

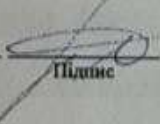


## КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему Метод автоматичного створення бази даних водіїв та номерів  
автомобілів за зображеннями

Галузь знань 12 – Інформаційні технології  
Шифр і назва галузі знань  
Спеціальність 122 – Комп'ютерні науки  
Шифр і назва спеціальності  
Освітня програма Комп'ютерні науки  
Назва освітньої програми

Виконав: студент групи КН-20-2  Олександр СКРИПНІЮК  
Група виконавця Підпис Ім'я, ПРІЗВИЩЕ  
Керівник: д.т.н., проф. каф. КН  Едуард МАНЗІЮК  
Науковий ступінь, посада Підпис Ім'я, ПРІЗВИЩЕ  
Нормоконтроль: к.т.н., доц. каф. КН  Руслан БАГРІЙ  
Науковий ступінь, посада Підпис Ім'я, ПРІЗВИЩЕ

До захисту допускаю:

Зав. кафедри КН, д.т.н., професор

12 06 2024 р.

  
Підпис

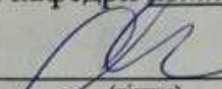
Олександр БАРМАК  
Ім'я, ПРІЗВИЩЕ

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій  
Кафедра комп'ютерних наук  
Освітній ступінь бакалавр  
Галузь знань 12 – Інформаційні технології  
Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук



(підпис)

д.т.н., професор Олександр БАРМАК

« 16 » 02 2024 року

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

1. Тема кваліфікаційної роботи бакалавра: «Метод автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями»

2. Завдання видано студентові Олександр Скрипнюку  
(ім'я, прізвище)

3. Керівник роботи професор кафедри КН Едуард МАНЗЮК  
(посада, ім'я, прізвище)

4. Затверджено наказом університету від « 15 » 02 2024 р. № 8


5. Дата видачі завдання студенту: « 16 » 02 2024 р.

6. Зміст пояснювальної записки (перелік задач) та вихідні дані:

Мета роботи – спрощення формування бази даних водіїв та номерів автомобілів за допомогою застосування інтелектуальних систем. Під час створення програми слід виконати наступні задачі, провести аналіз предметної області щодо методів розпізнавання тексту на номерних знаках, обличчя водія та виконати аналіз наявних рішень щодо подібних задач, розробити метод визначення текстової інформації на номерних знаках автомобілів також розробити метод детектування та визначення обличчя водіїв автомобілів, виконати програмну реалізацію розроблених методів детектування обличчя водіїв та текстової інформації на номерних знаках автомобілів та провести експериментальні дослідження розроблених методів та розробленої програмної реалізації.

7. Календарний план виконання кваліфікаційної роботи бакалавра:

№	Назва етапів (розділів) кваліфікаційної роботи бакалавра	Термін виконання	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи бакалавра з керівником, складання календарного графіка виконання роботи	січень 2024	Виконано
2	Ознайомлення з предметною областю, формулювання мети та задач дослідження, визначення об'єкта та предмета дослідження	лютий 2024	Виконано
3	Проектування та розробка загальної архітектури програмного забезпечення, інтерфейсу користувача, вибір засобів реалізації програмного забезпечення	березень 2024	Виконано
4	Створення та тестування програмного забезпечення	квітень 2024	Виконано
5	Написання пояснювальної записки, урахування зауважень керівника, оформлення згідно вимог	травень 2024	Виконано
6	Розробка презентаційних матеріалів та попередній захист кваліфікаційної роботи	травень 2024	Виконано
7	Отримання відгуку керівника, рецензії, перевірка на плагіат, нормоконтроль	червень 2024	Виконано
8	Підготовка до захисту та захист кваліфікаційної роботи бакалавра	червень 2024	

Виконавець: студент групи КН-20-2  Олександр СКРИПНЮК  
Група виконавця Підпис Ім'я, ПРІЗВИЩЕ

Керівник: в.т.н., проф. каф. КН  Едуард МАНЗЮК  
Науковий ступінь, посада Підпис Ім'я, ПРІЗВИЩЕ

## Анотація

Тема кваліфікаційної роботи бакалавра: «Метод автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями»

Виконавець кваліфікаційної роботи бакалавра: студент групи КН-20-2  
Олександр СКРИПНЮК

Керівник кваліфікаційної роботи бакалавра: к.т.н., професор кафедри КН  
Едуард МАНЗЮК

Кваліфікаційна робота бакалавра містить:

Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
60	27	9	40	3

Метою кваліфікаційної роботи бакалавра є спрощення формування бази даних водіїв та номерів автомобілів за допомогою застосування інтелектуальних систем. Для спрощення формування бази даних водіїв та номерів автомобілів за допомогою застосування інтелектуальних систем був створений наступний метод, в якому на вхід дається зображення транспортного засобу. Це зображення проходять попередню обробку, включаючи фільтрацію шумів та нормалізацію освітлення, що покращує якість для подальшого аналізу. Потім для розпізнавання номерних знаків та облич водіїв використовується згортова нейронна мережа, OpenCV, що забезпечує точну ідентифікацію номерних знаків та облич водіїв на основі вхідного зображення. Для оптичного розпізнавання символів на фото номера використовується Tesseract, що дозволяє перетворювати зображення номерних знаків у текстові дані. Всі отримані дані автоматично записуються в базу даних.

Розроблений метод дозволяє автоматично створювати та оновлювати базу даних водіїв та номерних знаків автомобілів, забезпечуючи швидкий та ефективний контроль транспортних засобів.

Розроблена програмна реалізація методу може використовуватися на пропускних пунктах та платних автомобільних дорогах, а також система може сприяти покращенню безпеки дорожнього руху та зменшенню незаконного використання доріг, забезпечуючи точний та швидкий контроль транспортних засобів та їх власників.

Ключові слова: номер, машина, визначення, фото, база даних, обличчя, людина.

Виконавець: студент групи КН-20-2

Група виконавця



Підпис

Олександр СКРИПНЮК

Ім'я, ПРІЗВИЩЕ

## Зміст

Перелік скорочень .....	4
Вступ.....	5
Розділ 1 Аналіз систем та методів з автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями.....	7
1.1 Аналіз предметної області .....	7
1.2 Аналіз рішень з розпізнавання автомобільних номерів та водіїв .....	8
1.3 Огляд методів розпізнавання номерів та водіїв .....	9
1.4 Огляд та аналіз існуючих систем розпізнавання номерів та водіїв .....	10
1.5 Мета, та задачі до реалізації системи.....	15
Розділ 2 Проектування метода автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями .....	17
2.1 Моделі та методи автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями.....	17
2.2 Функціональна структура програмної реалізації.....	21
2.3 Розробка архітектури нейронної мережі .....	25
2.4 Проектна архітектура системи та варіанти використання.....	27
2.5 Структура системи та проектування бази даних .....	31
2.6 Підготовка робочих вхідних даних для системи .....	37
2.7 Особливості використання спеціалізованих програмних компонентів .....	38
2.8 Оцінювання ефективності розроблених методів .....	39
2.9 Висновки до розділу 2 .....	40
Розділ 3 Програмна реалізація методу автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями.....	42
3.1 Визначення шляхів дослідження та засобів створення програмного забезпечення .....	42
3.2 Вибір засобів розробки програмної реалізації запропонованого методу ....	43
3.3 Структура та функціональне призначення програмних складових системи.....	45
3.4 Особливості реалізації програмних складових системи .....	48
3.5 Тестування програмної реалізації метода та вимоги до розгортання .....	53
3.6 Експериментальні дослідження ефективності розроблених методів .....	61

3.7 Висновки до розділу 3.....	63
Загальні висновки.....	65
Перелік посилань.....	67
Додатки	

**Перелік скорочень**

<b>Скорочення, термін, позначення</b>	<b>Пояснення</b>
БД	База даних
НМ	Нейронна мережа
ПП	Програмний продукт
КРБ	Кваліфікаційна робота бакалавра
СКБД	Система керування базами даних
MS	Microsoft

## Вступ

Кваліфікаційна робота бакалавра показує комплексний процес розробки та впровадження методу, спрямованого на автоматизацію створення бази даних, яка містить інформацію про водіїв та номери автомобілів, і цей метод базується на аналізі та обробці зображень.

**Актуальність.** Розвиток автомобільної індустрії та широке впровадження штучного інтелекту у сферу транспорту прогнозується, як фактор, що може вплинути на можливість заміни людей автоматизованими системами на дорогах та пропускних пунктах. Зокрема, зростання швидкості, з якою автомобілі рухаються, ставить перед собою завдання забезпечення безпеки на дорогах, що може бути складним завданням для людських операторів. Отже, розробка продукту для автоматичного розпізнавання номерів та водіїв стає актуальною.

Актуальність цієї проблематики відображається у багатьох аспектах дорожньої безпеки та ефективного управління транспортним потоком. По-перше, забезпечення безпеки на дорогах є однією з головних завдань держави та органів управління. Автоматизовані системи розпізнавання номерів та водіїв можуть сприяти у вчасному виявленні порушень правил дорожнього руху та запобігати аварійним ситуаціям.

По-друге, ефективний контроль дорожнього руху є важливим для забезпечення плавного руху транспорту та запобігання заторів. Застосування технологій розпізнавання номерів та водіїв може допомогти в ідентифікації транспортних засобів, їх швидкості та маршрутів, що сприятиме більш ефективному управлінню дорожнім рухом.

І нарешті, оптимізація використання дорожнього простору є ключовим завданням для міст та мегаполісів, де дефіцит місця на дорогах є проблемою. Впровадження систем автоматичного розпізнавання може сприяти більш ефективному розподілу транспортних засобів та зменшенню заторів, що відобразиться на загальній продуктивності та комфортності дорожнього руху.

