та подіями в межах єдиної системи [9], [10], [14].

Серед недоліків програмної системи PlateSmart можна відзначити високу залежність продуктивності від якості та розташування камер, складність початкового впровадження та інтеграції, що потребує спеціальної IT-компетенції, а також обмежену документацію для користувачів менших об’єктів, що може ускладнювати налаштування та експлуатацію у невеликих масштабах [9], [10], [14].

Програмна система AutoVu (рис. 1.3) є високотехнологічним програмним рішенням для автоматизованого розпізнавання номерних знаків транспортних засобів, яке поєднує програмне забезпечення та камери на основі штучного інтелекту [9], [10], [15].

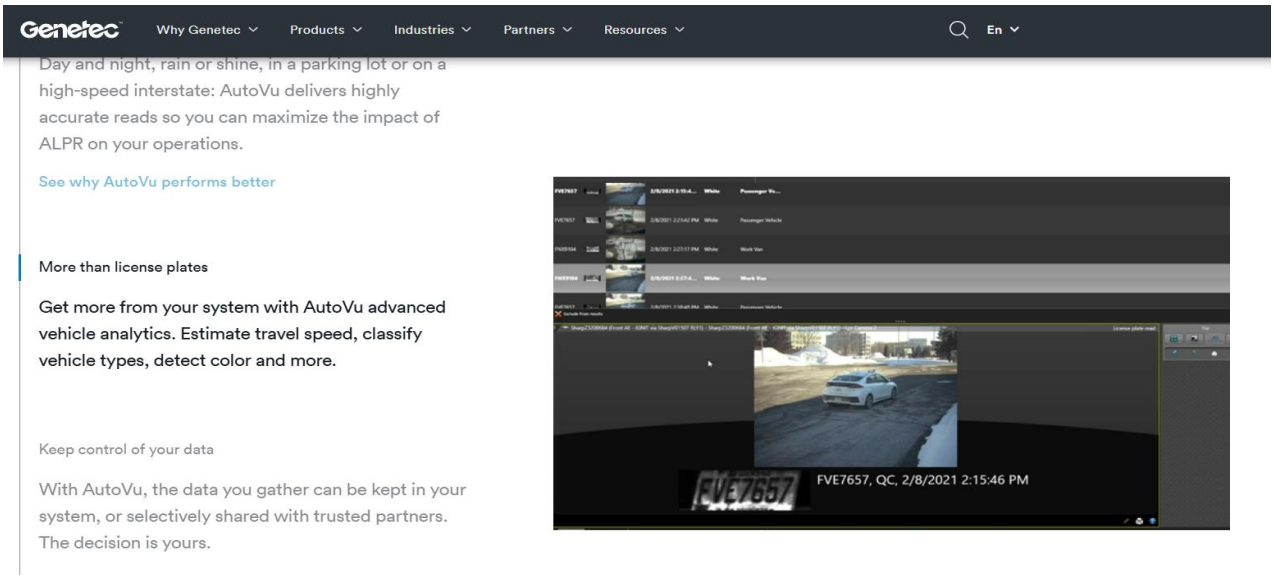


Рисунок 1.3 – Програмна система AutoVu [15]

У програмній системі AutoVu реалізовано алгоритми автоматичного виявлення, зчитування та аналізу номерних знаків у режимі реального часу, що дозволяє працювати в різноманітних умовах, включаючи паркові майданчики, автомагістралі та міські вулиці. Програмне забезпечення адаптоване до складних умов освітлення та погодних факторів, забезпечуючи високу точність розпізнавання навіть у ситуаціях із рухомими об’єктами та низькою видимістю [9], [10], [15].

Система AutoVu інтегрована з платформою Genetec Security Center, що дозволяє поєднувати функції управління відеоспостереженням з можливостями контролю доступу, забезпечення правил паркування та проведення судово-аналітичних розслідувань. Автоматизація процесу зчитування номерних знаків спрощує роботу органів правопорядку, муніципальних служб та комерційних підприємств, допомагаючи оперативно визначати транспортні засоби, що представляють інтерес, і контролювати дотримання правил паркування [9], [10], [15].

Рішення AutoVu передбачає використання як стаціонарних, так і мобільних камер, що робить його гнучким для широкого спектру застосувань і організацій. Система здатна захоплювати максимальну кількість номерних знаків, мінімізуючи ймовірність пропуску цільових транспортних засобів, а інтерфейс програмного забезпечення створено так, щоб оператори могли зосередитися на прийнятті рішень, а не на ручному перегляді результатів зчитування. Надійність і точність роботи підтримується в умовах будь-якого часу доби та будь-яких погодних умов, що дозволяє ефективно застосовувати систему як на парковках, так і на швидкісних магістралях [9], [10], [15].

Система AutoVu орієнтована на великі підприємства та охоронні структури, які вже використовують інфраструктуру Genetec і потребують інтегрованого, масштабованого рішення для моніторингу великого потоку транспортних засобів. Висока точність і глибока інтеграція з відеоплатформою дозволяють підвищити ефективність контролю за рухом транспортних засобів та безпеку об’єктів, забезпечуючи надійне та централізоване управління

аналітичними процесами. Ціна використання формується індивідуально та залежить від обсягу функціоналу, конфігурації камери та типу ліцензії, що робить продукт більш орієнтованим на корпоративний сегмент [9], [10], [15].

Програмна система AutoVu має такі особливості [9], [10], [15]:

– висока точність розпізнавання: у програмній системі AutoVu застосовуються інтелектуальні камери та алгоритми глибокого навчання для точної ідентифікації номерних знаків навіть за складного освітлення чи руху транспортних засобів [9], [10], [15];

– інтеграція з платформою Genetec: програмна система AutoVu повністю сумісна із Genetec Security Center для централізованого управління відеоспостереженням та аналітикою [9], [10], [15];

– підтримка стаціонарних і мобільних камер: програмна система AutoVu адаптована для роботи як у фіксованих точках, так і на пересувних об’єктах [9], [10], [15];

– масштабованість розгортання: програмна система AutoVu має можливість використання в межах одного об’єкта або в розподілених мережах з великою кількістю відеоканалів [9], [10], [15];

– автоматизація аналітики: у програмній системі AutoVu зчитування номерів відбувається без ручного перегляду, що дозволяє операторам зосередитися на прийнятті рішень [9], [10], [15];

– висока надійність роботи: точність функціонування зберігається у будь-яких погодних умовах та в різний час доби [9], [10], [15];

– підтримка корпоративних застосувань: програмна система AutoVu орієнтована на великі підприємства та охоронні структури, які потребують інтегрованого рішення для моніторингу великого потоку різних транспортних засобів [9], [10], [15];

– гнучке адміністрування: забезпечується управління користувачами, ролями та подіями через централізовану платформу [9], [10], [15].

Серед недоліків програмної системи AutoVu можна відзначити складність початкового налаштування та інтеграції, що потребує технічних

знань і підготовки користувачів, високу вартість ліцензій та апаратної інтеграції, а також оптимальну ефективність лише при використанні камер та обладнання екосистеми Genetec, що обмежує гнучкість у малих і незалежних інсталяціях [9], [10], [15].

Порівняльну характеристику систем опрацювання даних для розпізнавання автомобільних номерів наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння систем опрацювання даних для розпізнавання автомобільних номерів

Критерій порівняння	Neuro-technology SentiVeillance [13]	PlateSmart [14]	AutoVu [15]
Висока точність розпізнавання в складних умовах	+	+–	+
Підтримка роботи з існуючими IP-камерами	+	+	+–
Інтеграція зі сторонніми платформами	+–	+	+
Легкість налаштування для початку роботи	–	+–	–
Залежність від специфічного апаратного середовища	+–	–	+
Сталий функціонал аналітики транспортних характеристик	+–	+	+–

За результатами проведеного аналізу можна зробити висновок, що у наш час існує досить багато систем опрацювання даних для розпізнавання автомобільних номерів. Проте деякі системи опрацювання даних для розпізнавання автомобільних номерів характеризуються вразливістю до

варіативності погодних умов, форматів номерних знаків різних країн або регіонів у межах однієї держави. Необхідність постійного оновлення баз даних, адаптації алгоритмів до нових стандартів та перенавчання моделей створює додаткове навантаження на розробників і операторів таких програмних систем. Високі вимоги до обчислювальних ресурсів, особливо при обробці потокового відео в режимі реального часу, зумовлюють потребу у продуктивному обладнанні та масштабованій інфраструктурі, що підвищує загальну вартість впровадження й експлуатації. Тому актуальною є розробка системи опрацювання даних для розпізнавання автомобільних номерів.

При розробці системи опрацювання даних для розпізнавання автомобільних номерів необхідно забезпечити такі функціональні вимоги:

– підтримка можливості завантаження відеопотоків для подальшого розпізнавання автомобільних номерів;

– підтримка збереження результатів розпізнавання у базі даних;

– підтримка можливості перегляду історії виконаних запитів і отриманих результатів розпізнавання;

– можливість авторизації користувачів системи;

– забезпечення коректної роботи системи в умовах різної якості зображень та змін освітлення;

– підтримка можливості отримання інформації про ймовірність правильного розпізнавання автомобільного номеру.

1.3 Висновки за розділом 1

Визначено, що система опрацювання даних для розпізнавання автомобільних номерів являє собою програмно-апаратний комплекс, призначений для автоматичного виявлення, зчитування та інтерпретації символів номерних знаків транспортних засобів. У її основі лежать алгоритми комп’ютерного зору, машинного навчання та аналізу зображень, за допомогою яких здійснюється обробка відеопотоку або статичних кадрів, виділення

області номерного знака та перетворення графічної інформації на текстові дані. Отримані результати передаються до інформаційних систем для подальшого зберігання, пошуку або аналітичної обробки.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато систем опрацювання даних для розпізнавання автомобільних номерів. Проте деякі системи опрацювання даних для розпізнавання автомобільних номерів характеризуються вразливістю до варіативності погодних умов, форматів номерних знаків різних країн або регіонів у межах однієї держави. Необхідність постійного оновлення баз даних, адаптації алгоритмів до нових стандартів та перенавчання моделей створює додаткове навантаження на розробників і операторів таких програмних систем. Високі вимоги до обчислювальних ресурсів, особливо при обробці потокового відео в режимі реального часу, зумовлюють потребу у продуктивному обладнанні та масштабованій інфраструктурі, що підвищує загальну вартість впровадження й експлуатації. Тому актуальною є розробка системи опрацювання даних для розпізнавання автомобільних номерів.

Сформульовано функціональні вимоги до системи опрацювання даних для розпізнавання автомобільних номерів.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування

При розробці системи опрацювання даних для розпізнавання автомобільних номерів було використано мову програмування C++ [16], [17].

Мова програмування C++ є інструментом для високопродуктивних обчислень та повноцінною екосистемою для створення складних програмних систем різного призначення. Її еволюція відбувалася на основі мови C, що зумовило збереження ефективності та низькорівневого контролю, водночас розширивши можливості за рахунок підтримки об’єктно-орієнтованої, узагальненої та багатопарадигмальної розробки. Такий синтез дозволяє поєднувати структурний підхід із сучасними механізмами абстракції, не жертвуючи продуктивністю [16], [17].

Мова C++ є стандартизованою мовою, розвиток якої координується міжнародним комітетом ISO. Регулярне оновлення стандартів сприяє впровадженню нових мовних конструкцій, покращенню безпеки коду та підвищенню зручності розробки [16], [17].

Суттєвою характеристикою C++ є поєднання високого рівня абстракції з прямим доступом до пам’яті. Це дозволяє реалізовувати як системне програмування, так і складні прикладні рішення, включаючи графічні рушії, фінансові системи, ігрові платформи та вбудовані застосунки. Мова активно використовується в індустріях, де критично важливими є швидкодія та мінімальні затримки, зокрема в телекомунікаціях, банківських сервісах, робототехніці та високопродуктивних обчисленнях [16], [17].

Окремої уваги заслуговує стандартна бібліотека шаблонів, яка забезпечує ефективні контейнери, алгоритми та механізми роботи з ітераторами. Узагальнене програмування в C++ дозволяє створювати гнучкі та повторно використовувані компоненти без втрати швидкодії, оскільки шаблони компілюються з урахуванням конкретних типів даних. Це сприяє

формуванню масштабованих архітектур і підвищенню якості програмного забезпечення [16], [17].

Мова програмування C++ знайшла широке застосування у розробці операційних систем, драйверів та програмних платформ. Значна частина ядра сучасних систем та інструментів розробки створена з використанням цієї мови або має компоненти, реалізовані на ній. Високий рівень сумісності з іншими мовами програмування забезпечує можливість інтеграції з мовами Python, Java чи іншими технологіями через API або бібліотеки спільного використання [16], [17].

Отже, мова C++ характеризується гнучкістю, багатопарадигмальністю, високою ефективністю виконання та постійним розвитком стандартів. Поєднання системного контролю з сучасними абстракціями робить її універсальним інструментом для створення як низькорівневих, так і складних прикладних програмних рішень, що потребують надійності, масштабованості та продуктивності [16], [17].

Доцільність вибору мови програмування C++ для створення системи опрацювання даних з розпізнавання автомобільних номерів обґрунтовується високими вимогами до продуктивності, ефективності використання ресурсів та можливості роботи в режимі реального часу. Під час обробки відеопотоку або серії зображень виконується значний обсяг обчислень, пов’язаний із фільтрацією, сегментацією, детекцією об’єктів та розпізнаванням символів. У таких умовах критичним є мінімальний час відгуку та раціональне використання оперативної пам’яті, що забезпечується завдяки низькорівневому контролю над ресурсами, характерному для C++ [16], [17].

Суттєвою перевагою є можливість безпосередньої взаємодії з апаратним забезпеченням, включаючи камери відеоспостереження, спеціалізовані модулі обробки сигналів та графічні прискорювачі. Це дозволяє реалізовувати оптимізовані алгоритми з урахуванням архітектурних особливостей процесора або використання паралельних обчислень. Підтримка багатопотоковості та сучасних стандартів мови сприяє створенню

масштабованих рішень, здатних одночасно обробляти кілька потоків даних без істотного зниження швидкодії [16], [17].

Важливим чинником виступає наявність розвиненої екосистеми бібліотек для комп'ютерного зору та машинного навчання, зокрема OpenCV, яка надає широкий набір інструментів для роботи із зображеннями та відео. Значна частина алгоритмів глибинного навчання реалізована або має високопродуктивні інтерфейси саме для C++, що дозволяє інтегрувати моделі розпізнавання без втрати швидкодії. Крім того, використання цієї мови забезпечує можливість створення кросплатформних застосунків для різних операційних систем без суттєвих змін у вихідному коді [16], [17].

Надійність і передбачуваність виконання програм також відіграють роль у системах, які функціонують у безперервному режимі. Завдяки статичній типізації та компіляції безпосередньо у машинний код зменшується кількість непередбачених помилок під час виконання. Це є важливим для систем безпеки, де навіть незначні затримки або збої можуть мати критичні наслідки [16], [17].

Додатково враховується можливість глибокої оптимізації алгоритмів на рівні коду, включаючи використання SIMD-інструкцій, апаратного прискорення та спеціалізованих структур даних. У випадках, коли система має працювати на вбудованих або ресурсно обмежених пристроях, ефективність мови C++ дозволяє досягти необхідної швидкодії без надмірних апаратних витрат [16], [17].

Отже, вибір мови C++ для розроблення системи опрацювання даних з розпізнавання автомобільних номерів обумовлюється поєднанням високої продуктивності, гнучкості, широких можливостей оптимізації та наявності спеціалізованих інструментів, що відповідають вимогам до реального часу, масштабованості та надійності функціонування [16], [17].

Обґрунтування вибору мови програмування для розробки системи опрацювання даних для розпізнавання автомобільних номерів наведено у таблиці 2.1.

Таблиця 2.1 – Обґрунтування вибору мови програмування для розробки системи опрацювання даних для розпізнавання автомобільних номерів

Критерій порівняння мов програмування	Мова програмування		
	C++	Python	C#
Продуктивність виконання	+	+–	+–
Контроль управління пам’яттю	+	–	–
Інтеграція з бібліотеками комп’ютерного зору	+	+	+–
Можливість низькорівневої оптимізації	+	–	–
Стабільність у довготривалому безперервному режимі	+	+–	+

Отже, для реалізації системи опрацювання даних для розпізнавання автомобільних номерів обрано мову програмування C++, що обумовлюється поєднанням високої продуктивності, гнучкості, широких можливостей оптимізації та наявності спеціалізованих інструментів, що відповідають вимогам до реального часу, масштабованості та надійності функціонування.

2.2 Вибір середовища розробки для створення системи опрацювання даних для розпізнавання автомобільних номерів

В якості середовища розробки для створення системи опрацювання даних для розпізнавання автомобільних номерів було обрано середовище Visual Studio [18], [19].

Середовище розробки Visual Studio є комплексною платформою, що охоплює повний життєвий цикл програмного забезпечення. Воно підтримує проєктування, розроблення, тестування, налагодження, профілювання, розгортання та супровід застосунків у межах єдиного інтерфейсу. Така інтегрованість зменшує кількість сторонніх інструментів і підвищує керованість процесу розробки [18], [19].

Суттєвою перевагою Visual Studio є модульна архітектура середовища, яка дозволяє встановлювати лише необхідні компоненти залежно від типу проєкту. Підтримується розробка настільних, веб- і хмарних застосунків, сервісів, бібліотек, а також програм для вбудованих систем. Гнучкість конфігурації дає змогу адаптувати робочий простір під конкретні потреби команди або індивідуального розробника [18], [19].

Особливу увагу у середовищі розробки Visual Studio відіграє інтелектуальна система підказок і аналізу коду IntelliSense, що підвищує швидкість написання програм та зменшує кількість синтаксичних помилок. Інструменти статичного аналізу дозволяють виявляти потенційні проблеми ще до запуску програми, а механізми автоматичного рефакторингу спрощують вдосконалення структури коду без зміни його функціональності. Це є важливим для великих проєктів, де підтримка читабельності та узгодженості коду має стратегічне значення [18], [19].

Середовище розробки Visual Studio забезпечує розвинені можливості тестування, включаючи модульні тести, інструменти вимірювання покриття коду та інтеграцію з системами безперервної інтеграції. Підтримка профілювання дозволяє аналізувати продуктивність застосунків, визначати «вузькі місця» та оптимізувати алгоритми. Для систем, що обробляють значні обсяги даних або працюють у режимі реального часу, такі інструменти є особливо корисними [18], [19].

Варто також відзначити глибоку інтеграцію середовища розробки Visual Studio з екосистемою Microsoft, включаючи хмарні сервіси, системи керування версіями та платформи розгортання. Це відкриває можливості для

створення масштабованих рішень із підтримкою віддаленої роботи, спільної розробки та автоматизованого розгортання [18], [19].

Важливою характеристикою є підтримка розширень, які дозволяють адаптувати функціональність середовища під специфічні завдання. Розробник може інтегрувати додаткові засоби аналізу, інструменти роботи з базами даних, дизайнерські модулі або компоненти для роботи з контейнерами й віртуалізацією [18], [19].

Таким чином, Visual Studio виступає не просто редактором коду, а повноцінною платформою інженерної діяльності, що забезпечує комплексну підтримку процесу створення складних програмних систем, підвищує продуктивність праці розробників і сприяє підтриманню високих стандартів якості програмного забезпечення [18], [19].

Доцільність вибору середовища розробки Microsoft Visual Studio для створення системи опрацювання даних з розпізнавання автомобільних номерів зумовлюється поєднанням функціональності, стабільності та високого рівня інтеграції з інструментами, необхідними для реалізації складних алгоритмів комп’ютерного зору. Під час розроблення подібних систем виникає потреба у зручному керуванні проєктами великого обсягу, налаштуванні бібліотек, організації багатофайлової структури та забезпеченні ефективного налагодження, що повністю підтримується цим середовищем [18], [19].

Visual Studio забезпечує глибоку інтеграцію з мовою C++, включаючи підтримку сучасних стандартів, інтелектуальне автодоповнення коду, статичний аналіз та засоби профілювання продуктивності. Це дозволяє контролювати якість програмного коду, своєчасно виявляти помилки та оптимізувати ресурсоємні ділянки програми. Для систем розпізнавання номерних знаків, де критично важливою є швидкодія та мінімальні затримки, можливість детального аналізу використання пам’яті та процесорного часу має особливе значення [18], [19].

Важливою перевагою є зручна робота з зовнішніми бібліотеками, зокрема такими як OpenCV. Середовище дозволяє легко налаштовувати

залежності, керувати шляхами підключення заголовкових файлів і бібліотек, а також компілювати проєкти з використанням різних конфігурацій. Це спрощує інтеграцію алгоритмів обробки зображень, нейронних мереж і модулів апаратного прискорення [18], [19].

Засоби налагодження, що входять до складу Visual Studio, забезпечують покрокове виконання коду, перегляд значень змінних у реальному часі, аналіз стеку викликів та відстеження винятків. У складних системах обробки відеопотоку такі інструменти дозволяють оперативно локалізувати логічні та ресурсні помилки, що виникають під час роботи з великими масивами даних [18], [19].

Обґрунтування вибору середовища розробки для створення системи опрацювання даних для розпізнавання автомобільних номерів наведено у таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору середовища розробки для створення системи опрацювання даних для розпізнавання автомобільних номерів

Критерій порівняння середовищ розробки	Середовища розробки		
	Visual Studio	CLion	Qt Creator
Зручність інтеграції бібліотек	+	+	+–
Інструменти профілювання продуктивності	+	+–	+–
Інтеграція з системами контролю версій	+	+	+–
Документація та професійна підтримка	+	+–	+–
Інтуїтивність інтерфейсу	+	+	+–

Середовище Visual Studio підтримує системи контролю версій, можливість командної розробки та розширюваність через плагіни. Це створює умови для масштабування проєкту та його подальшої модернізації. Високий рівень стабільності та регулярні оновлення середовища забезпечують сумісність із сучасними стандартами мови та новими технологічними рішеннями [18], [19].

Таким чином, для створення системи опрацювання даних для розпізнавання автомобільних номерів обрано середовище розробки Visual Studio, що зумовлюється поєднанням функціональності, стабільності та високого рівня інтеграції з інструментами, необхідними для реалізації складних алгоритмів комп’ютерного зору. Крім того, Visual Studio надає зручні засоби для налаштування бібліотек, організації багатофайлової структури та забезпечення ефективного налагодження.

2.3 Висновки за розділом 2

Для реалізації системи опрацювання даних для розпізнавання автомобільних номерів обрано мову програмування C++, що обумовлюється поєднанням високої продуктивності, гнучкості, широких можливостей оптимізації та наявності спеціалізованих інструментів, що відповідають вимогам до реального часу, масштабованості та надійності функціонування.

Для створення системи опрацювання даних для розпізнавання автомобільних номерів обрано середовище розробки Visual Studio, що зумовлюється поєднанням функціональності, стабільності та високого рівня інтеграції з інструментами, необхідними для реалізації складних алгоритмів комп’ютерного зору. Крім того, Visual Studio надає зручні засоби для налаштування бібліотек, організації багатофайлової структури та забезпечення ефективного налагодження.

3 ОПИС СИСТЕМИ ОПРАЦЮВАННЯ ДАНИХ

3.1 Структура системи опрацювання даних для розпізнавання автомобільних номерів

Структуру системи опрацювання даних для розпізнавання автомобільних номерів наведено у вигляді діаграми класів на рис. 3.1.

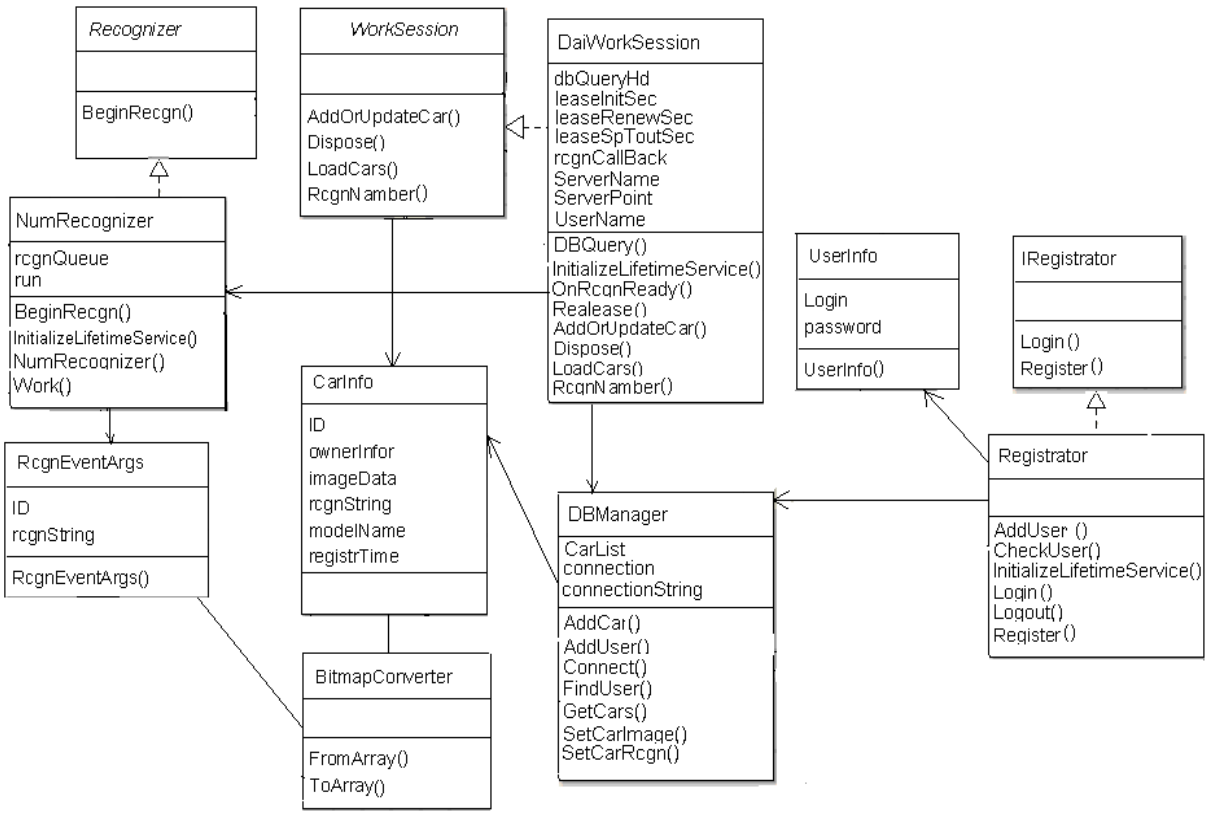


Рисунок 3.1 – Структура системи опрацювання даних для розпізнавання автомобільних номерів у вигляді діаграми класів

Структурна організація системи опрацювання даних для розпізнавання автомобільних номерів подана у вигляді діаграми класів, що відображає взаємозв’язки між основними програмними компонентами та розподіл відповідальності між ними. Архітектура побудована за клієнт-серверним принципом, де центральне місце посідає серверна частина, яка забезпечує приймання запитів, їх обробку, взаємодію з базою даних і повернення результатів клієнтам.

Взаємодія користувача із системою здійснюється через клас WorkSession, який реалізує механізм робочої сесії. Через цей інтерфейс ініціюється передавання зображень на сервер, отримання результатів розпізнавання та обмін службовими повідомленнями. Оскільки передбачено віддалений доступ, відповідний клас підтримує мережеву взаємодію та забезпечує ізоляцію клієнтської логіки від внутрішніх механізмів серверної обробки. Таким чином формується рівень представлення, який не містить алгоритмів розпізнавання, а лише координує обмін даними.