**Об'єкт дослідження** – процес автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями.

**Предмет дослідження** – методи та технології машинного навчання для розпізнавання та порівняння.

**Мета кваліфікаційної роботи бакалавра** – спрощення формування бази даних водіїв та номерів автомобілів за допомогою застосування інтелектуальних систем.

**Завдання кваліфікаційної роботи бакалавра:**

– провести аналіз предметної області щодо методів розпізнавання тексту на номерних знаках, обличчя водія та виконати аналіз наявних рішень щодо подібних задач;

– розробити метод визначення текстової інформації на номерних знаках автомобілів;

– розробити метод детектування та визначення облич водіїв автомобілів;

– виконати програмну реалізацію розроблених методів детектування облич водіїв та текстової інформації на номерних знаках автомобілів;

– провести експериментальні дослідження розроблених методів та розробленої програмної реалізації.

## **Розділ 1 Аналіз систем та методів з автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями**

### **1.1 Аналіз предметної області**

У даному підрозділі буде проведено аналіз предметної області та розглянуто ключові технологічні аспекти для виконання КРБ(Кваліфікаційна робота бакалавра), а також можливості використання. Зосередимося на вивченні сучасних підходів до розпізнавання об'єктів на зображеннях, що є критичним для ефективної реалізації системи.

Під час розробки систем для автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями, зазвичай стикається з рядом технологічних викликів. Одним із найважливіших є те, що потрібно використовувати передові алгоритми машинного навчання та комп'ютерного зору для точного та ефективного розпізнавання автомобілів та номерних знаків на зображеннях. Додатково, потрібні ще складніші алгоритми для визначення обличчя на зображенні, оскільки потрібно враховувати скло автомобіля та інші фактори, які можуть вплинути на якість зображення.

Оскільки метод має використовуватися в реальному часі для обробки великого потоку зображень з камер спостереження, важливо забезпечити високу точність та швидкість обробки. Але, в реальності не вийде нескінченно збільшувати точність або швидкість, і зазвичай настає момент, коли потрібно починати шукати компроміс між швидкістю та точністю, оптимізуючи ресурси обчислювальної системи.

Робота з особистими даними водіїв, їх зображеннями та номерами автомобілів, вимагає дотримання відповідних законодавчих вимог щодо захисту даних та конфіденційності. Метод повинен гарантувати безпеку та захищеність персональної інформації, збереженої в базі даних. Дані повинні зберігатися в захищеному вигляді на випадок витоку та злому, і важливо регулярно проводити аудит безпеки для виявлення потенційних ризиків.

Однак, не зважаючи на ці аспекти, однією з головних переваг системи є можливість використання зібраних даних для аналізу дорожнього руху, статистичного аналізу аварійності, виявлення та прогнозування тенденцій у руховому потоці, а також для оптимізації дорожньої інфраструктури та підвищення рівня безпеки на дорогах.

Після аналізу предметної області було визначено, що аспекти, які розглядалися вище в тексті, додадуть складність для створення програми. Однак, не зважаючи на ці аспекти, що ускладнюють написання програми, однією з головних переваг системи є можливість використання зібраних даних для аналізу дорожнього руху, статистичного аналізу аварійності, виявлення та прогнозування тенденцій у руховому потоці, а також для оптимізації дорожньої інфраструктури та підвищення рівня безпеки на дорогах.

## **1.2 Аналіз рішень з розпізнавання автомобільних номерів та водіїв**

Одним з найпоширеніших рішень для виявлення об'єктів та тексту на зображенні є використання глибоких нейронних мереж (Deep Neural Networks - DNN) [1]. Ці мережі зазвичай навчаються на великих наборах даних, що містять зображення об'єктів для визначення, на якій основі вони будуть використовуватися.

Іншим рішенням є використання комбінації алгоритмів комп'ютерного зору [2] та обробки зображень для виявлення об'єктів та тексту на зображеннях. Ці алгоритми можуть базуватися на виявленні контурів та особливостей зображень, після чого використовуються методи шаблонного порівняння або класифікації для розпізнавання об'єктів та тексту.

Також часто в готових рішеннях використовуються готові програмні бібліотеки, такі як OpenALPR [3] або Tesseract [4], які також є поширеним підходом для розпізнавання об'єктів та тексту. Ці бібліотеки вже мають вбудовані моделі для розпізнавання об'єктів та можуть бути легко інтегровані у систему.

Після огляду вище перерахованих рішень було проведено їх аналіз. Рішення з використанням комп'ютерного зору, яке використовується для виявлення об'єктів та тексту на зображеннях, базується на використанні фільтрів для виділення контурів або тексту, а також використанні алгоритмів для визначення областей різних форм на зображеннях, де можуть знаходитися об'єкти та текст.

Рішення з використанням глибокого навчання, таке як згорткові нейронні мережі (Convolutional Neural Networks - CNN) [5], широко використовуються для розпізнавання тексту на зображеннях. Цей метод використовує великий набір даних з зображеннями, на яких зображено об'єкти, які потрібно розпізнати.

Додатковими рішеннями є обробка зображень, ці рішення містять фільтрацію, сегментацію та виокремлення різних елементів на зображенні. Наприклад, можна використовувати алгоритми для виокремлення номерних знаків від інших об'єктів на зображенні та підготовки їх для подальшого розпізнавання.

Після огляду та аналізу рішень для розпізнавання автомобільних номерів та водіїв, було вирішено, що глибокі нейронні мережі з додатковими рішеннями для обробки зображень є кращим варіантом, оскільки вони демонструють високу точність розпізнавання та можуть працювати в реальному часі.

### **1.3 Огляд методів розпізнавання номерів та водіїв**

Зараз для розпізнавання автомобільних номерів та водіїв використовуються різні методи з різними підходами, які можна класифікувати наступним чином:

Метод комп'ютерного зору та обробки зображень [6]. Цей підхід полягає в застосуванні алгоритмів обробки зображень для виявлення та розпізнавання номерних знаків на фотографіях або відео. Він може включати такі кроки, як визначення контурів, сегментація областей, виявлення символів та їх розпізнавання.

Метод глибокого навчання [7]. Цей підхід використовує глибокі нейронні мережі для автоматичного розпізнавання номерних знаків та символів на них. Він вимагає великого обсягу даних для навчання та може включати такі етапи, як сегментація зображення та класифікація символів.

Метод аналізу геометричних особливостей [8]. Цей підхід зосереджений на використанні геометричних характеристик, таких як розмір, форма та розміщення символів, для розпізнавання номерних знаків. Він може включати аналіз форми символів, розташування та взаємних відстаней між ними.

Метод зіставлення шаблонів [9]. Цей підхід використовує попередньо відомі шаблони номерних знаків для порівняння зі зображеннями та визначення відповідності. Він може включати такі методи, як порівняння шаблонів за допомогою кореляційних алгоритмів або шаблонного визначення символів.

Метод, що базується на використанні баз даних [10]. Цей підхід використовує бази даних для зберігання та пошуку інформації про номерні знаки та водіїв для подальшого визначення їх ідентифікації. Він може включати методи пошуку та порівняння зі збереженими даними для визначення відповідності.

У загальному, в розпізнаванні автомобільних номерів та водіїв використовуються різноманітні методи, які поєднують у собі комп'ютерний зір, нейронні мережі, обробку зображень та бази даних. Вибір конкретного методу залежить від вимог до точності та швидкодії системи.

#### **1.4 Огляд та аналіз існуючих систем розпізнавання номерів та водіїв**

Одним популярним сервісом з розпізнаванням номерів є «OpenALPR» [11] (рисунок 1.1), який також пропонує розпізнавання номерних знаків з відео та фотографій у реальному часі. Його можна використовувати для різних цілей, таких як забезпечення безпеки, контроль доступу та автоматизація паркування. З переваг можна виділити можливість обробляти фото номерів в реальному часі з відео та збереження результатів.

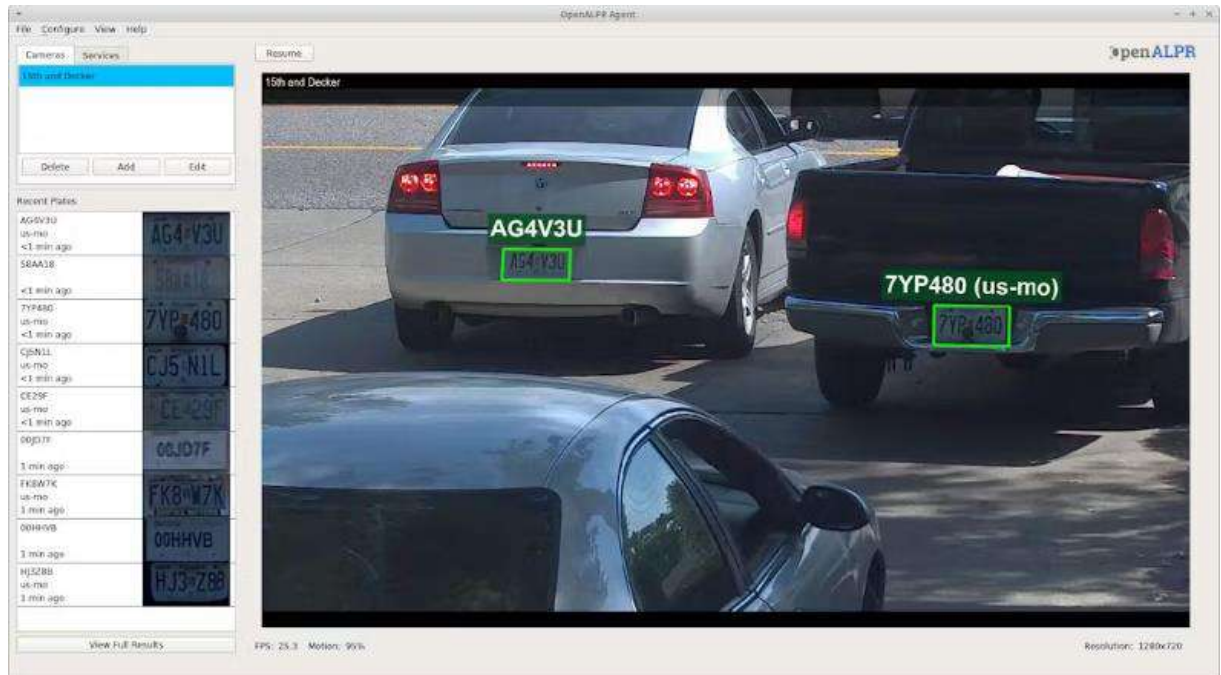


Рисунок 1.1 – Сторінка розпізнавання в програмі «OpenALPR» [12]

Великим недоліком системи «OpenALPR», є те що ця система може розпізнавати тільки автомобільні номери.