Функціонування механізму розпізнавання покладено на клас Recognizer, який надає інтерфейс до сервера обробки зображень. Сам сервер реалізовано у вигляді програмного продукту RecognitionServer, що інкапсулює логіку розпізнавання в класі NumRecognizer. Останній є конкретною реалізацією абстрактного базового класу BaseServices.Recognizer, завдяки чому забезпечується можливість розширення або модифікації алгоритмів без зміни загальної структури системи. Після надходження запиту зображення не обробляється негайно, а поміщається у внутрішню чергу. Це пояснюється тим, що одночасна обробка декількох зображень не передбачається, тому використовується механізм послідовного виконання. Робочий потік сервера вибирає елементи з черги, здійснює збереження зображення у визначеному каталозі та ініціює запуск модуля розпізнавання. Після завершення обчислень отриманий текстовий результат передається клієнтові через відповідний інтерфейс.

Питання автентифікації та контролю доступу покладено на інтерфейс IRegistator, який забезпечує перевірку облікових даних та реєстрацію нових користувачів у базі даних. Через цей компонент реалізується перевірка логіна й пароля, що дозволяє обмежити доступ до функціоналу системи та підтримувати безпечну роботу в мережевому середовищі. Дані користувачів інкапсулюються у класі UserInfo, який зберігає відповідні атрибути та використовується для взаємодії з підсистемою авторизації.

Інформація про транспортний засіб представлена класом CarInfo, у якому акумулюються відомості про власника, марку автомобіля та результат розпізнавання номерного знака. Цей клас використовується для формування записів у базі даних і для передачі структурованої інформації клієнтським застосункам. Таким чином забезпечується логічне відокремлення даних від алгоритмічної частини.

Окрему роль у структурі відіграє клас BitmapConverter, призначений для трансформації об’єкта зображення у масив бітів. Такий підхід дозволяє уніфікувати формат передавання графічної інформації, спростити її серіалізацію та забезпечити коректну передачу мережею. Конвертація зображень у стандартизоване подання є важливою умовою стабільної взаємодії між клієнтською та серверною частинами системи.

Узгоджена робота зазначених класів формує цілісну багаторівневу структуру, де відокремлено інтерфейси взаємодії, механізми безпеки, обробку зображень і збереження результатів. Така організація сприяє модульності, спрощує супровід і створює передумови для подальшого розширення функціональних можливостей.

Отже, описана структура програми відзначається чітким розподілом обов’язків між класами, використанням абстракцій та серверної черги запитів, що забезпечує впорядковану й надійну обробку даних у системі розпізнавання автомобільних номерів.

3.2 Функціонування системи опрацювання даних для розпізнавання автомобільних номерів

Функціонування системи опрацювання даних для розпізнавання автомобільних номерів подамо за схеми, зображеної на рис. 3.2.

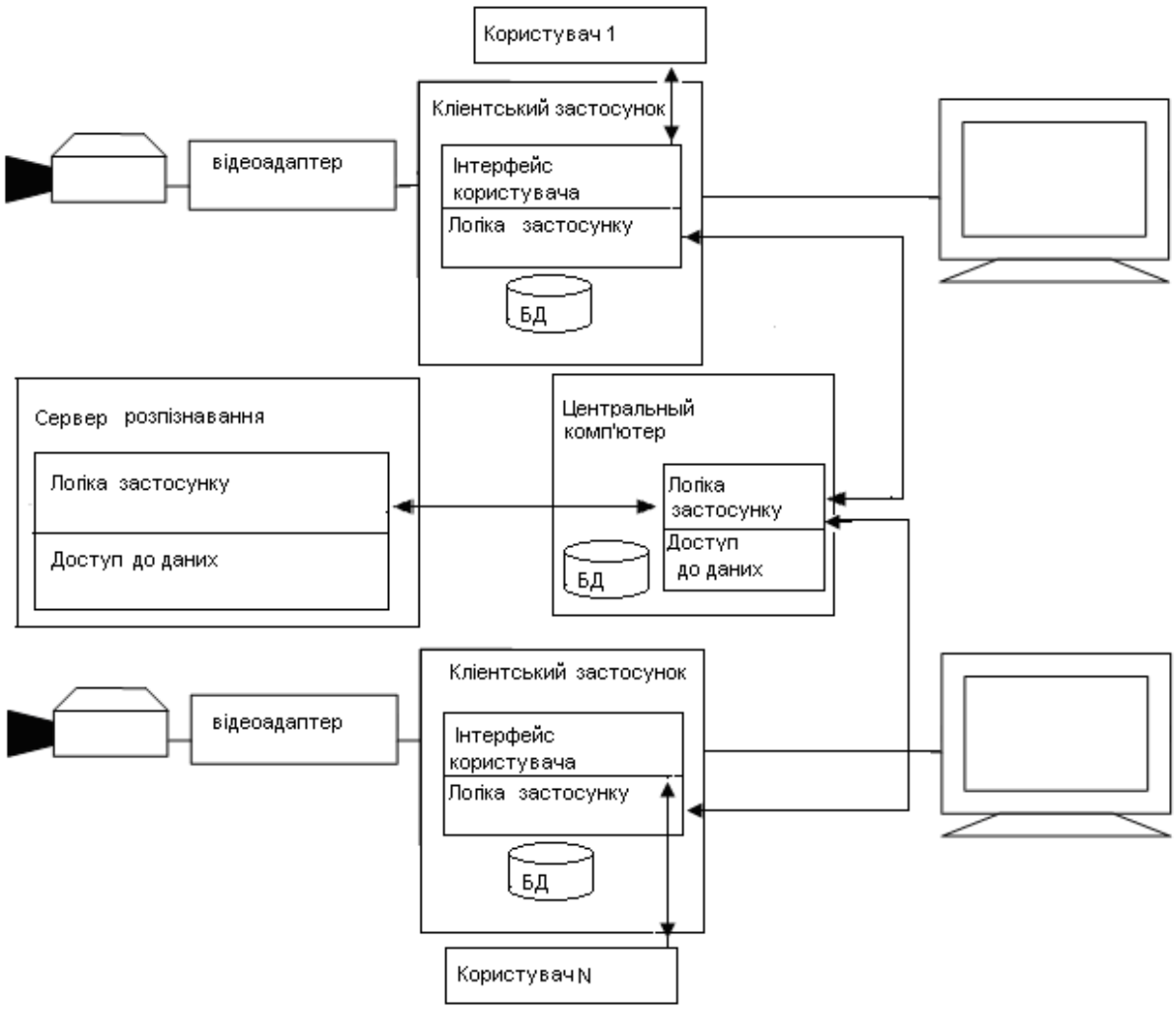


Рисунок 3.2 – Схема функціонування системи опрацювання даних для розпізнавання автомобільних номерів

Схема функціонування системи опрацювання даних для розпізнавання автомобільних номерів відображає багаторівневу організацію взаємодії між користувачами, клієнтськими застосунками, центральним обчислювальним вузлом і сервером розпізнавання. У межах такої архітектури забезпечується розподіл відповідальності між рівнем представлення, координаційним рівнем і спеціалізованим модулем обробки зображень, що дозволяє досягти узгодженості процесів і стабільності роботи системи в умовах одночасного підключення кількох клієнтів.

Взаємодія розпочинається на рівні користувача, який через двонаправлений канал зв’язку працює з клієнтським застосунком. Такий формат взаємодії передбачає не лише передавання команд і зображень до

системи, а й отримання результатів розпізнавання, повідомлень про стан обробки та інших службових даних. Усередині клієнтського застосунку виділяється інтерфейс користувача, що відповідає за відображення інформації та приймання вхідних дій. Логіка застосунку виконує попередню обробку запитів, формує структури даних і забезпечує коректне передавання інформації на серверний рівень. Наявність локальної бази даних дозволяє зберігати службові записи, історію звернень або параметри конфігурації, що підвищує автономність клієнтського модуля.

Після формування запиту дані передаються до центрального комп’ютера, який виконує роль координаційного ядра системи. У його межах функціонує логіка доступу, що контролює автентифікацію, маршрутизацію запитів і розмежування прав. Компонент доступу до даних забезпечує взаємодію з центральною базою даних, у якій зберігаються відомості про користувачів, результати розпізнавання та службова інформація. Таким чином формується проміжний рівень, який узгоджує дії клієнтів і керує передаванням інформації до спеціалізованого модуля розпізнавання.

Центральний комп’ютер взаємодіє із сервером розпізнавання, що виконує безпосередню обробку зображень. У структурі цього сервера виділяється логіка застосунку, яка реалізує алгоритми аналізу зображення, виділення області номерного знака та розпізнавання символів. Компонент доступу до даних на цьому рівні відповідає за збереження проміжних результатів і передачу підсумкової інформації назад до центрального вузла. Після завершення розпізнавання результат повертається через центральний комп’ютер до відповідного клієнтського застосунку, а звідти — до користувача.

Особливістю схеми є наявність кількох користувачів і кількох клієнтських застосунків, що одночасно взаємодіють із центральним вузлом. Така організація передбачає підтримку багатокористувацького режиму, забезпечення синхронізації доступу до ресурсів і стабільне функціонування за умов паралельних запитів. Розмежування обчислювальних функцій між

рівнями дозволяє оптимізувати навантаження, ізолювати обробку зображень від клієнтської частини та підвищити безпеку даних.

Загалом функціональна схема демонструє послідовний рух інформації від користувача до спеціалізованого модуля розпізнавання й у зворотному напрямку, із чітким розподілом ролей між компонентами. Такий підхід забезпечує масштабованість, керованість і надійність роботи системи в умовах одночасного доступу багатьох клієнтів.

3.3 Розробка бази даних системи опрацювання даних для розпізнавання автомобільних номерів

База даних системи опрацювання даних для розпізнавання автомобільних номерів забезпечує збереження результатів обробки, облік користувачів та цілісність інформації. Логічна модель бази даних побудована на основі чотирьох основних таблиць, які охоплюють користувачів, транспортні засоби, результати розпізнавання та журнали запитів.

Схему бази даних системи опрацювання даних для розпізнавання автомобільних номерів наведено на рис. 3.3.

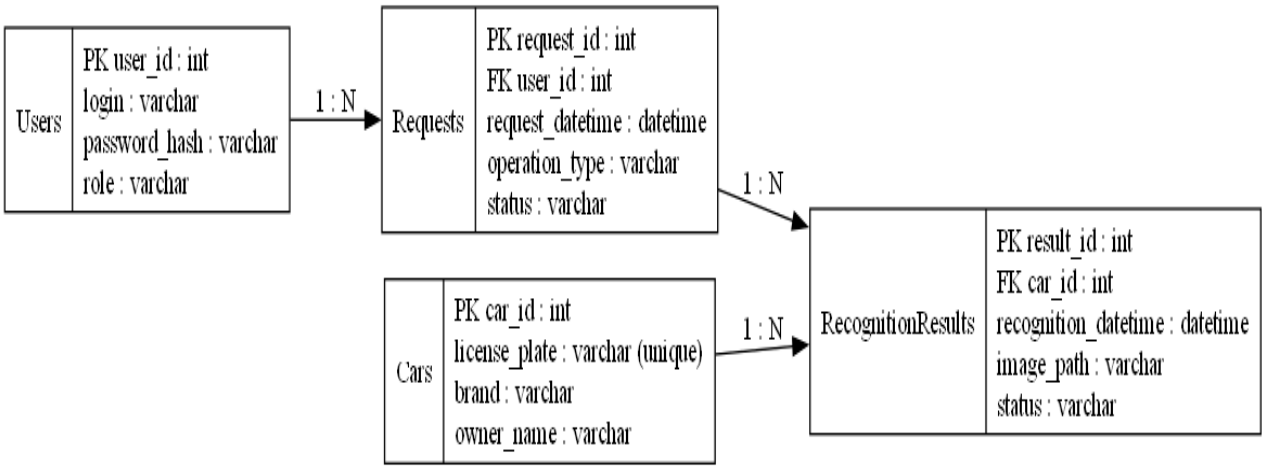


Рисунок 3.3 – Схема бази даних системи опрацювання даних для розпізнавання автомобільних номерів

Центральною сутністю бази даних є таблиця Users, у якій зберігаються ідентифікаційні дані користувачів системи. Для кожного запису передбачається первинний ключ, логін, хеш пароля та службові атрибути, що визначають роль або рівень доступу. Така структура дозволяє реалізувати механізми автентифікації та розмежування прав. З таблицею Users встановлюється зв’язок з іншими сутностями через зовнішні ключі, що забезпечує контроль цілісності.

Інформація про транспортні засоби концентрується в таблиці Cars. У цій таблиці зберігається унікальний ідентифікатор автомобіля, розпізнаний номерний знак, марка та, за потреби, дані про власника. Поле з номерним знаком зроблено унікальним для запобігання дублюванню записів. Таблиця Cars пов’язується з таблицею результатів розпізнавання через відношення «один до багатьох», оскільки один автомобіль може фіксуватися системою неодноразово.

Фактичні результати обробки зображень зберігаються в таблиці RecognitionResults. Для кожного запису передбачається власний первинний ключ, посилання на автомобіль, дата й час фіксації, шлях до збереженого зображення або його ідентифікатор у файловій системі, а також службові атрибути, що характеризують якість або статус розпізнавання. Наявність зовнішнього ключа на таблицю Cars забезпечує узгодженість між зафіксованим номером і конкретним транспортним засобом. Таким чином реалізується історія спостережень, що може використовуватися для аналітики або аудиту.

Для відображення факту звернення користувача до системи формується таблиця Requests. У ній фіксується ідентифікатор запиту, посилання на користувача, дата й час ініціювання, тип операції та статус виконання. Зв’язок між Users і Requests визначається як «один до багатьох», оскільки один користувач може створювати численні запити. Така таблиця виконує функцію журналу подій і дозволяє відстежувати активність у системі.

Між таблицями реалізуються чіткі логічні зв’язки через первинні та зовнішні ключі, що забезпечує референтну цілісність. Схема характеризується відсутністю надлишковості даних, оскільки інформація про користувачів, автомобілі та результати зберігається окремо, а взаємозв’язки встановлюються через ідентифікатори. Подібна модель є достатньою для невеликої системи, але водночас передбачає можливість розширення шляхом додавання нових атрибутів або сутностей без суттєвої перебудови структури.

Отже, логічна модель бази даних системи розпізнавання автомобільних номерів побудована на основі чотирьох основних таблиць, які охоплюють користувачів, транспортні засоби, результати розпізнавання та журнали запитів, що забезпечує збереження інформації про ключові сутності системи, підтримку історії розпізнавань та контроль доступу користувачів. Чітко визначені зв’язки між таблицями створюють основу для стабільного функціонування системи та подальшого масштабування.

3.4 Проєктування інтерфейсу системи опрацювання даних для розпізнавання автомобільних номерів

Проєктування інтерфейсу взаємодії користувача з системою опрацювання даних для розпізнавання автомобільних номерів виконано з урахуванням вимог технічного завдання. При розробленні системи опрацювання даних для розпізнавання автомобільних номерів враховані функціональні вимоги до програми:

– підтримка можливості завантаження відеопотоків для подальшого розпізнавання автомобільних номерів;

– підтримка збереження результатів розпізнавання у базі даних;

– підтримка можливості перегляду історії виконаних запитів і отриманих результатів розпізнавання;

– можливість авторизації користувачів системи;

– забезпечення коректної роботи системи в умовах різної якості зображень та змін освітлення;

– підтримка можливості отримання інформації про ймовірність правильного розпізнавання автомобільного номеру.

Інтерфейс головної форми системи опрацювання даних для розпізнавання автомобільних номерів наведено на рис. 3.4.

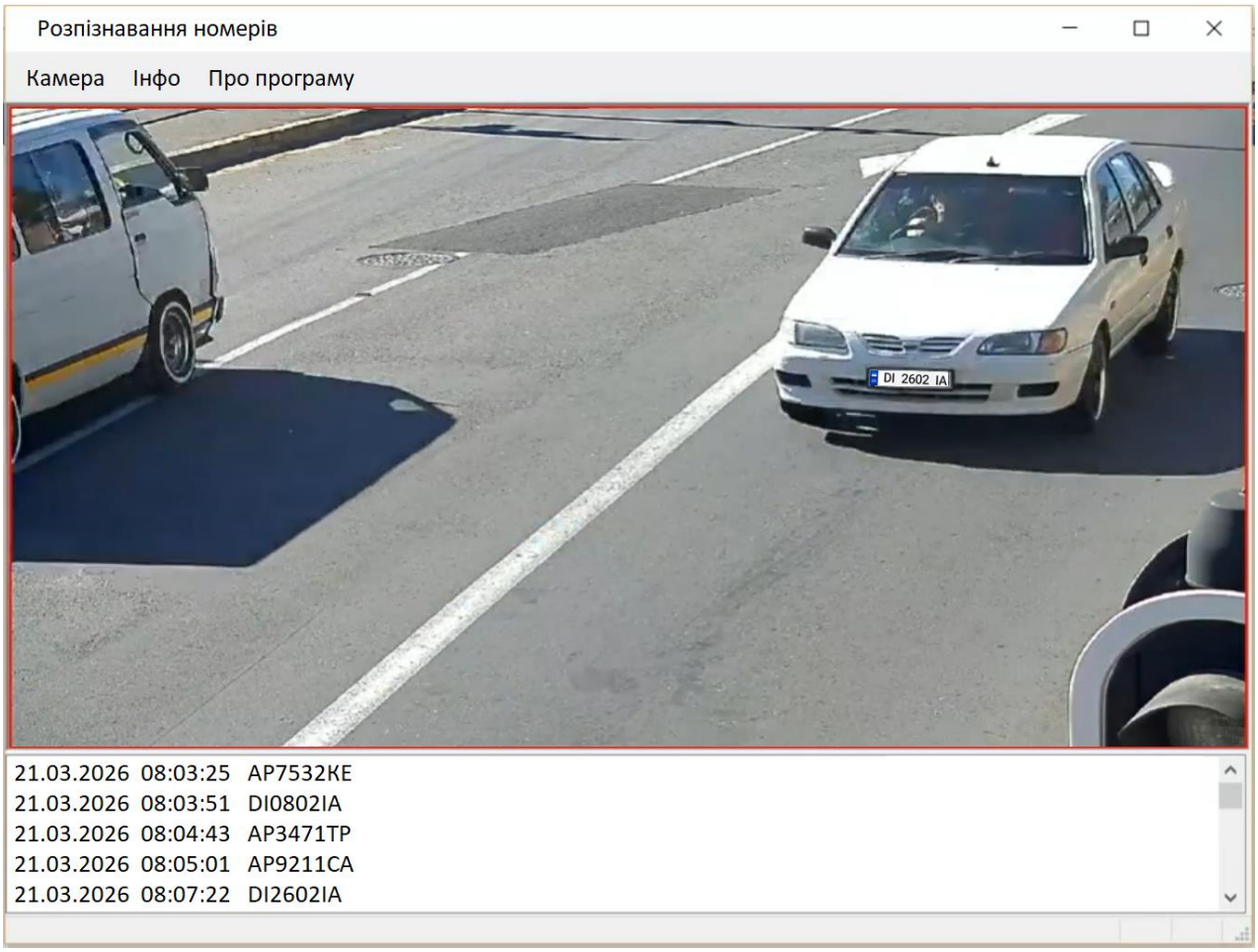


Рисунок 3.4 – Інтерфейс головної форми системи опрацювання даних для розпізнавання автомобільних номерів

Інтерфейс сторінки з результатами розпізнавання автомобільних номерів наведено на рис. 3.5.

Інтерфейс сторінки авторизації наведено на рис. 3.6.

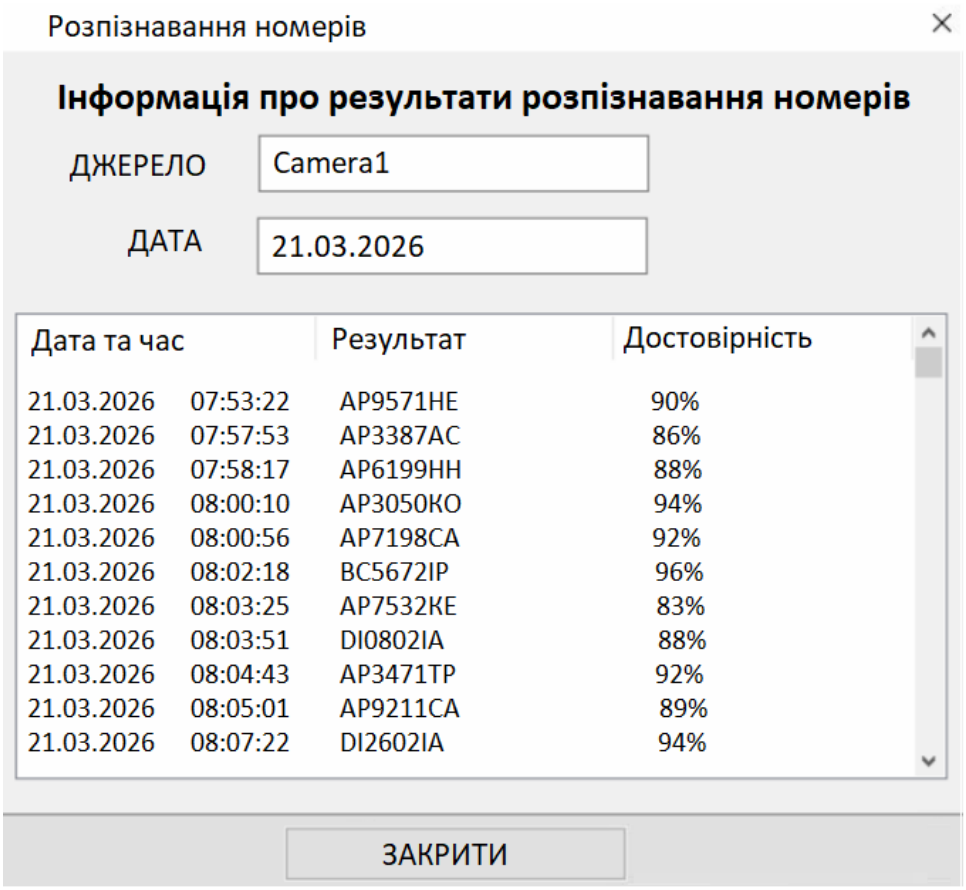


Рисунок 3.5 – Інтерфейс сторінки з результатами розпізнавання автомобільних номерів

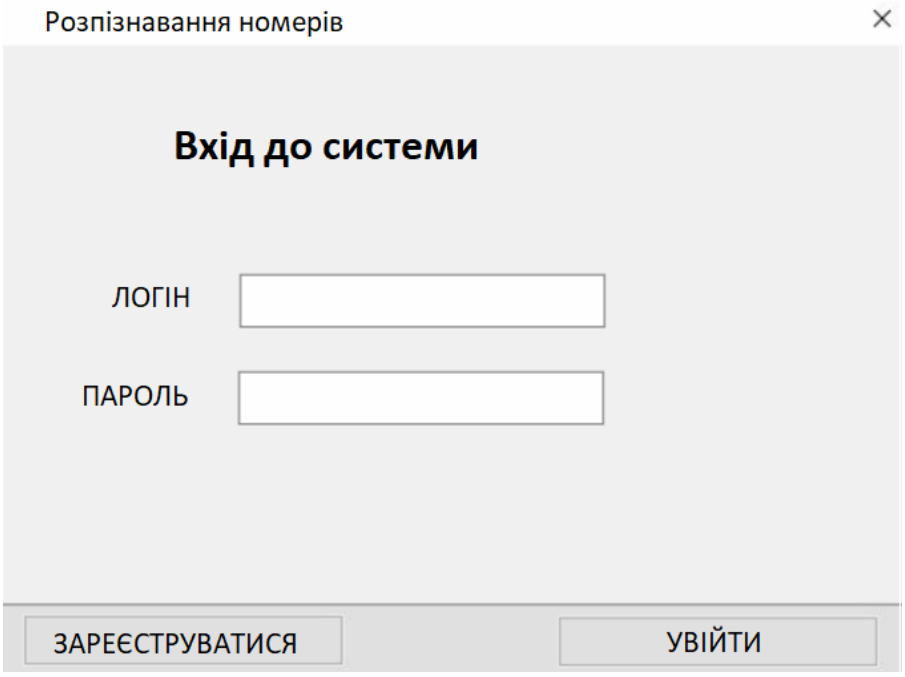


Рисунок 3.6 – Інтерфейс сторінки авторизації

3.5 Висновки за розділом 3

Розроблено систему опрацювання даних для розпізнавання автомобільних номерів.

Структурну організацію системи опрацювання даних для розпізнавання автомобільних номерів подано у вигляді діаграми класів, що відображає взаємозв’язки між основними програмними компонентами та розподіл відповідальності між ними. Архітектура побудована за клієнт-серверним принципом, де центральне місце посідає серверна частина, яка забезпечує приймання запитів, їх обробку, взаємодію з базою даних і повернення результатів клієнтам.

Описано функціонування системи опрацювання даних для розпізнавання автомобільних номерів. Схема функціонування системи опрацювання даних для розпізнавання автомобільних номерів відображає багаторівневу організацію взаємодії між користувачами, клієнтськими застосунками, центральним обчислювальним вузлом і сервером розпізнавання. У межах такої архітектури забезпечується розподіл відповідальності між рівнем представлення, координаційним рівнем і спеціалізованим модулем обробки зображень, що дозволяє досягти узгодженості процесів і стабільності роботи системи в умовах одночасного підключення кількох клієнтів.

Розроблено базу даних системи опрацювання даних для розпізнавання автомобільних номерів, яка побудована на основі чотирьох основних таблиць, які охоплюють користувачів, транспортні засоби, результати розпізнавання та журнали запитів, що забезпечує збереження інформації про ключові сутності системи, підтримку історії розпізнавань та контроль доступу користувачів. Чітко визначені зв’язки між таблицями створюють основу для стабільного функціонування системи та подальшого масштабування.

Виконано проєктування інтерфейсу взаємодії користувача з системою опрацювання даних для розпізнавання автомобільних номерів.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ СИСТЕМИ ОПРАЦЮВАННЯ ДАНИХ

4.1 Призначення й умови застосування системи опрацювання даних

Система опрацювання даних призначена для підтримки процесу розпізнавання автомобільних номерів. Система опрацювання даних написана на мові C++, що обумовлюється поєднанням високої продуктивності, гнучкості, широких можливостей оптимізації та наявності спеціалізованих інструментів, що відповідають вимогам до реального часу, масштабованості та надійності функціонування. В якості середовища розробки для створення системи опрацювання даних для розпізнавання автомобільних номерів обрано Visual Studio, що зумовлюється поєднанням функціональності, стабільності та високого рівня інтеграції з інструментами, необхідними для реалізації складних алгоритмів комп’ютерного зору.

Система опрацювання даних для розпізнавання автомобільних номерів забезпечує виконання таких функцій:

– підтримка можливості завантаження відеопотоків для подальшого розпізнавання автомобільних номерів;

– підтримка збереження результатів розпізнавання у базі даних;

– підтримка можливості перегляду історії виконаних запитів і отриманих результатів розпізнавання;

– можливість авторизації користувачів системи;

– забезпечення коректної роботи системи в умовах різної якості зображень та змін освітлення;

– підтримка можливості отримання інформації про ймовірність правильного розпізнавання автомобільного номеру.

Для роботи системи опрацювання даних для розпізнавання автомобільних номерів потрібні такі технічні та програмні ресурси: апаратна платформа з процесором з тактовою частотою не нижче 3,0 ГГц із підтримкою

багатопотокової обробки та наявністю щонайменше чотирьох фізичних ядер. Для стабільної роботи серверної частини доцільно застосовувати процесори класу Intel Core i5 / i7 або аналогічні за продуктивністю моделі AMD Ryzen. У випадку інтенсивної експлуатації або одночасного обслуговування декількох клієнтів бажаною є наявність шести чи восьми ядер.

Обсяг оперативної пам’яті повинен становити не менше 8 ГБ для базової конфігурації. Для серверної інсталяції або роботи з великими потоками відеоданих рекомендується 16 ГБ і більше. Для зберігання зображень і результатів розпізнавання необхідний накопичувач обсягом не менше 256 ГБ.