Також було розглянуто програму від компанії «Rekor Systems» [13] (рисунок 1.2). Ця компанія також пропонує сервіси для розпізнавання номерних знаків та водіїв. Вони надають рішення для безпеки, паркування, транспортних перевезень та багато іншого. З переваг можна виділити безпеку програми та невелику ціну.



Рисунок 1.2 – Розпізнавання номерів в програмі «Rekor Systems» [14]

Недоліком є те, що для високого показника розпізнавань в «Rekor Systems» потрібне додаткове обладнання, а також програма не може працювати з обличчями.

Однією з систем з розпізнаванням облич є «Amazon Rekognition» [15] (рисунок 1.3), вона собою являє інтелектуальну систему комп'ютерного зору від Amazon Web Services (AWS), яка надає послуги розпізнавання облич, об'єктів, тексту та сцен на зображеннях і відео яка є відкритим програмним забезпеченням, яке надає широкі можливості для обробки зображень і відео.

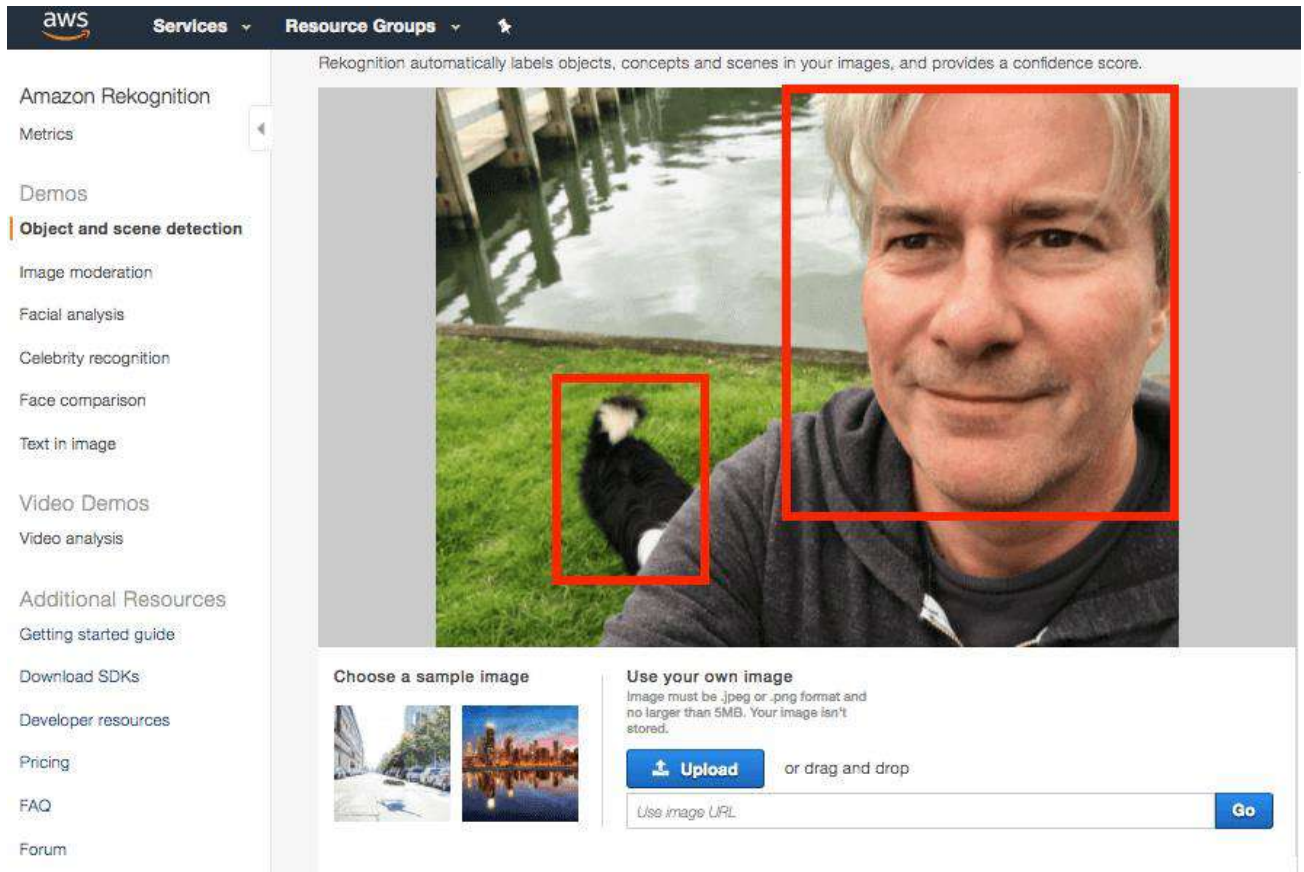


Рисунок 1.3 – Розпізнавання облич в системі «Amazon Rekognition» [16]

Великим недоліком системи «Amazon Rekognition», є те що вона є хмарним сервісом і в місцях без інтернету не зможе працювати.

Також було розглянуто одну з найпопулярніших популярних систем з розпізнаванням облич є «OpenCV» [17] (рисунок 1.4), яка є відкритим програмним забезпеченням, яке надає широкі можливості для обробки зображень і відео. Цей сервіс використовується в різних галузях, включаючи безпеку, медицину, робототехніку та інші.

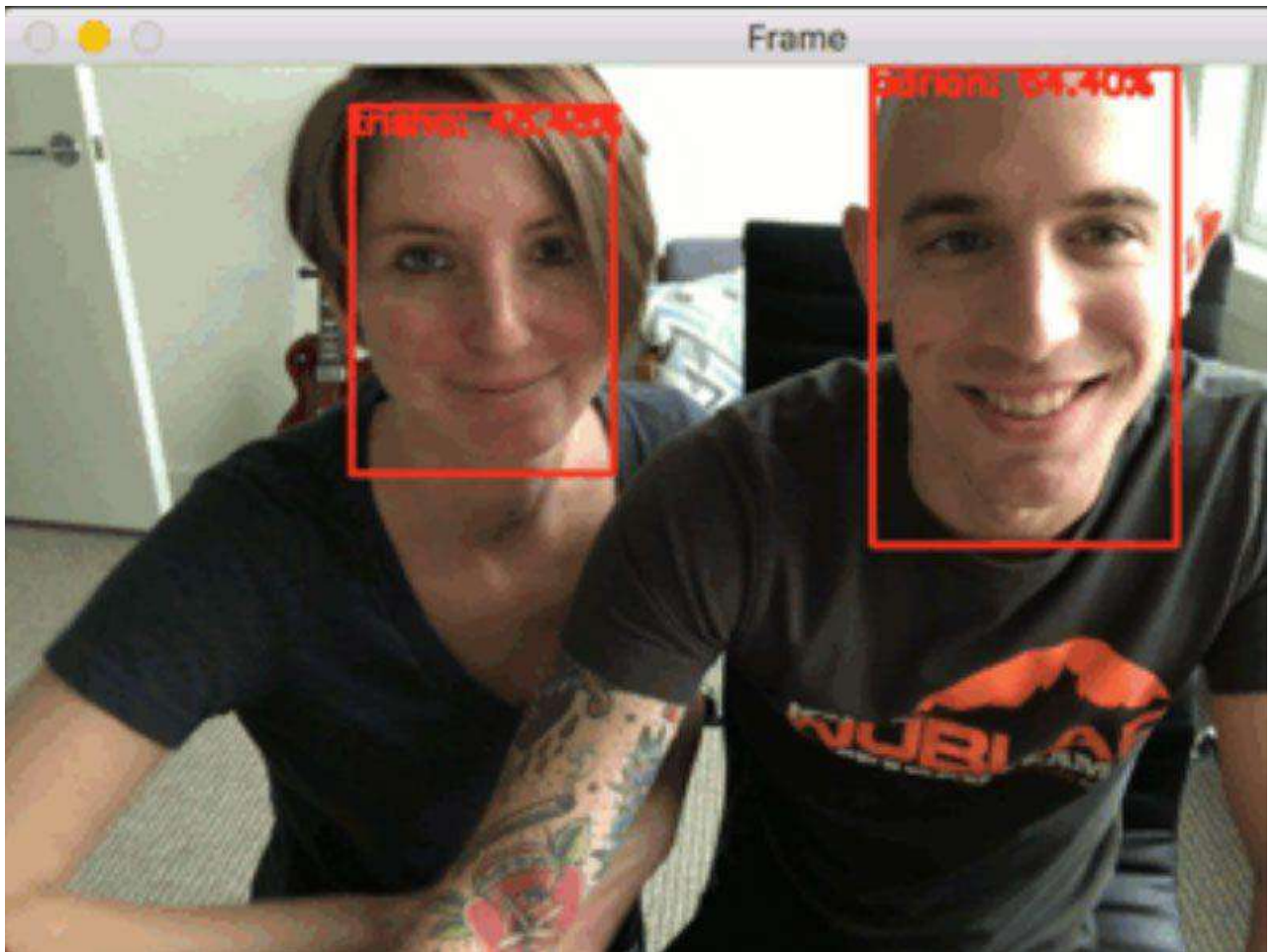


Рисунок 1.4 – Розпізнавання облич в системі «OpenCV» [18]