Програмне забезпечення повинно передбачати встановлення операційної системи сімейства Windows або Linux із підтримкою 64-бітної архітектури. Для компіляції та супроводу програмного коду доцільним є використання інтегрованого середовища розробки Microsoft Visual Studio з підтримкою стандартів C++ не нижче C++17.

Для функціонування клієнтської частини достатньо комп’ютера з процесором не нижче 2,0 ГГц, оперативною пам’яттю від 4 ГБ та встановленою операційною системою, сумісною із серверною частиною.

Функціональні характеристики розробленої системи опрацювання даних для розпізнавання автомобільних номерів наведено у технічному завданні (додаток А). Фрагмент тексту програмного коду системи опрацювання даних для розпізнавання автомобільних номерів наведено у додатку Б.

4.2 Характеристики системи опрацювання даних для розпізнавання автомобільних номерів

Система опрацювання даних для розпізнавання автомобільних номерів характеризується комплексністю функціонування та орієнтацією на автоматизацію процесів ідентифікації транспортних засобів за зображеннями

або відеопотоком. У її основі передбачено використання алгоритмів комп’ютерного зору, які забезпечують виявлення номерної області на зображенні, виділення символів та їх подальше перетворення у текстове представлення. Обробка даних здійснюється з урахуванням можливих шумів, змін освітлення, різних ракурсів зйомки та часткових перекриттів номерного знака.

Для системи притаманна модульна структура, що передбачає відокремлення підсистем збору даних, їх аналізу, збереження результатів та формування звітної інформації. Такий підхід забезпечує гнучкість налаштування та можливість адаптації до різних умов експлуатації. Передбачається підтримка інтеграції з зовнішніми інформаційними ресурсами, що дозволяє розширювати функціональні можливості та застосовувати систему у межах більших програмно-апаратних комплексів.

Інтерфейс користувача орієнтовано на наочне відображення результатів розпізнавання, включаючи візуалізацію вихідного зображення та текстового представлення номерного знака. Забезпечується розмежування прав доступу до функціональних можливостей відповідно до ролей користувачів, що підвищує рівень інформаційної безпеки. Передбачено можливість журналювання дій, що сприяє контролю за використанням системи та забезпечує прозорість її роботи.

У цілому система відзначається надійністю, масштабованістю та можливістю адаптації до різних сфер застосування, включаючи контроль доступу, моніторинг транспортних потоків і автоматизацію облікових процесів.

4.3 Інструкція по експлуатації програми

4.3.1 Звернення до програми

Для запуску системи опрацювання даних для розпізнавання автомобільних номерів на серверній частини необхідно запустити відповідний

файл. Для звертання до клієнтської сторони необхідно перейти на відповідну вебсторінку.

4.3.2 Вхідні й вихідні дані

Вхідними даними до системи опрацювання даних для розпізнавання автомобільних номерів є цифрові зображення автомобілів, включно з номерними знаками.

Вихідними даними системи опрацювання даних для розпізнавання автомобільних номерів є текстове представлення розпізнаного номерного знака, інформація про ймовірність правильно розпізнавання та інша інформація.

4.3.3 Повідомлення

Користувач системи опрацювання даних для розпізнавання автомобільних номерів може отримати такі повідомлення: повідомлення про успішне завантаження потоку зображень, попередження про неможливість чіткого визначення номерного знака через низьку якість зображення або невідповідність формату даних. Крім того, може відображатися інформація про внутрішні помилки обробки або рекомендації щодо повторного виконання операції з іншими параметрами.

4.4 Виконання системи опрацювання даних для розпізнавання автомобільних номерів

Після запуску системи опрацювання даних для розпізнавання автомобільних номерів користувачу відображається сторінка входу до програми (рис. 4.1).

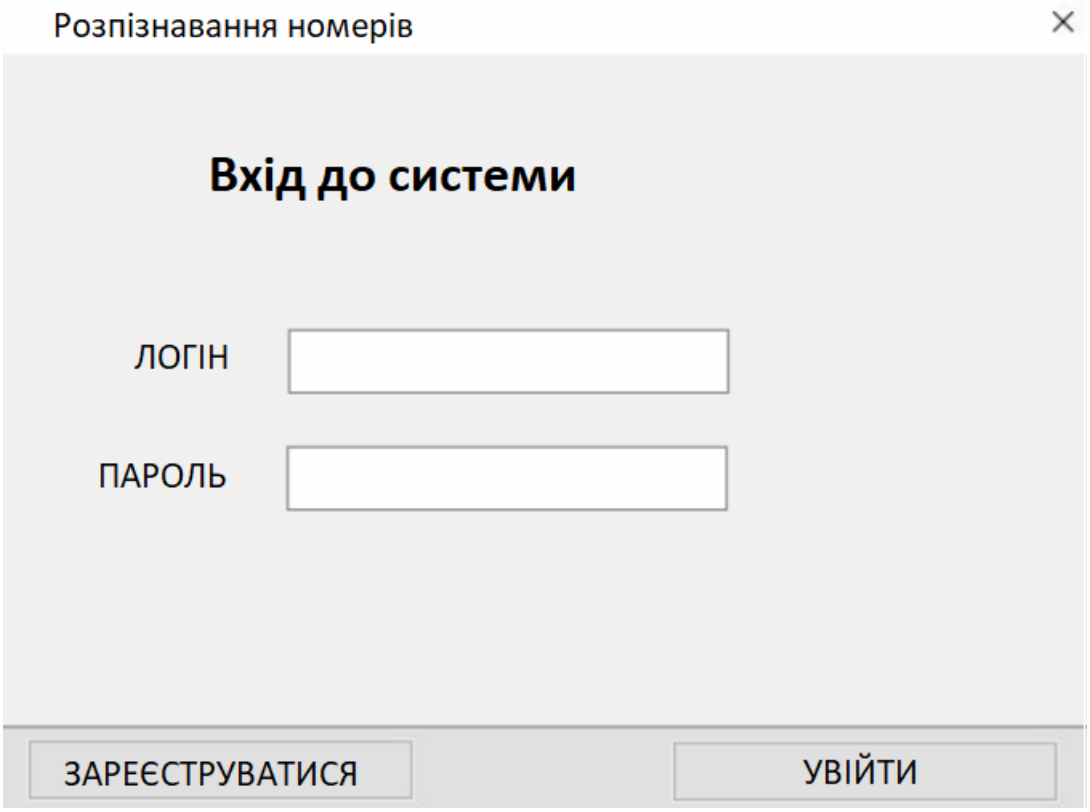


Рисунок 4.1 – Інтерфейс форми авторизації у системі опрацювання даних для розпізнавання автомобільних номерів

Після авторизації відображається головна сторінка (рис. 4.2) з результатами розпізнавання автомобільних номерів з поточного джерела надходження зображень (наприклад, з попередньо обраної камери). Користувачу надається зображення, а також в окремому полі історія розпізнавання автомобільних номерів на зображенні з датою, часом та результатом розпізнавання.

За необхідності користувач може змінити джерело надходження зображень з автотранспортними засобами (рис. 4.3), для чого необхідно ввести шлях до камери або відеопотоку та, за потреби, пароль доступу.

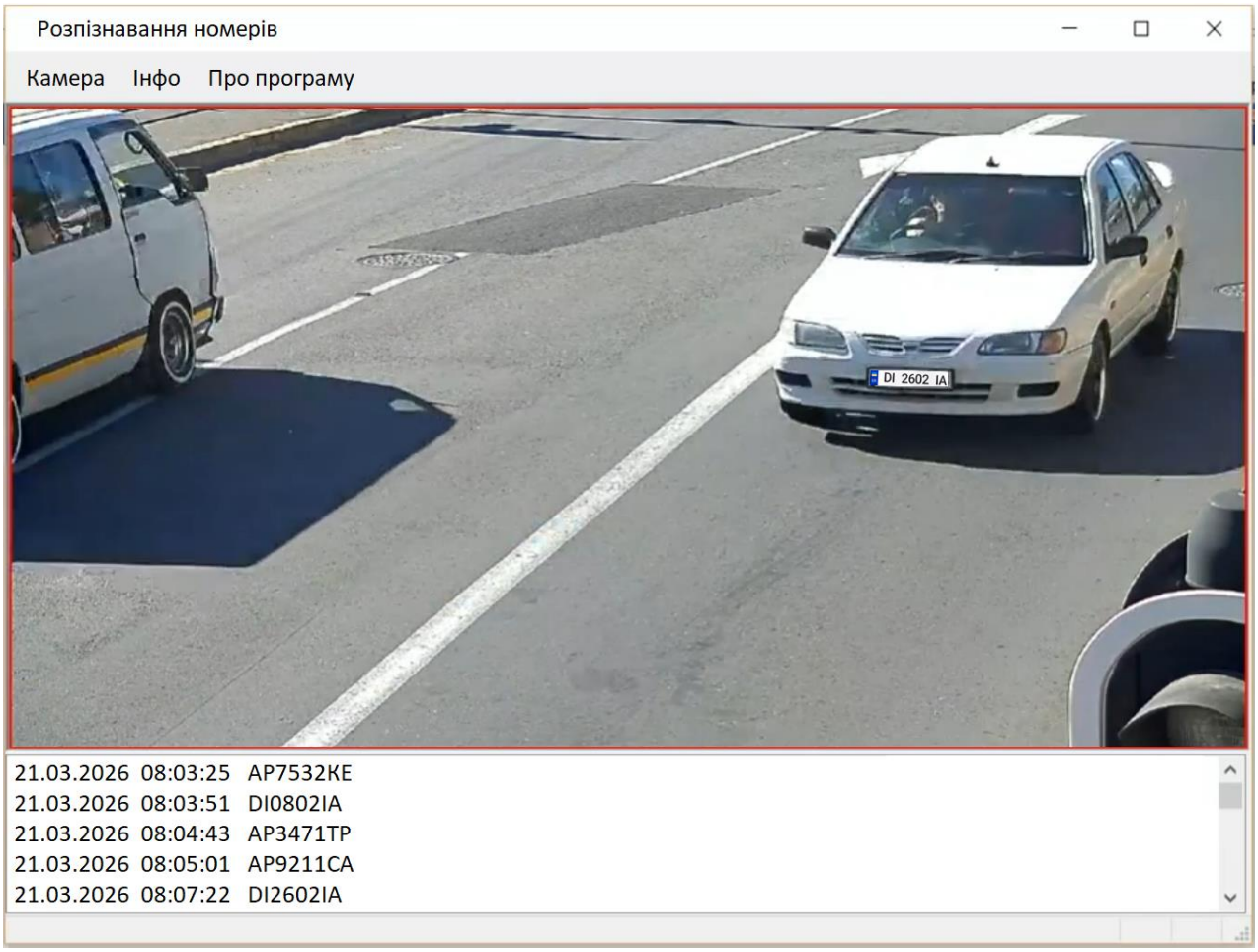


Рисунок 4.2 – Розпізнавання номерних знаків автомобілів

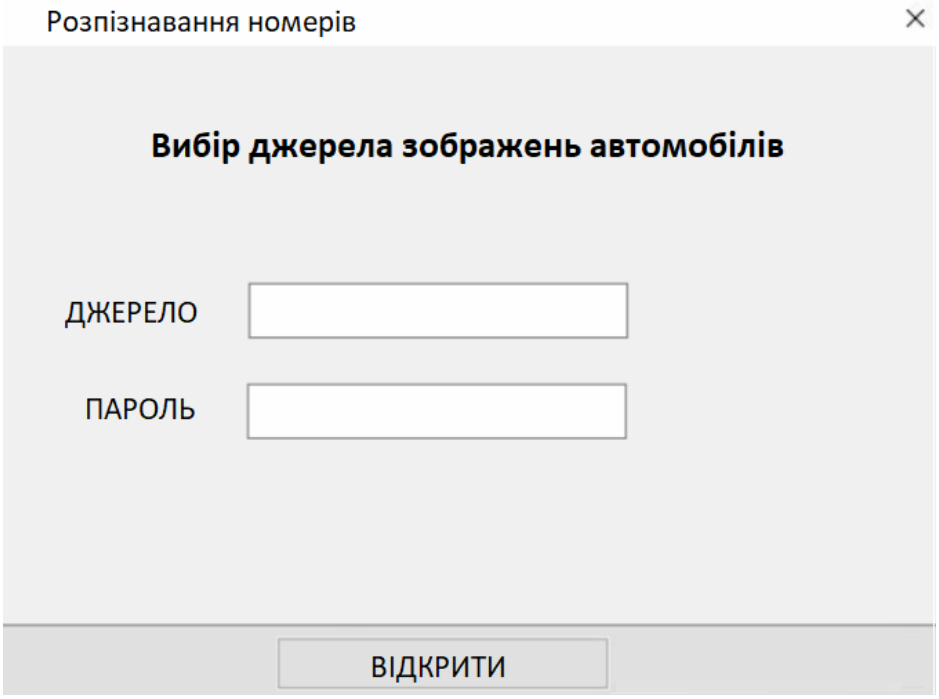


Рисунок 4.3 – Вибір джерела зображень автотранспортних засобів

За допомогою пункту меню Інфо (рис. 4.4) користувач може отримати інформацію (зокрема, і ймовірність правильного прогнозу) про результати розпізнавання номерів автомобілів на обрану дату.