В «OpenCV» [19] є просто безліч переваг перед своїми конкурентами це:

- відкритий код, що забезпечує гнучкість та можливість налаштування під конкретні потреби користувача;
- широкі можливості для обробки зображень, відео, а також для реалізації різноманітних алгоритмів комп'ютерного зору;
- висока ефективність та швидкість обробки завдяки оптимізації коду та використанню оптимізованих алгоритмів;
- розширюваність за допомогою додаткових модулів та бібліотек, які регулярно додаються та підтримуються спільнотою;
- легкість використання завдяки інтуїтивно зрозумілому API та документації, що спрощує процес розробки та навчання для новачків у сфері комп'ютерного зору;
- кросплатформенність OpenCV підтримує різні операційні системи, включаючи Windows, macOS, Linux та мобільні платформи Android, iOS;

- активна спільнота користувачів та розробників OpenCV надає значну підтримку через форуми, блоги, навчальні матеріали та інші ресурси;
- інтеграція з іншими бібліотеками та фреймворками OpenCV може легко інтегруватися з іншими інструментами для машинного навчання, такими як TensorFlow, Keras, PyTorch, що дозволяє створювати більш складні та потужні системи обробки зображень.

Ці переваги роблять OpenCV однією з найпопулярніших та найефективніших платформ для розробки систем розпізнавання зображень та комп'ютерного зору.

Також оскільки для написання системи планується використовувати бібліотеку EMGU що являється форком OpenCV для платформи .Net то можна рахувати що OpenCV є найкращим варіантом системи для визначення облич і подальший розгляд інших систем не буде виконуватись.

Після огляду зразків програм було зроблено висновок що перспективність розробки продукту для розпізнавання номерів та водіїв є достатньо висока, але якщо розробка буде проводитись з використанням системи OpenCV, то перспективність розробки ще зросте. На теперішній час актуальними напрямками є розпізнавання номерів та людей на стоянках та дорогах.

### **1.5 Мета, та задачі до реалізації системи**

Метою КРБ є спрощення формування бази даних водіїв та номерів автомобілів за допомогою застосування інтелектуальних систем, основні задачі для досягнення цієї мети включають:

- проведення аналізу предметної області щодо методів розпізнавання тексту на номерних знаках, обличчя водія та виконати аналіз наявних рішень щодо подібних задач;
- розробка методу визначення текстової інформації на номерних знаках автомобілів;
- розробка методу детектування та визначення обличчя водіїв автомобілів;

– виконання програмної реалізації розроблених методів детектування обличь водіїв та текстової інформації на номерних знаках автомобілів;

– проведення експериментальних досліджень розроблених методів та розробленої програмної реалізації.

## Розділ 2 Проектування метода автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями

### 2.1 Моделі та методи автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями

Для аналізу та роз'яснень процесу автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями було створено схему методу яка зображена на рисунку 2.1.

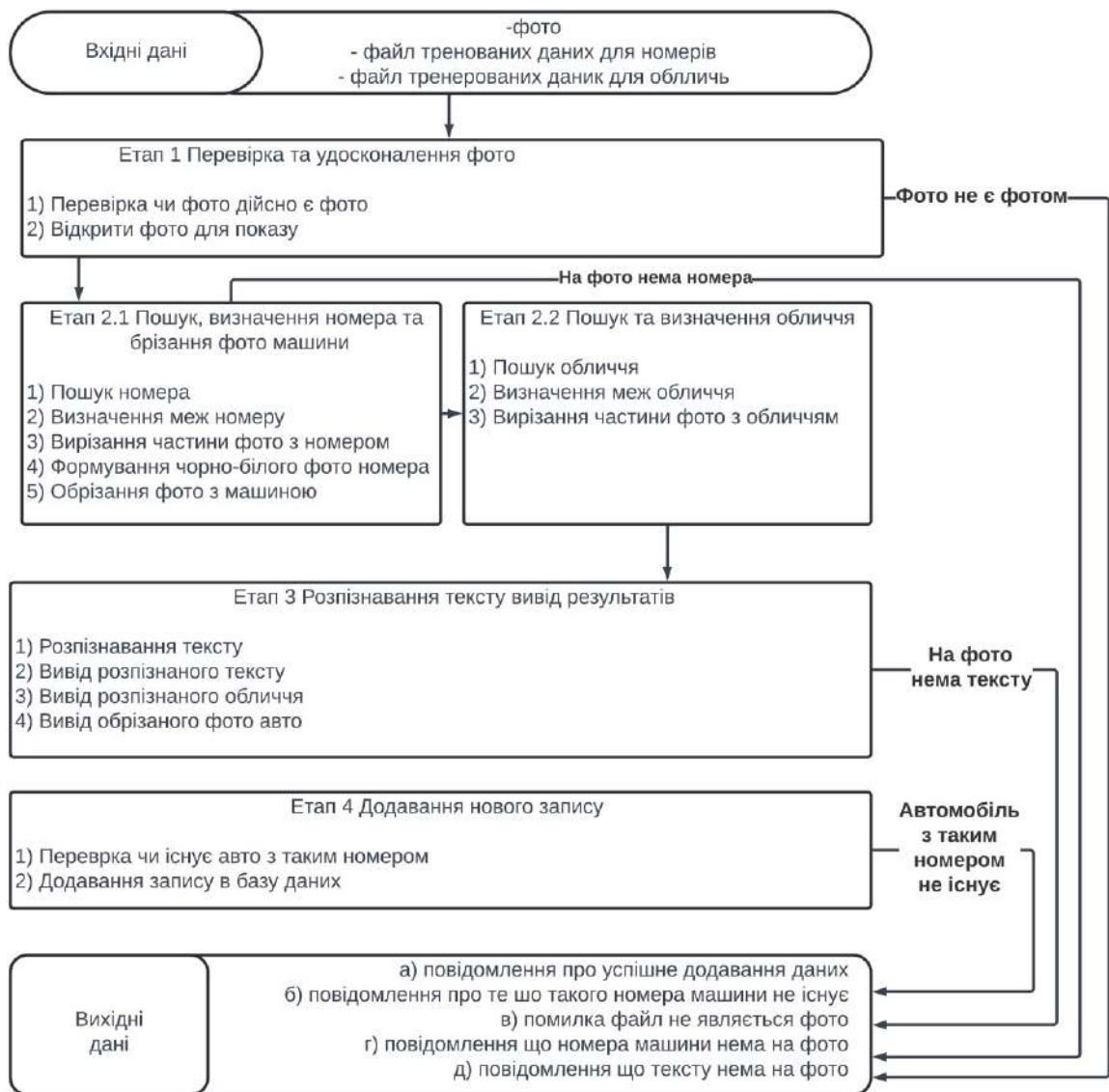


Рисунок 2.1 – Схема методу автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями.

Далі наведемо детальний опис кожної частини схеми.

*Вхідними даними* методу автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями.

– Фото: Це зображення, на якому програма проводитиме пошук та аналіз номерів автомобілів та облич водіїв. Фото може бути отримане з будь-якого джерела, такого як камера спостереження, фотоапарат або мобільний телефон.

– Файл тренуваних даних для номерів: Цей файл містить необхідні дані для розпізнавання тексту номерних знаків. Він включає алфавіт країни, можливі цифри та символи, а також методи читання тексту, які використовуються алгоритмом розпізнавання.

– Файл тренуваних даних для облич: Цей файл містить метрики та параметри, які використовуються для пошуку та визначення облич водіїв на зображеннях. Він може включати параметри детектора облич, які допомагають алгоритму знаходити та виділяти обличчя на фотографіях.

Етап 1 – Перевірка та удосконалення фото:

– Перевірка чи фото являється фото: Під час цієї операції програма автоматично аналізує вхідне зображення, щоб переконатися, що воно відповідає вимогам формату фотографії. Для цього вона може використовувати різні методи, такі як перевірка розширення файлу, аналіз метаданих або визначення співвідношення сторін. Якщо фотографія відповідає критеріям, програма продовжує свою роботу; в іншому випадку вона виводить повідомлення про помилку з підказкою «Файл не являється фотографією».

– Відкрити фото для показу: Ця операція полягає у відображенні відкритого фото у лівій частині програми. Це дозволяє користувачу переглянути фотографію перед подальшою обробкою та впевнитися, що вона відповідає його очікуванням. Відображення фотографії також може бути корисним для визначення правильного ракурсу чи вибору областей для аналізу.

Етап 2.1 – Пошук, визначення номера та обрізання фото машини.

– Пошук номера: Під час цієї операції запускається метод FindLicensePlate, який використовується для пошуку всіх можливих областей на

фотографії, де може знаходитися автомобільний номер. Для цього використовуються різні техніки та алгоритми, наприклад, пошук форми знаку на фото.

– Визначення меж номера: Після того як, програма знаходить область, де знаходиться номер, вона визначає точні межі цієї області. Це важливий крок, оскільки він дозволяє точно виділити номер для подальшої обробки.

– Вирізання частини фото з номером: Після визначення меж номера програма вирізає з вхідного фото меншу область, що містить саме зображення номера. Це дозволяє зосередитися на аналізі самого номера без зайвих даних.

– Формування чорно-білого зображення номера: Після вирізання фотографії з номером програма перетворює її в чорно-біле зображення. Це дозволяє полегшити подальший процес розпізнавання тексту.

– Обрізання фото з машиною: Окрім номера, програма також вирізає меншу область з вихідного зображення, яка містить автомобіль. Це допомагає ідентифікувати та зберегти лише ті області фотографії, які є важливими для подальшого аналізу.

У випадку, якщо програма не знайде жодного номера на фотографії, вона показує користувачеві повідомлення з помилкою «На фото нема номерів авто» та очікує наступну фотографію для обробки.

Етап 2.2 – Пошук та визначення обличчя.

– Пошук обличчя: Під час цього етапу запускається метод `ProcessImageFaceAsync`, який використовується для пошуку всіх можливих областей на фотографії, де може знаходитися обличчя людини. Для досягнення цього використовуються різні техніки, включаючи пошук контурів, що нагадують обличчя на фото.

– Визначення меж обличчя: Після знаходження областей, де можуть знаходитися обличчя, програма визначає точні межі цих областей. Це допомагає у точній ідентифікації та виділенні обличчя для подальшої обробки.

– Вирізання частини фото з обличчям: Після визначення меж обличчя програма вирізає з вихідного фото меншу область, яка містить саме обличчя людини.

Етап 3 – Розпізнання тексту вивід даних.

– Розпізнання тексту з чорно-білого фото: Під час цього етапу програма спробує розпізнати текст, який може бути присутній на чорно-білому фото номерів автомобілів. Вона використовує для цього відповідні алгоритми розпізнавання тексту, щоб знайти та інтерпретувати будь-який текст на фотографії.

– Вивід розпізнаного тексту: Після розпізнавання тексту програма запускає метод `AddLabelAndImage` для виводу розпізнаного тексту та відповідного фото. Це дозволяє користувачу переглянути розпізнаний текст разом з відповідною фотографією номера автомобіля у чорно-білому вигляді.

– Вивід розпізнаного обличчя: Програма також виводить обрізане обличчя водія для подальшого перегляду. Це дозволяє впевнитися, що обличчя було коректно розпізнано та може бути ідентифіковано.

– Вивід обрізаного фото авто: Крім того, програма виводить обрізане фото автомобіля для перегляду. Це допомагає користувачу перевірити, чи було фото автомобіля коректно оброблено та чи вдалося визначити номер автомобіля.

У випадку, якщо програма не може знайти жодного тексту на фото номера автомобіля, вона показує повідомлення у вигляді `MessageBox` з помилкою «На фото нема тексту» та переходить до очікування наступного фото для подальшого аналізу.

Етап 4 – Додавання нового запису:

– перевірка чи існує авто з таким номером спочатку програма перевіряє, чи існує запис в базі даних авто з таким державним номером автомобіля;

– додавання запису в базу даних якщо виявлено, що автомобіль з таким номером існує в базі даних, програма додає новий запис до бази.

*Вихідні дані:*

а) повідомлення про успішне додавання даних з'являється якщо додавання запису до бази даних пройшло успішно, програма виводить повідомлення про це користувачу;

б) повідомлення про те, що такого номера машини не існує виводиться якщо в результаті перевірки виявлено, що автомобіль з таким номером вже присутній в базі даних, програма повідомляє користувача про це;

в) помилка "файл не являється фото" виводиться, якщо вхідні дані програми не відповідають вимогам, тобто якщо вказаний файл не є фотографією;

г) повідомлення, що номера машини нема на фото виводиться, якщо на вхідному фото відсутній номер автомобіля;

д) повідомлення, що тексту нема на фото виводиться, якщо програма не змогла розпізнати жодного тексту на вхідному фото номера автомобіля.

## **2.2 Функціональна структура метода автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями**

У цьому підрозділі розглядається внутрішній механізм системи, що забезпечує виконання функцій системи розпізнавання номерних знаків. Для цього система розділяється на декілька основних функціональних груп, кожна з яких відповідає за конкретний аспект її роботи. Наведені нижче функціональні групи допоможуть структурувати та пояснити роботу системи.

Групи функцій та функції у цих групах.

Функціональна структура методів для класу `NumberPlateRecognizer` складається з наведених нижче основних груп функцій.

1. Ініціалізація OCR та налаштування.
2. Попередня обробка зображення.
3. Виявлення контурів та ліцензійних знаків.
4. Афінні перетворення та нормалізація зображень.
5. Розпізнавання тексту.
6. Фільтрація та очищення результатів.

Далі розглянемо кожен з цих груп детальніше.

– Ініціалізація OCR та налаштування – ця група функцій відповідає за ініціалізацію та налаштування OCR-двигуна, який буде використовуватися для розпізнавання тексту на номерних знаках. Ініціалізація містить завантаження необхідних даних для OCR, вибір мови розпізнавання, а також налаштування чорного списку символів, які не повинні бути розпізнані.

– Попередня обробка зображення – ця група функцій займається підготовкою зображення для подальшого аналізу. Вона містить перетворення зображення в градації сірого, що зменшує кількість даних для обробки та виділяє основні контури. Потім застосовується алгоритм Canny для виявлення країв, що допомагає ідентифікувати межі об'єктів на зображенні.

– Виявлення контурів та ліцензійних знаків – ця група функцій відповідає за виявлення та ідентифікацію контурів, які можуть відповідати номерним знакам. Система виконує пошук контурів на зображенні, аналізує ієрархію контурів для виявлення потенційних номерних знаків, та обчислює мінімальні обмежувальні прямокутники для кожного контуру. Важливим аспектом є відбір контурів за певними критеріями, такими як розмір та кількість дітей в ієрархії.

– Афінні перетворення та нормалізація зображень – ця група функцій забезпечує корекцію перспективи та масштабування зображення номерного знаку. Це дозволяє отримати зображення у прямокутній формі, навіть якщо воно було знято під кутом. Після цього зображення масштабується до стандартного розміру, що полегшує подальшу обробку та розпізнавання.

– Розпізнавання тексту – ця група функцій відповідає за безпосереднє розпізнавання тексту на зображеннях номерних знаків за допомогою OCR. OCR-двигун аналізує підготовлене зображення та витягує текст, який потім використовується для подальшої обробки.

– Фільтрація та очищення результатів – ця група функцій займається очищенням та фільтрацією результатів розпізнавання, щоб забезпечити коректність та релевантність розпізнаних номерних знаків. Очищення включає

видалення зайвих пробілів та невідповідних символів. Результати фільтруються на основі заданих критеріїв, таких як мінімальна довжина тексту та відсутність українських символів.

Функціональна структура системи розпізнавання номерних знаків складається з декількох основних груп функцій, кожна з яких виконує важливу роль у процесі обробки та розпізнавання. Кожна група функцій відповідає за конкретний аспект роботи системи, починаючи з ініціалізації OCR-двигуна і закінчуючи фільтрацією результатів. Така структура дозволяє ефективно організувати роботу програми, забезпечуючи чітке розділення відповідальностей між різними частинами системи. Це, у свою чергу, сприяє підвищенню точності та надійності системи розпізнавання номерних знаків.

Також для того, щоб показати приблизний вигляд інтерфейсу програми було створено схематичний вигляд першої сторінки програми, з розпізнаванням номера та обличчя, було створено дві приблизні схеми (рисунок 2.2 – 2.3), з підписами назв елементів.



Рисунок 2.2 – Перший схематичний вигляд першої сторінки програми.

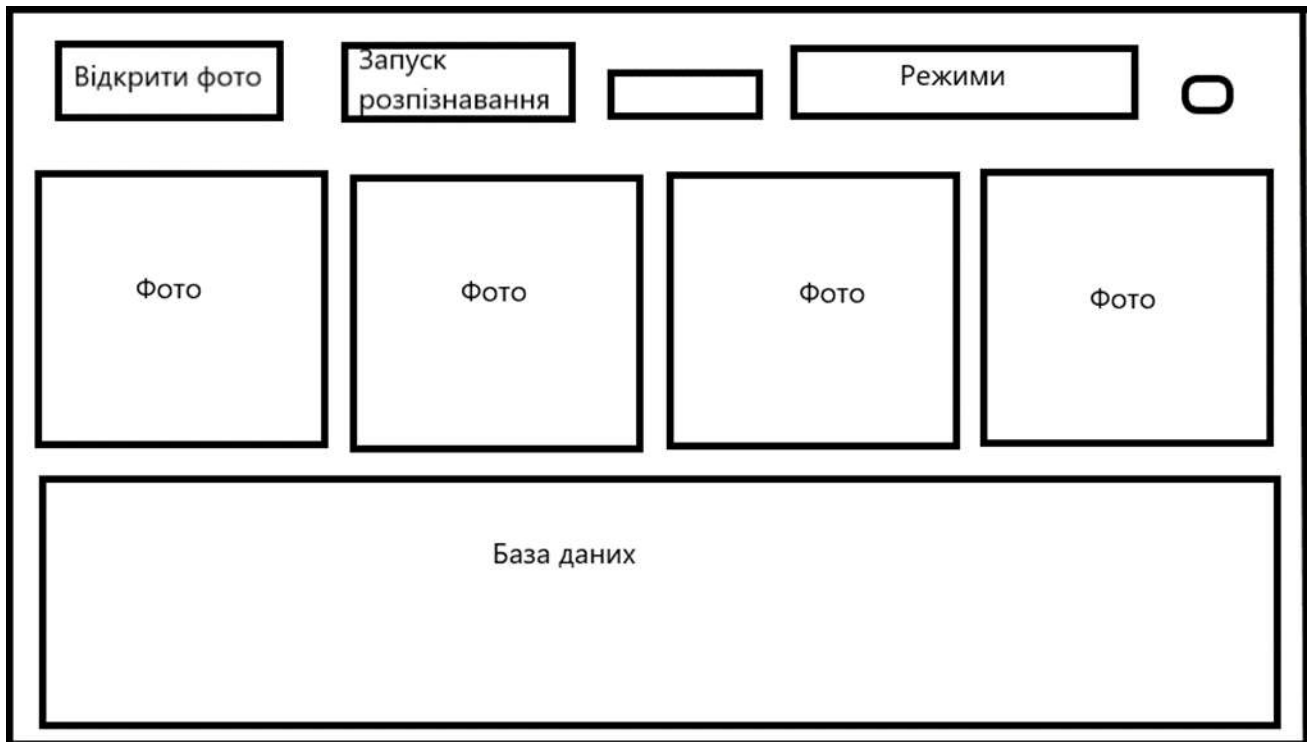


Рисунок 2.3 – Другий схематичний вигляд першої сторінки програми.