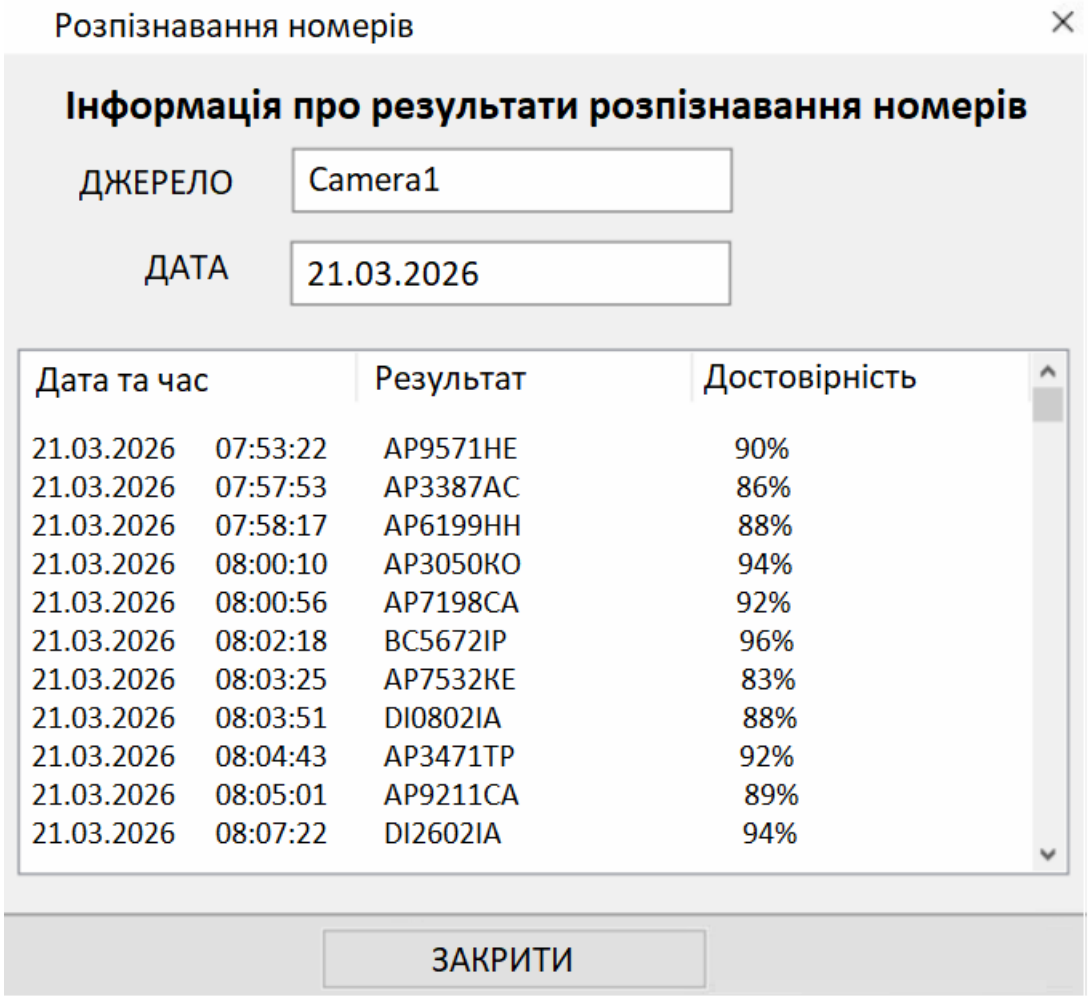


Рисунок 4.4 – Інформація про результати розпізнавання автомобільних номерів

4.5 Тестування системи опрацювання даних для розпізнавання автомобільних номерів

Проведено тестування системи опрацювання даних для розпізнавання автомобільних номерів.

Підтверджено, що системи опрацювання даних для розпізнавання автомобільних номерів працює коректно і стабільно, виконуючи всі необхідні

функціональні вимоги та успішно справляючись із основною задачею підтримки процесу взаємодії користувачів.

Розроблена система забезпечує організацію та опрацювання даних, пов’язаних із підтримкою процесу розпізнавання автомобільних номерів.

4.6 Висновки за розділом 4

Описано систему опрацювання даних для розпізнавання автомобільних номерів.

Проведено тестування створеної системи опрацювання даних для розпізнавання автомобільних номерів. Результати тестування показали, що система забезпечує організацію та опрацювання даних, пов’язаних із підтримкою процесу розпізнавання автомобільних номерів.

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи бакалавра було проаналізовано та досліджено процеси обчислень, пов’язані з організацією та опрацюванням даних для розпізнавання автомобільних номерів.

Визначено, що система опрацювання даних для розпізнавання автомобільних номерів являє собою програмно-апаратний комплекс, призначений для автоматичного виявлення, зчитування та інтерпретації символів номерних знаків транспортних засобів. У її основі лежать алгоритми комп’ютерного зору, машинного навчання та аналізу зображень, за допомогою яких здійснюється обробка відеопотоку або статичних кадрів, виділення області номерного знака та перетворення графічної інформації на текстові дані. Отримані результати передаються до інформаційних систем для подальшого зберігання, пошуку або аналітичної обробки.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато систем опрацювання даних для розпізнавання автомобільних номерів. Проте деякі системи опрацювання даних для розпізнавання автомобільних номерів характеризуються вразливістю до варіативності погодних умов, форматів номерних знаків різних країн або регіонів у межах однієї держави. Необхідність постійного оновлення баз даних, адаптації алгоритмів до нових стандартів та перенавчання моделей створює додаткове навантаження на розробників і операторів таких програмних систем. Високі вимоги до обчислювальних ресурсів, особливо при обробці потокового відео в режимі реального часу, зумовлюють потребу у продуктивному обладнанні та масштабованій інфраструктурі, що підвищує загальну вартість впровадження й експлуатації. Тому актуальною є розробка системи опрацювання даних для розпізнавання автомобільних номерів.

Сформульовано функціональні вимоги до системи опрацювання даних для розпізнавання автомобільних номерів.

Для реалізації системи опрацювання даних для розпізнавання автомобільних номерів обрано мову програмування C++, що обумовлюється поєднанням високої продуктивності, гнучкості, широких можливостей оптимізації та наявності спеціалізованих інструментів, що відповідають вимогам до реального часу, масштабованості та надійності функціонування.

Для створення системи опрацювання даних для розпізнавання автомобільних номерів обрано середовище розробки Visual Studio, що зумовлюється поєднанням функціональності, стабільності та високого рівня інтеграції з інструментами, необхідними для реалізації складних алгоритмів комп'ютерного зору. Крім того, Visual Studio надає зручні засоби для налаштування бібліотек, організації багатофайлової структури та забезпечення ефективного налагодження.

Розроблено систему опрацювання даних для розпізнавання автомобільних номерів.

Структурну організацію системи опрацювання даних для розпізнавання автомобільних номерів подано у вигляді діаграми класів, що відображає взаємозв'язки між основними програмними компонентами та розподіл відповідальності між ними. Архітектура побудована за клієнт-серверним принципом, де центральне місце посідає серверна частина, яка забезпечує приймання запитів, їх обробку, взаємодію з базою даних і повернення результатів клієнтам.

Описано функціонування системи опрацювання даних для розпізнавання автомобільних номерів. Схема функціонування системи опрацювання даних для розпізнавання автомобільних номерів відображає багаторівневу організацію взаємодії між користувачами, клієнтськими застосунками, центральним обчислювальним вузлом і сервером розпізнавання. У межах такої архітектури забезпечується розподіл відповідальності між рівнем представлення, координаційним рівнем і спеціалізованим модулем обробки зображень, що дозволяє досягти узгодженості процесів і стабільності роботи системи в умовах одночасного підключення кількох клієнтів

Розроблено базу даних системи опрацювання даних для розпізнавання автомобільних номерів, яка побудована на основі чотирьох основних таблиць, які охоплюють користувачів, транспортні засоби, результати розпізнавання та журнали запитів, що забезпечує збереження інформації про ключові сутності системи, підтримку історії розпізнавань та контроль доступу користувачів. Чітко визначені зв’язки між таблицями створюють основу для стабільного функціонування системи та подальшого масштабування.

Виконано проєктування інтерфейсу взаємодії користувача з системою опрацювання даних для розпізнавання автомобільних номерів.

Проведено тестування створеної системи опрацювання даних для розпізнавання автомобільних номерів. Результати тестування показали, що система забезпечує організацію та опрацювання даних, пов’язаних із підтримкою процесу розпізнавання автомобільних номерів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. The Basics of Automatic License Plate Recognition. URL: https://talkmarkets.com/article/the-basics-of-automatic-license-plate-recognition (date of access: 04.05.2026).

2. What is License Plate Recognition Software? A Beginner's Guide. URL: https://www.vaxtor.com/news/what-is-license-plate-recognition-software-a-beginners-guide/ (date of access: 04.05.2026).

3. Automatic Number Plate Recognition (ANPR) Guide. URL: https://viso.ai/computer-vision/automatic-number-plate-recognition-anpr/ (date of access: 04.05.2026).

4. How Automated License Plate Recognition Systems Work. URL: https://www.avidbeam.com/how-automated-license-plate-recognition-systems-work/ (date of access: 04.05.2026).

5. How Does ANPR Work? A Comprehensive Guide to Automatic Number Plate Recognition. URL: https://www.carmencloud.com/how-does-anpr-work-a-comprehensive-guide-to-automatic-number-plate-recognition/ (date of access: 04.05.2026).

6. Automatic Number Plate Recognition: A Detailed Survey of Relevant Algorithms. URL: https://www.mdpi.com/1424-8220/21/9/3028 (date of access: 04.05.2026).

7. A Comprehensive Review of License Plate Recognition Techniques. URL: https://www.ijnrd.org/papers/IJNRD2504327.pdf (date of access: 04.05.2026).

8. Complete Guide To License Plate Recognition System. URL: https://ngssolution.com/blogs/license-plate-recognition-system/ (date of access: 04.05.2026).

9. Top 10 Best License Plate Recognition Software. URL: https://gitnux.org/best/license-plate-recognition-software/ (date of access: 04.05.2026).

10. Best Automatic License Plate Recognition Software. URL: https://sourceforge.net/software/automatic-license-plate-recognition-alpr/ (date of access: 04.05.2026).

11. A List of the Top Free ANPR Software Options. URL: https://visualfoodie.com/a-list-of-the-top-free-anpr-software-options/ (date of access: 04.05.2026).

12. The Best Parking ANPR Systems. URL: https://www.folio3.ai/blog/parking-anpr-systems/ (date of access: 04.05.2026).

13. Neurotechnology SentiVeillance. URL: https://www.neurotechnology.com/press_release_sentiveillance_10_0.html (date of access: 04.05.2026).

14. PlateSmart. URL: https://www.platesmart.com/ares-alpr/ (date of access: 04.05.2026).

15. Genetec AutoVu. URL: https://www.genetec.com/products/unified-security/autovu (date of access: 04.05.2026).

16. C++ Tutorial. URL: https://www.w3schools.com/cpp/ (date of access: 04.05.2026).

17. C++ Core Guidelines. URL: https://isocpp.github.io/CppCoreGuidelines/CppCoreGuidelines (date of access: 04.05.2026).

18. Overview of C++ development in Visual Studio. URL: https://learn.microsoft.com/en-us/cpp/overview/overview-of-cpp-development (date of access: 04.05.2026).

19. Powerful IDE, made smarter with AI. URL: https://visualstudio.microsoft.com/vs/ (date of access: 04.05.2026).

ДОДАТОК А

Технічне завдання

Вступ

Програмна система опрацювання даних може використовуватися для підтримки процесу розпізнавання номерів автотранспортних засобів.

А.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Створення програмної системи опрацювання даних для розпізнавання автомобільних номерів», затверджене наказом Національного університету «Запорізька політехніка» № 139 від 7 квітня 2026 р.

А.2 Призначення розробки

Система опрацювання даних призначена для розпізнавання автомобільних номерів.

А.3 Основні вимоги до програми, що розробляється

А.3.1 Вимоги до функціональних характеристик

Система опрацювання даних для розпізнавання автомобільних номерів забезпечує виконання таких функцій:

– підтримка збереження результатів розпізнавання у базі даних;

– підтримка можливості перегляду історії виконаних запитів і отриманих результатів розпізнавання;

– можливість авторизації користувачів системи;

– забезпечення коректної роботи системи в умовах різної якості зображень та змін освітлення;

– підтримка можливості отримання інформації про ймовірність правильного розпізнавання автомобільного номеру.

А.3.2 Вимоги до інтерфейсу програми

Інтерфейс системи опрацювання даних для розпізнавання автомобільних номерів повинен бути зручним для користувачів. Повинна бути забезпечена можливість роботи в візуальному режимі.

А.3.3 Вимоги до надійності

Система опрацювання даних для розпізнавання автомобільних номерів повинна забезпечити надійне функціонування.

А.3.4 Умови експлуатації

Для експлуатації системи опрацювання даних для розпізнавання автомобільних номерів необхідна наявність персонального комп’ютера.

А.3.5 Вимоги до складу та параметрів технічний засобів

Для роботи системи опрацювання даних для розпізнавання автомобільних номерів потрібні такі технічні та програмні ресурси: апаратна платформа з процесором з тактовою частотою не нижче 3,0 ГГц із підтримкою багатопотокової обробки та наявністю щонайменше чотирьох фізичних ядер. Для стабільної роботи серверної частини доцільно застосовувати процесори класу Intel Core i5 / i7 або аналогічні за продуктивністю моделі AMD Ryzen. У випадку інтенсивної експлуатації або одночасного обслуговування декількох клієнтів бажаною є наявність шести чи восьми ядер.

Обсяг оперативної пам’яті повинен становити не менше 8 ГБ для базової конфігурації. Для серверної інсталяції або роботи з великими

потоками відеоданих рекомендується 16 ГБ і більше. Для зберігання зображень і результатів розпізнавання необхідний накопичувач обсягом не менше 256 ГБ.

Програмне забезпечення повинно передбачати встановлення операційної системи сімейства Windows або Linux із підтримкою 64-бітної архітектури. Для компіляції та супроводу програмного коду доцільним є використання інтегрованого середовища розробки Microsoft Visual Studio з підтримкою стандартів C++ не нижче C++17.

Для функціонування клієнтської частини достатньо комп'ютера з процесором не нижче 2,0 ГГц, оперативною пам'яттю від 4 ГБ та встановленою операційною системою, сумісною із серверною частиною.

А.3.6 Вимоги до маркування і пакування

Система опрацювання даних для розпізнавання автомобільних номерів може бути записана на будь-якому носії інформації.

На пакуванні повинна бути назва – «Система опрацювання даних для розпізнавання автомобільних номерів».

А.4 Вхідні дані до роботи

Вхідними даними до системи опрацювання даних для розпізнавання автомобільних номерів є цифрові зображення автомобілів, включно з номерними знаками.

Вихідними даними системи опрацювання даних для розпізнавання автомобільних номерів є текстове представлення розпізнаного номерного знака, інформація про ймовірність правильно розпізнавання та інша інформація.