Також було створено схематичний вигляд другої сторінки головного меню, для роботи з БД(База даних), було створено приблизну схему (рисунок 2.4), з підписами назв елементів.



Рисунок 2.4 – Схематичний вигляд другої сторінки програми.

Схематичний вигляд був створений тільки як приклад можливого вигляду. Так що вигляд може змінюватися під час створення програми.

### 2.3 Розробка архітектури нейронної мережі

Для реалізації методу автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями використовуються інструменти OpenCV та Tesseract. У цьому розділі буде описано архітектуру нейронних мереж та інших алгоритмів штучного інтелекту, які застосовуються для розпізнавання обличчя водіїв і тексту на номерних знаках автомобілів.

Використання OpenCV для розпізнавання обличчя та номерних знаків.

OpenCV — це бібліотека для комп'ютерного зору, яка дозволяє вирішувати широкий спектр задач, включаючи пошук об'єктів на зображеннях. Для розпізнавання обличчя і номерних знаків використовуються каскадні класифікатори Хаара та глибокі нейронні мережі (DNN). Архітектура системи розпізнавання обличчя та номерних знаків за допомогою OpenCV складається з таких компонентів:

#### 1. Вхідні дані:

- розмірність вхідних даних: зображення, розмір якого може варіюватися залежно від роздільної здатності камери, наприклад, зображення може мати розмір 1024x768 пікселів;

- формат даних: RGB або Grayscale зображення.

#### 2. Попередня обробка зображення:

- перетворення зображення у відтінки сірого;
- використання гаусового розмиття для зменшення шуму;
- нормалізація зображення для покращення якості визначення.

#### 3. Визначення номерних знаків:

- модель є каскадним класифікатором Хаара або DNN для визначення об'єктів на зображенні;

- процес є методом який шукає області, де може знаходитись номерний знак, визначає точні межі знаку і вирізає частину зображення з номерним знаком;

- розмірністю вихідних даних є координати прямокутника, який обмежує номерний знак (x, y, ширина, висота).

#### 4. Пошук обличчя:

- модель є каскадним класифікатор Хаара або DNN для пошуку обличчя на зображенні;

- процес є методом який знаходить області, де може знаходитися обличчя, визначає точні межі обличчя і вирізає частину зображення з обличчям;

- розмірністю вихідних даних є координати прямокутника, який обмежує обличчя (x, y, ширина, висота).

Використання Tesseract для розпізнавання тексту на номерних знаках.

Tesseract – це система оптичного розпізнавання тексту (OCR), яка використовується для розпізнавання тексту на номерних знаках автомобілів. Архітектура системи розпізнавання тексту за допомогою Tesseract складається з таких компонентів:

##### 1. Вхідні дані:

- розмірність вхідних даних: зображення номерного номера, вирізане з основного зображення. Мінімальний розмір 200x40 пікселів;

- формат даних: Grayscale зображення для покращення контрасту.

##### 2. Попередня обробка зображення:

- бінаризація зображення для покращення контрасту між текстом і фоном;

- видалення шуму за допомогою морфологічних операцій.

##### 3. Розпізнавання тексту:

- модель є Tesseract OCR моделлю, яка тренована на алфавіті та цифрах, які відповідають номерним знакам країни;

- процес виконує визначення меж тексту на зображенні, розпізнавання символів та формування текстового рядка;

– розмірністю вихідних даних є текстовий рядок, що містить символи номерного номера.

Інтеграція компонентів в загальну систему. Загальна система автоматичного створення бази даних водіїв та номерів автомобілів складається з наступних компонентів:

Вхідний шар:

- отримання зображення від камери або іншого джерела;
- попередня обробка зображення;
- перетворення зображення у формат, придатний для подальшої обробки (відтінки сірого, нормалізація).

Пошук та розпізнавання:

- використання OpenCV для пошуку номерних знаків та облич;
- використання Tesseract для розпізнавання тексту на номерних знаках.

Вихідні дані:

- збереження розпізнаних номерів та облич у базу даних;
- вивід результатів на екран для перевірки та підтвердження.

Розмірність вхідних та вихідних даних

- вхідні дані це зображення (розмір: мінімум 512x512 пікселів, формат: RGB або Grayscale);
- вихідні дані (OpenCV) це координати обличчя (x, y, ширина, висота), координати номерного знаку (x, y, ширина, висота).
- вихідні дані (Tesseract) це розпізнаний текст (номер автомобіля).

Наведена архітектура забезпечує ефективне розпізнавання обличчя водіїв та номерних знаків автомобілів, що дозволяє автоматично створювати базу даних з отриманими даними.

## **2.4 Проектна архітектура системи та варіанти використання**

Відповідно до методу автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями, було спроектовано відповідну проектну

архітектуру системи, зображену на рисунку 2.5, яка містить в собі наступні підсистеми:

- підсистема роботи з базою даних;
- підсистема обробки зображень;
- підсистема розпізнання номерів;
- підсистема розпізнання обличчя.

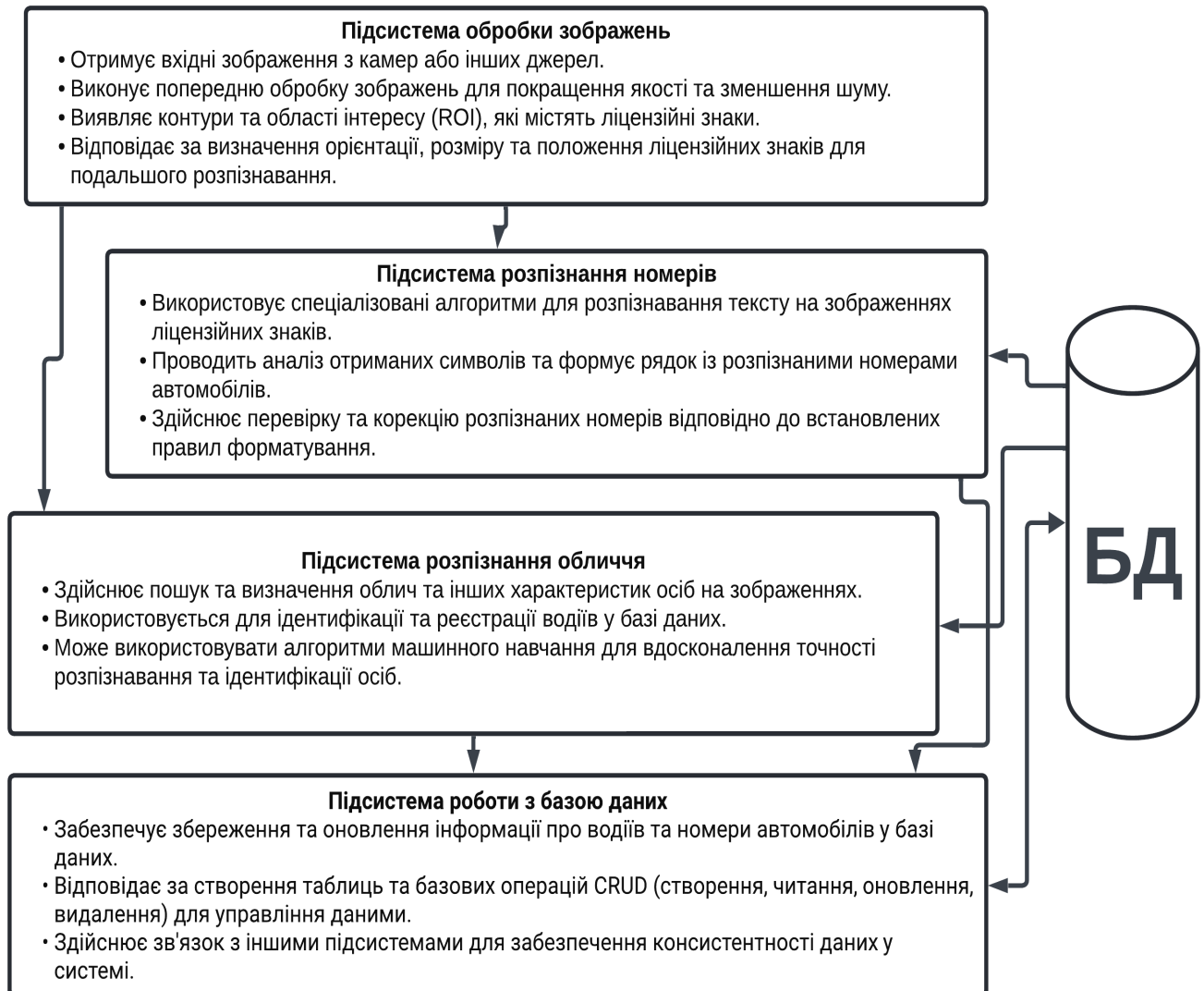


Рисунок 2.5 – Схема проектної архітектури процесу автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями

Підсистема роботи з базою даних виконує такі функції роботи:

- забезпечує збереження та оновлення інформації про водіїв та номери автомобілів у базі даних;

- відповідає за створення таблиць та базових операцій CRUD (створення, читання, оновлення, видалення) для управління даними;

- здійснює зв'язок з іншими підсистемами для забезпечення консистентності даних у системі.

Підсистема обробки зображень виконує такі функції роботи:

- отримує вхідні зображення з камер або інших джерел;
- виконує попередню обробку зображень для покращення якості та зменшення шуму;

- виявляє контури та області інтересу, які містять номерні знаки;
- відповідає за визначення орієнтації, розміру та положення номерних знаків для подальшого розпізнавання.

Підсистема розпізнавання номерів виконує такі функції роботи:

- використовує спеціалізовані алгоритми для розпізнавання тексту на зображеннях номерних знаків;

- проводить аналіз отриманих символів та формує рядок із розпізнаними номерами автомобілів;

- здійснює перевірку та корекцію розпізнаних номерів відповідно до встановлених правил форматування.

Підсистема розпізнавання обличчя виконує такі функції роботи:

- здійснює пошук та визначення облич та інших характеристик осіб на зображеннях;

- використовується для ідентифікації та реєстрації водіїв у базі даних;

- може використовувати алгоритми машинного навчання для вдосконалення точності розпізнавання та ідентифікації осіб.

Щоб показати як користувачі можуть взаємодіяти із програмою було створено діаграму варіантів використання. В якій пояснюється взаємодія програми за допомогою так званих варіантів використання, діаграма варіантів використання програми зображена на рисунку 2.6 та даний її короткий опис.

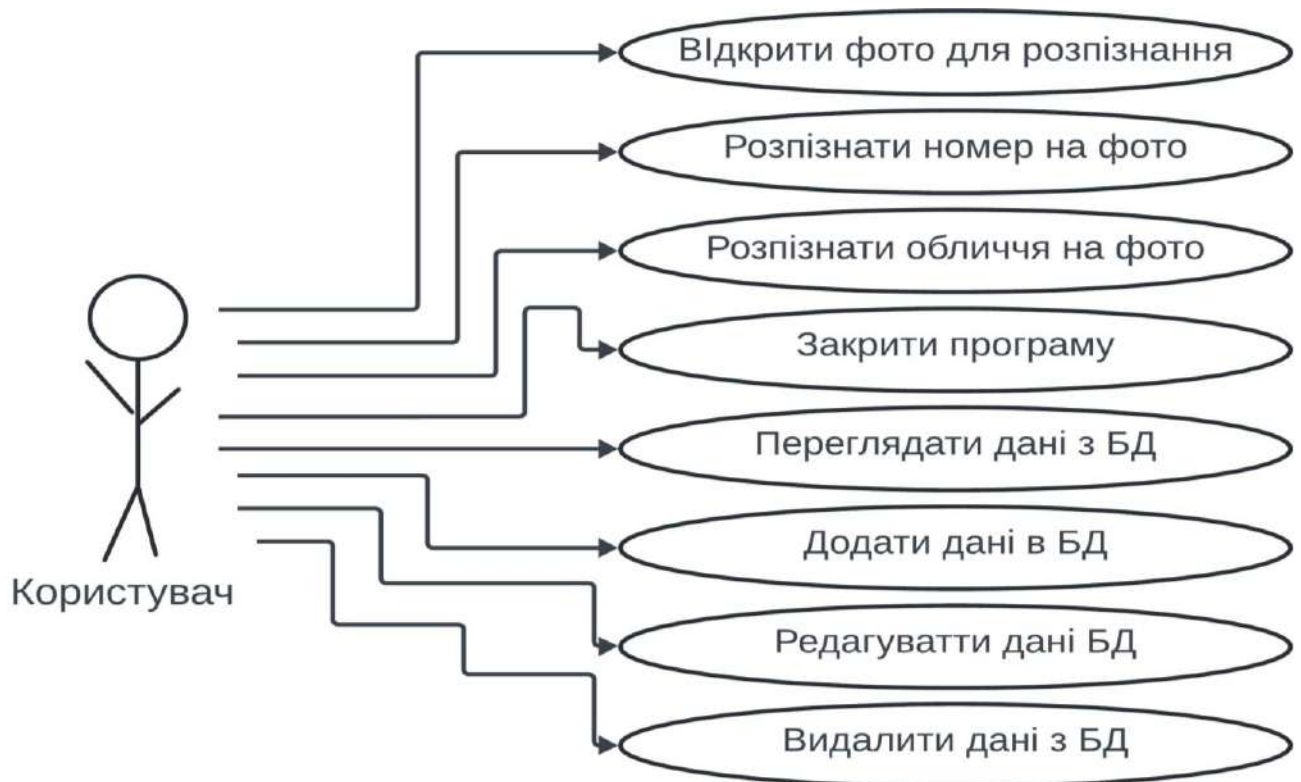


Рисунок 2.6 – Діаграма варіантів використання

При вході в програму відвідувач стає "Користувачем". "Користувач" потрапляє у головне вікно програми, яке містить кілька важливих дій, доступних для взаємодії, які наведені далі.

1. «Відкрити фото для розпізнання»: Після натиснення кнопки "Відкрити фото" відкривається файловий провідник. У цьому провіднику користувач має можливість вибрати фотографію, яку потрібно завантажити для подальшого розпізнання. Після вибору фото, зображення з'являється в основному вікні програми для перегляду.

2. «Розпізнати номер на фото»: Натиснувши кнопку "Запуск розпізнання" для номера, програма автоматично аналізує завантажене зображення, визначає область, де розташований номерний знак автомобіля, і розпізнає текст на цьому знаку. Після успішного розпізнання, номер відображається на екрані для перевірки користувачем.

3. «Розпізнати обличчя на фото»: Після натиснення кнопки "Запуск розпізнання" для обличчя, програма аналізує завантажене фото, виявляє обличчя людей на зображенні та виділяє їх. Обличчя, знайдені на фото, відображаються в

окремій області на екрані, щоб користувач міг перевірити результати розпізнавання.

4. «Переглянути дані з БД»: Натиснувши кнопку "SELECT", користувач може переглянути всі записи, що містяться в базі даних. Всі дані з таблиці бази даних виводяться на екран у зручному форматі, що дозволяє легко аналізувати збережену інформацію.

5. «Видалити дані з БД»: Натиснувши кнопку "DELETE", користувач може видалити певний запис з таблиці бази даних.

6. «Редагувати дані з БД»: Натиснувши кнопку "EDIT", користувач має можливість змінити певний запис у таблиці бази даних. Програма виведе попередні дані в які користувач може внести необхідні зміни до обраного запису. Після підтвердження змін, оновлені дані зберігаються в базі даних.

7. «Додати дані в БД»: Натиснувши кнопку "INSERT", користувач може додати новий запис у таблицю бази даних. Програма прочитає поля, де користувач заповнив всі необхідні поля для нового запису. Після підтвердження, новий запис додається до бази даних і відображається у загальному списку записів.

Ці функціональні можливості забезпечують повний цикл обробки даних, починаючи від завантаження та розпізнавання фотографій, і закінчуючи управлінням записами в базі даних, що робить програму зручною та ефективною для користувача.

## **2.5 Структура системи та проєктування бази даних**

На рисунку 2.7 зображена ER-діаграма для методу, який автоматично створює базу даних водіїв та номерів автомобілів за зображеннями. Ця схема відображає структуру даних та взаємозв'язки між сутностями цієї системи.

У цій ER-діаграмі подано 5 сутностей зі своїми атрибутами, які їх характеризують. Сутність "Водій" має атрибути: ім'я, прізвище, по батькові, фото, стать, телефон та адресу. "Автомобіль" включає атрибути: номер, фото,

власника та тип. Сутність "Тип автомобіля" містить атрибут "назва типу". "Стать" включає атрибут "назва статі". Сутність "Журнал проїздів" містить атрибути: номер автомобіля, фото номера, фото автомобіля, фото водія, ID водія, ID автомобіля та дату та час проїзду.

Також між об'єктами існують такі зв'язки:

1. Володіння водієм автомобілем: Кожен водій може володіти одним або кількома автомобілями.

2. Власник автомобіля: Кожен автомобіль має одного або кількох власників.

3. Стать водія: Кожен водій має тільки одну стать, але одна і та сама стать може бути в багатьох водіїв.

4. Тип автомобіля: Кожен автомобіль має тільки один тип, але один і той самий тип може бути в багатьох автомобілях.

5. Журнал проїздів машин: Журнал проїздів містить інформацію про проїзди, які включають багато водіїв та їх автомобілів.

Ця ER-діаграма є важливим інструментом для розробки бази даних та забезпечення правильної структури даних.