ДОДАТОК Б

Фрагмент тексту програми

```
void srvr_app()
{
    crow::SimpleApp pp_srv;

    CROW_ROUTE(pp_srv, "/")
    ([](){
        std::string rsp_txt = "<html><body><h2>PLT RCGN</h2>"
                              "<form action='/upld' method='post'
enctype='multipart/form-data'>"
                              "<input type='file' name='flnm'/>"
                              "<input type='submit'
value='SND'/>"
                              "</form></body></html>";
        return rsp_txt;
    });

    CROW_ROUTE(pp_srv, "/upld").methods("POST"_method)
    ([](const crow::request& rqst){
        std::string bdy = rqst.body;
        std::string pth_fl = sv_fl(bdy);
        std::string txt_rslt = prcss_img(pth_fl);
        std::string rsp = "<html><body><h3>RSLT:</h3><p>" +
txt_rslt + "</p></body></html>";
        return rsp;
    });

    pp_srv.port(8080).multithreaded().run();
}

std::string sv_fl(const std::string& dt_blb)
{
    std::string pth = "tmp.png";
    std::ofstream fl_strm(pth, std::ios::binary);

    std::string bfr;
    bfr.reserve(dt_blb.size());

    for(size_t ndx = 0; ndx < dt_blb.size(); ++ndx)
    {
        char ch = dt_blb[ndx];
        bfr.push_back(ch);
    }

    for(size_t ndx = 0; ndx < bfr.size(); ++ndx)
    {
        fl_strm.put(bfr[ndx]);
    }
```

```
    fl_strm.flush();
    fl_strm.close();

    std::stringstream ss;
    ss << pth;
    std::string rslt = ss.str();

    std::string vrf = "";
    for(size_t ndx = 0; ndx < rslt.size(); ++ndx)
    {
        vrf.push_back(rslt[ndx]);
    }

    return vrf;
}

std::string prcss_img(const std::string& pth_fl)
{
    cv::Mat frm = cv::imread(pth_fl);

    if(frm.empty())
    {
        return "PRCSS FLD";
    }

    cv::Mat gry;
    cv::cvtColor(frm, gry, cv::COLOR_BGR2GRAY);

    cv::Mat blr;
    cv::GaussianBlur(gry, blr, cv::Size(5,5), 0);

    cv::Mat thr;
    cv::threshold(blr, thr, 120, 255, cv::THRESH_BINARY);

    std::vector<std::vector<cv::Point>> cnts;
    cv::findContours(thr,           cnts,           cv::RETR_TREE,
cv::CHAIN_APPROX_SIMPLE);

    std::string txt_rslt = "";

    for(size_t ndx = 0; ndx < cnts.size(); ++ndx)
    {
        cv::Rect rct = cv::boundingRect(cnts[ndx]);
        if(rct.width > 60 && rct.height > 20)
        {
            txt_rslt = "PLT_DTCTD";
        }
    }
```

```
    if(txt_rslt.empty())
    {
        txt_rslt = "NT_FND";
    }

    std::string fnl = "";
    for(size_t ndx = 0; ndx < txt_rslt.size(); ++ndx)
    {
        char ch = txt_rslt[ndx];
        if(ch >= 'A' && ch <= 'Z')
        {
            fnl.push_back(ch);
        }
        else
        {
            fnl.push_back(ch);
        }
    }

    return fnl;
}

bool db_cntnct(const std::string& hst, const std::string& usr,
const std::string& psw, const std::string& dbn)
{
    int rc = sqlite3_open(dbn.c_str(), &db_cnn);
    if(rc != SQLITE_OK)
    {
        db_cnn = nullptr;
        return false;
    }

    const char* crt_tb1 = "CREATE TABLE IF NOT EXISTS rslt (d
INTEGER PRIMARY KEY, plt TEXT, img TEXT, usr TEXT);";
    const char* crt_tb2 = "CREATE TABLE IF NOT EXISTS sr (d
INTEGER PRIMARY KEY, nm TEXT, psh TEXT, rll TEXT);";

    char* err = nullptr;
    sqlite3_exec(db_cnn, crt_tb1, nullptr, nullptr, &err);
    sqlite3_exec(db_cnn, crt_tb2, nullptr, nullptr, &err);

    return true;
}

bool db_dsctnct()
{
    if(db_cnn)
    {
```

```
        sqlite3_close(db_cnn);
        db_cnn = nullptr;
        return true;
    }
    return false;
}

bool db_srt_rslt(const std::string& plt_txt, const std::string&
img_pth, const std::string& usr_nmm)
{
    if(!db_cnn) return false;

    std::stringstream ss;
    ss << "INSERT INTO rslt (plt,img,usr) VALUES('"
       << plt_txt << "','" << img_pth << "','" << usr_nmm <<
"');";

    char* err = nullptr;
    int rc = sqlite3_exec(db_cnn, ss.str().c_str(), nullptr,
nullptr, &err);

    if(rc != SQLITE_OK)
    {
        return false;
    }

    return true;
}

std::vector<std::string> db_gt_hstry(const std::string& usr_nmm)
{
    std::vector<std::string> vct;
    if(!db_cnn) return vct;

    std::stringstream ss;
    ss << "SELECT plt FROM rslt WHERE usr='" << usr_nmm << "';";

    sqlite3_stmt* stmt = nullptr;
    sqlite3_prepare_v2(db_cnn, ss.str().c_str(), -1, &stmt,
nullptr);

    while(sqlite3_step(stmt) == SQLITE_ROW)
    {
        const unsigned char* txt = sqlite3_column_text(stmt, 0);
        if(txt)
        {
            std::string val(reinterpret_cast<const char*>(txt));
            vct.push_back(val);
        }
```

```
    }

    sqlite3_finalize(stmt);
    return vct;
}

bool db_vrf_usr(const std::string& usr_nmm, const std::string&
psw_hsh)
{
    if(!db_cnn) return false;

    std::stringstream ss;
    ss << "SELECT psh FROM sr WHERE nm='" << usr_nmm << "';";

    sqlite3_stmt* stmt = nullptr;
    sqlite3_prepare_v2(db_cnn, ss.str().c_str(), -1, &stmt,
nullptr);

    bool rs = false;

    if(sqlite3_step(stmt) == SQLITE_ROW)
    {
        const unsigned char* txt = sqlite3_column_text(stmt, 0);
        if(txt)
        {
            std::string db_h(reinterpret_cast<const char*>(txt));
            if(db_h == psw_hsh)
            {
                rs = true;
            }
        }
    }

    sqlite3_finalize(stmt);
    return rs;
}

bool db_add_usr(const std::string& usr_nmm, const std::string&
psw_hsh, const std::string& rll)
{
    if(!db_cnn) return false;

    std::stringstream ss;
    ss << "INSERT INTO sr (nm,psh,rll) VALUES('"
       << usr_nmm << "','" << psw_hsh << "','" << rll << "');";

    char* err = nullptr;
    int rc = sqlite3_exec(db_cnn, ss.str().c_str(), nullptr,
nullptr, &err);
```

```
    if(rc != SQLITE_OK)
    {
        return false;
    }

    return true;
}

std::string hsh_psw(const std::string& psw)
{
    std::stringstream ss;
    long sm = 0;

    for(size_t ndx = 0; ndx < psw.size(); ++ndx)
    {
        sm += psw[ndx] * (ndx + 3);
        sm ^= (psw[ndx] << 2);
    }

    ss << sm;
    return ss.str();
}

std::string crt_tkn(const std::string& usr_nmm)
{
    std::stringstream ss;
    std::time_t tmv = std::time(nullptr);

    long sm = tmv;

    for(size_t ndx = 0; ndx < usr_nmm.size(); ++ndx)
    {
        sm += usr_nmm[ndx] * (ndx + 5);
        sm ^= (usr_nmm[ndx] << 1);
    }

    ss << sm;
    return ss.str();
}

bool vrf_tkn(const std::string& tkn)
{
    if(tkn.size() < 5) return false;

    for(size_t ndx = 0; ndx < tkn.size(); ++ndx)
    {
        if(!isdigit(tkn[ndx]))
        {
```

```
            return false;
        }
    }

    return true;
}

void prcss_vdstrm(const std::string& src)
{
    cv::VideoCapture cp(src);

    if(!cp.isOpened())
    {
        return;
    }

    cv::Mat frm;
    std::string usr_nmm = "strm";

    while(true)
    {
        cp >> frm;
        if(frm.empty())
        {
            break;
        }

        std::string txt = xtrct_frm_txt(frm);

        if(txt != "NT_FND" && txt != "PRCSS FLD")
        {
            db_srt_rslt(txt, "vdstrm", usr_nmm);
        }

        int ky = cv::waitKey(10);
        if(ky == 27)
        {
            break;
        }
    }

    cp.release();
}

std::string xtrct_frm_txt(const cv::Mat& frm)
{
    cv::Mat gry;
    cv::cvtColor(frm, gry, cv::COLOR_BGR2GRAY);
```

```
    cv::Mat blr;
    cv::GaussianBlur(gry, blr, cv::Size(5,5), 0);

    cv::Mat thr;
    cv::Canny(blr, thr, 50, 150);

    std::vector<std::vector<cv::Point>> cnts;
    cv::findContours(thr,         cnts,         cv::RETR_EXTERNAL,
cv::CHAIN_APPROX_SIMPLE);

    std::string rs = "NT_FND";

    for(size_t ndx = 0; ndx < cnts.size(); ++ndx)
    {
        cv::Rect rct = cv::boundingRect(cnts[ndx]);

        if(rct.width > 80 && rct.height > 25)
        {
            rs = "PLT_STRM";
        }
    }

    return rs;
}
```

ДОДАТОК В

Слайди презентації

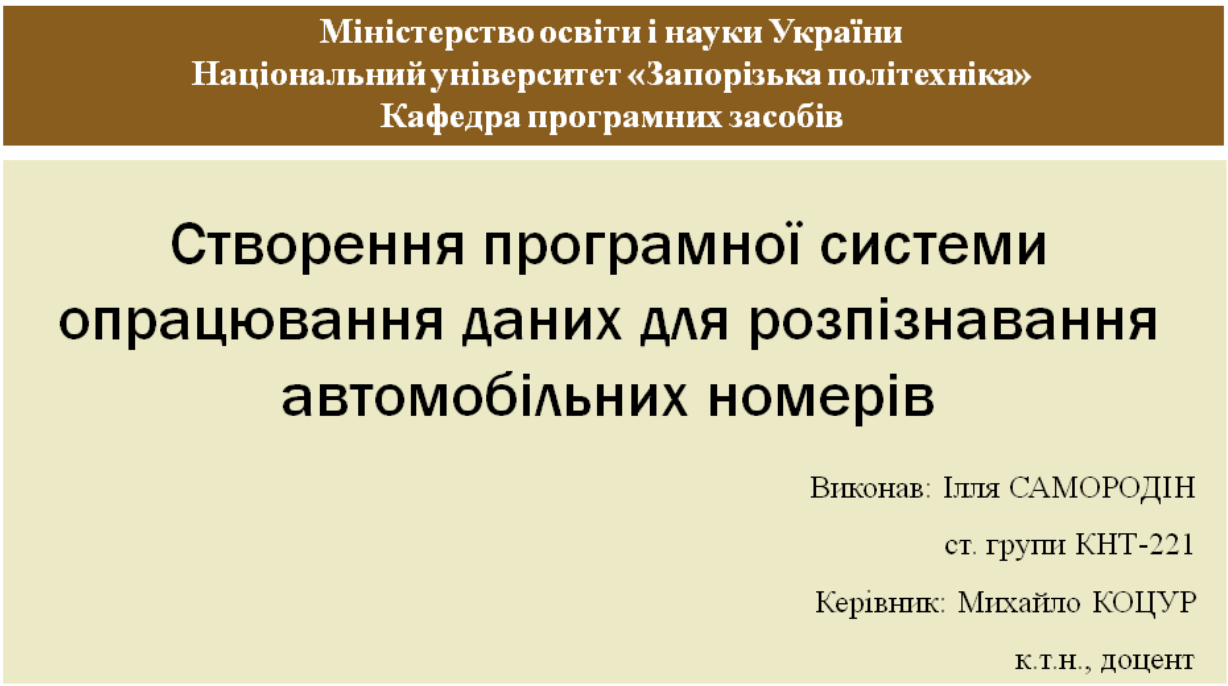


Рисунок В.1 – Слайд 1

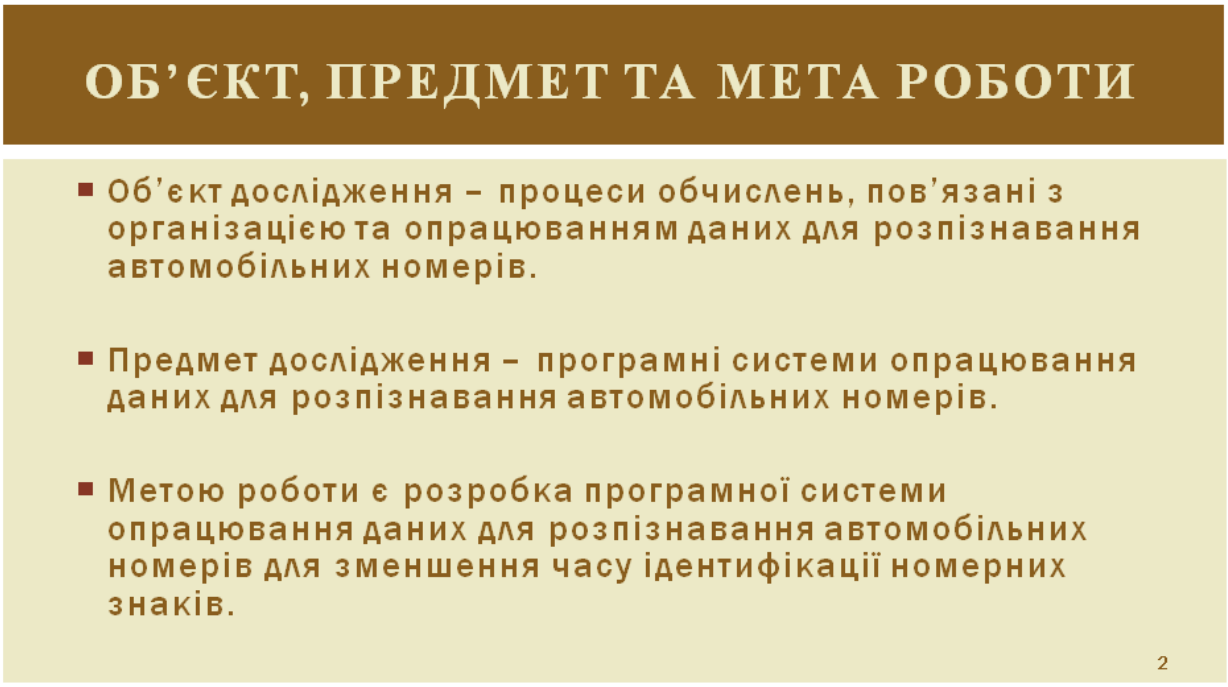


Рисунок В.2 – Слайд 2

ЗАВДАННЯ РОБОТИ

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

– виконати аналіз предметної області та систем опрацювання даних для розпізнавання автомобільних номерів;
– здійснити проєктування системи опрацювання даних для розпізнавання автомобільних номерів;
– створити систему опрацювання даних для розпізнавання автомобільних номерів;
– дослідити розроблену систему опрацювання даних для розпізнавання автомобільних номерів.

3

Рисунок В.3 – Слайд 3

ПОРІВНЯННЯ ІСНУЮЧИХ АНАЛОГІВ

Критерій порівняння	Neurotechnology SentiVeillance	PlateSmart	AutoVu
Висока точність розпізнавання в складних умовах	+	+–	+
Підтримка роботи з існуючими IP-камерами	+	+	+–
Інтеграція зі сторонніми платформами	+–	+	+
Легкість налаштування для початку роботи	–	+–	–
Залежність від специфічного апаратного середовища	+–	–	+
Сталий функціонал аналітики транспортних характеристик	+–	+	+–

4

Рисунок В.4 – Слайд 4

ПОСТАНОВКА ЗАВДАННЯ

При розробці системи опрацювання даних для розпізнавання автомобільних номерів необхідно забезпечити такі функціональні вимоги:

– підтримка можливості завантаження відеопотоків для подальшого розпізнавання автомобільних номерів;

– підтримка збереження результатів розпізнавання у базі даних;

– підтримка можливості перегляду історії виконаних запитів і отриманих результатів розпізнавання;

– можливість авторизації користувачів системи;

– забезпечення коректної роботи системи в умовах різної якості зображень та змін освітлення;

– підтримка можливості отримання інформації про ймовірність правильного розпізнавання автомобільного номеру.

5

Рисунок В.5 – Слайд 5

ПОРІВНЯННЯ МОВ ПРОГРАМУВАННЯ

Критерій порівняння мов програмування	Мова програмування		
	C++	Python	C#
Продуктивність виконання	+	+–	+–
Контроль управління пам'яттю	+	–	–
Інтеграція з бібліотеками комп'ютерного зору	+	+	+–
Можливість низькорівневої оптимізації	+	–	–
Стабільність у довготривалому безперервному режимі	+	+–	+

6

Рисунок В.6 – Слайд 6

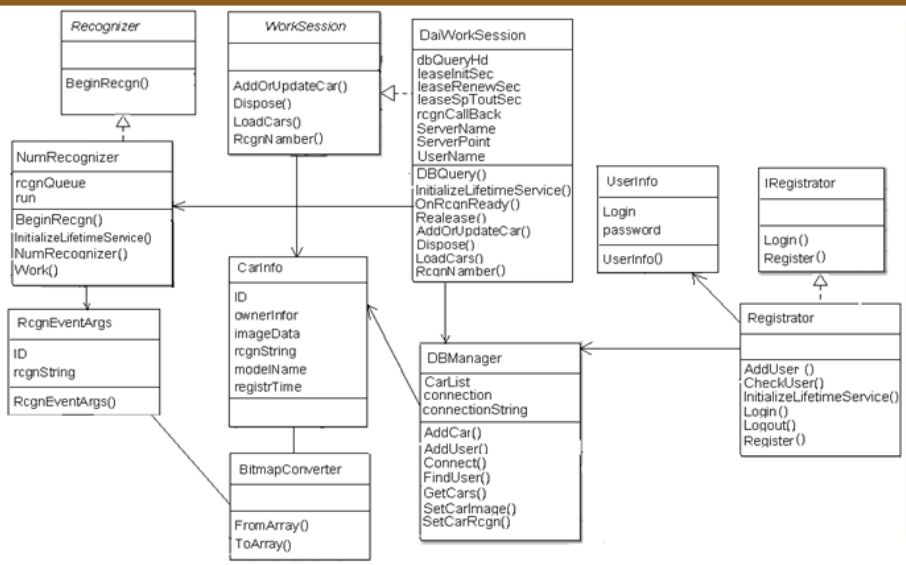
ПОРІВНЯННЯ СЕРЕДОВИЩ РОЗРОБКИ

Критерій порівняння середовищ розробки	Середовища розробки		
	Visual Studio	CLion	Qt Creator
Зручність інтеграції бібліотек	+	+	+–
Інструменти профілювання продуктивності	+	+–	+–
Інтеграція з системами контролю версій	+	+	+–
Документація та професійна підтримка	+	+–	+–
Інтуїтивність інтерфейсу	+	+	+–

7

Рисунок В.7 – Слайд 7

СТРУКТУРА ПРОГРАМИ



8

Рисунок В.8 – Слайд 8

ФУНКЦІОНУВАННЯ ПРОГРАМИ

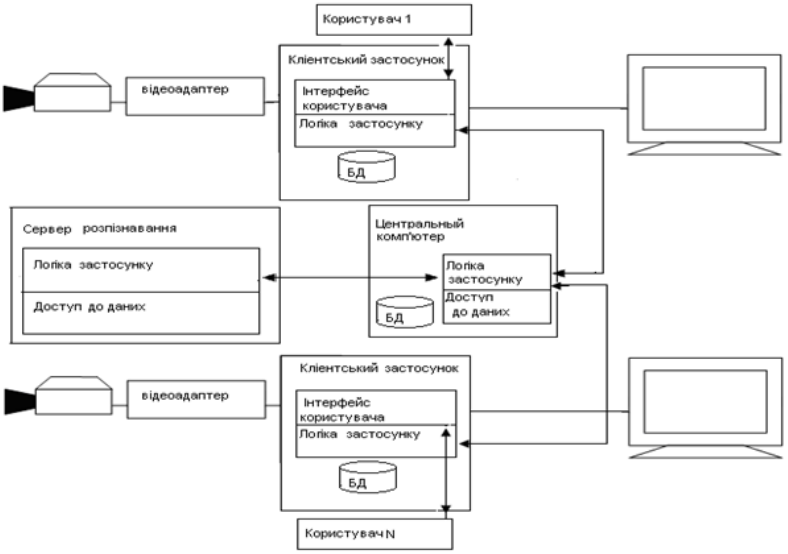


Рисунок В.9 – Слайд 9

БАЗА ДАНИХ

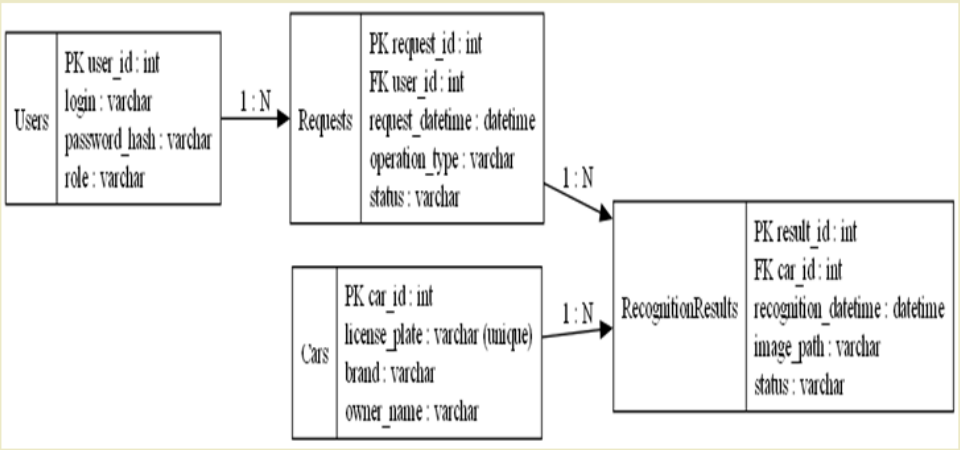


Рисунок В.10 – Слайд 10

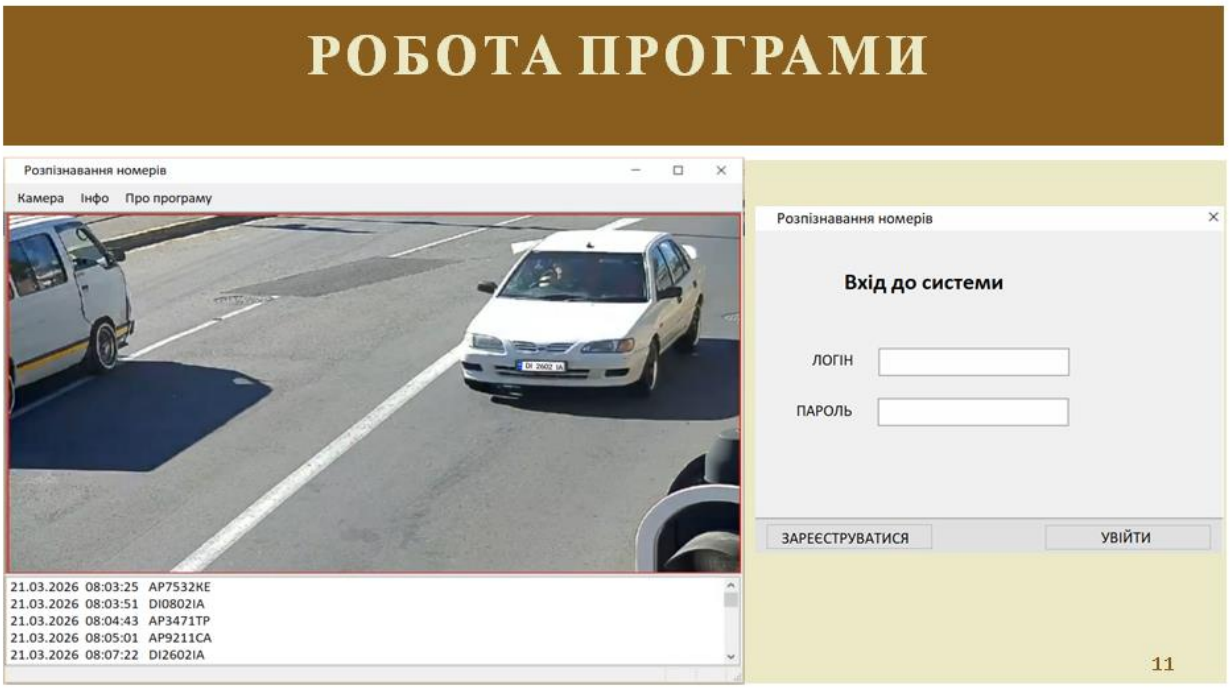


Рисунок В.11 – Слайд 11

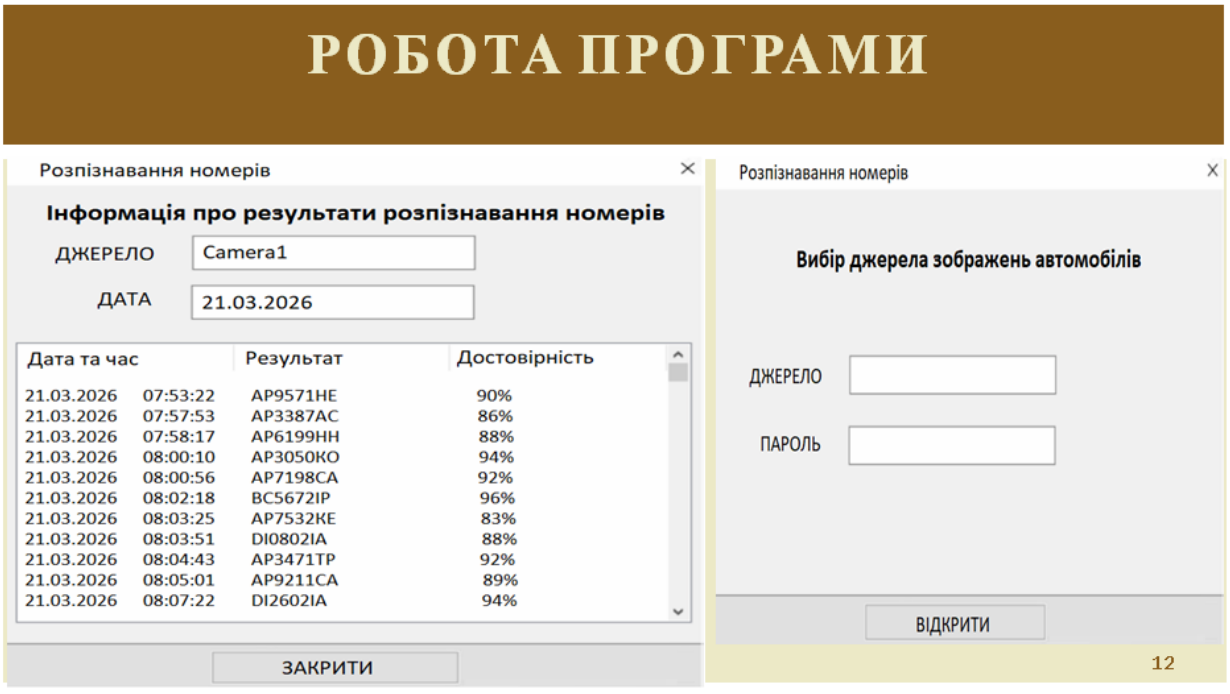


Рисунок В.12 – Слайд 12

Рисунок В.13 – Слайд 13