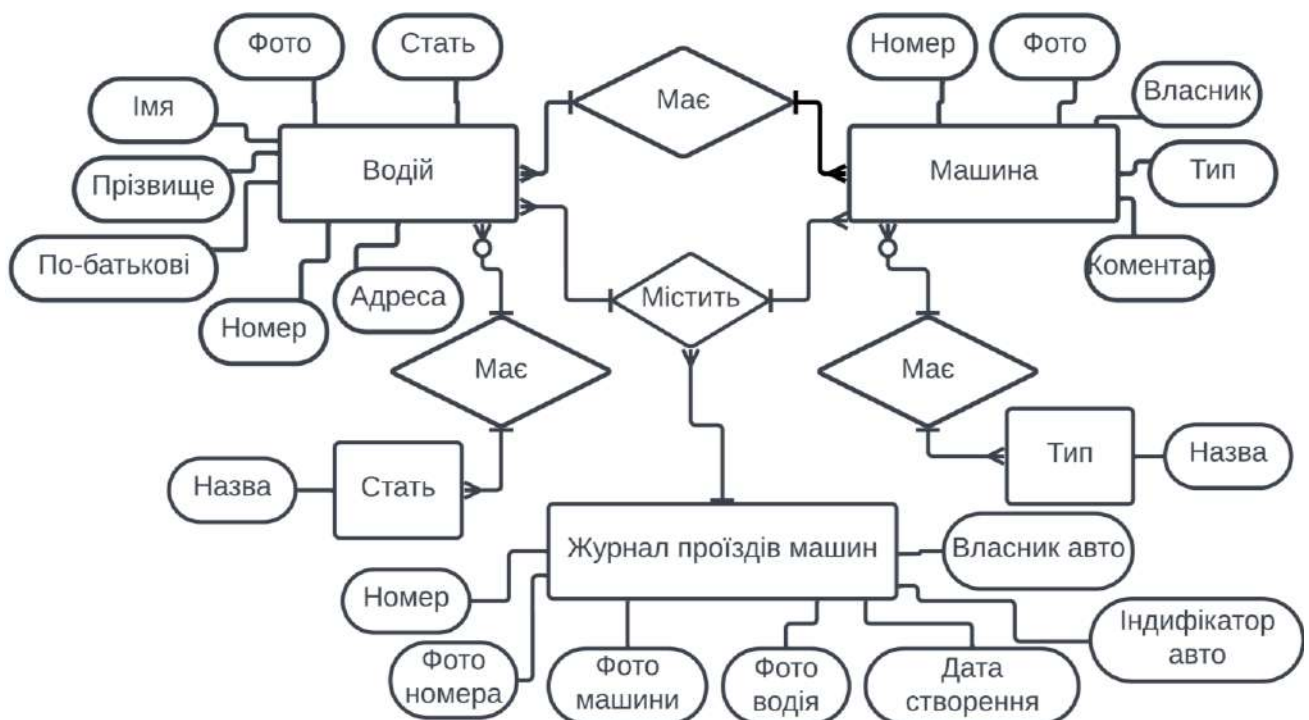


Рисунок 2.7 – ER-діаграма

Для розробки структури бази даних методу автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями, необхідно визначити сутності предметної області. Кожна із сутностей має набір властивостей, які її характеризують та зберігаються в окремих таблицях бази даних.

На рисунку 2.8 зображено даталогічну модель бази даних методу автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями. Створена база даних включає наступні таблиці: Sexes, Vehicles\_type, Driver, Vehicle та Vehicle\_log. Кожна таблиця призначена для зберігання даних про окрему сутність, що дозволяє забезпечити структурованість і організованість даних у системі.

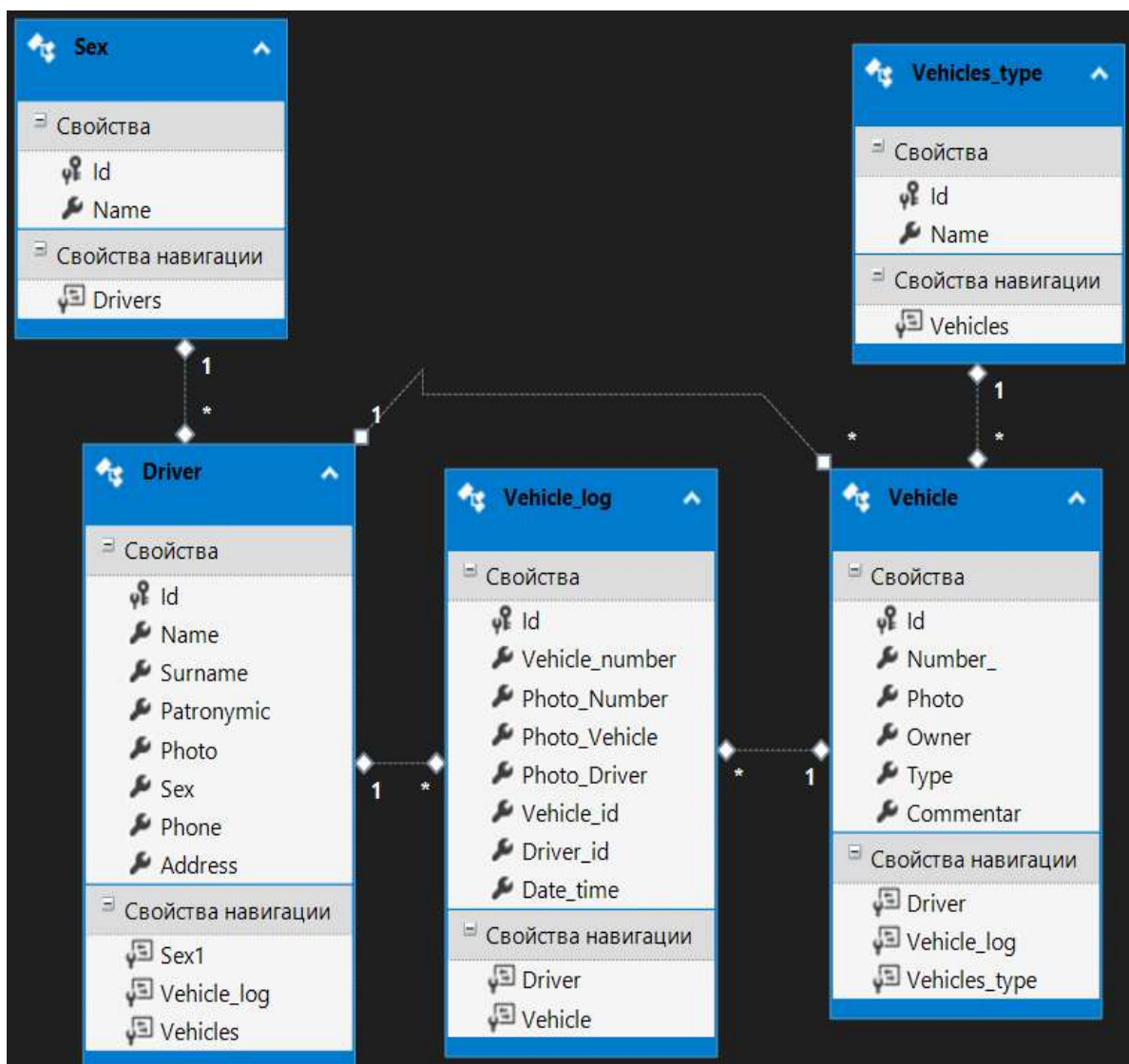


Рисунок 2.8 – Даталогічна модель бази даних методу з автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями

Структура бази даних проектується таким чином, щоб забезпечити взаємозв'язок між таблицями. Наприклад, таблиці Driver та Vehicle мають зв'язок через поле Driver\_ID, що дозволяє зв'язувати інформацію про водія з його транспортними засобами. Таблиця Vehicle\_log пов'язана з таблицею Vehicle через поле Vehicle\_ID, що забезпечує відстеження подій для кожного транспортного засобу.

Таблиця «Vehicle\_log» призначена для збереження даних після аналізу номера та обличчя водія.

Атрибутами таблиці є: Id, Vehicle\_number, Photo\_Number, Photo\_Vehicle, Photo\_Driver, Vehicle\_id, Driver\_id, Date\_time (Таблиця 2.1).

Таблиця 2.1 – Атрибути таблиці «Vehicle\_log»

№ п/п	Назва	Тип даних	Опис
1	Id	int	Первинний ключ. Унікальний ідентифікатор визначення.
2	Vehicle_number	nvarchar(10)	Текст. Номер машини.
3	Photo_Number	nvarchar(255)	Текст. Посилання на фото номера.
4	Photo_Vehicle	nvarchar(255)	Текст. Посилання на фото машини.
5	Photo_Driver	nvarchar(255)	Текст. Посилання на фото водія.
6	Vehicle_id	int	Вторинний ключ. Посилання на запис із таблиці «Vehicle». Використовується для зв'язку з машиною.
7	Driver_id	int	Вторинний ключ. Посилання на запис із таблиці «Driver». Використовується для зв'язку власником авто.
8	Date_time		Дата та час створення запису.

Таблиця «Vehicle» призначена для збереження даних про машину.

Атрибутами таблиці є: Id, Number, Photo, Owner, Type, Commentar (Таблиця 2.2).

Таблиця 2.2 – Атрибути таблиці «Vehicle»

№ п/п	Назва	Тип даних	Опис
1	Id	int	Первинний ключ. Унікальний ідентифікатор визначення.
2	Number	nvarchar(10)	Текст. Номер машини.
3	Photo	nvarchar(255)	Текст. Посилання на фото машини.
4	Owner	int	Вторинний ключ. Посилання на запис із таблиці «Driver». Використовується для зв'язку з власником машини.
5	Type	int	Вторинний ключ. Посилання на запис із таблиці «Vehicles_type». Використовується для зв'язку з типом машини.
6	Commentar	nvarchar(MAX)	Текст. Коментар.

Таблиця «Driver» призначена для збереження даних про водія/власника.

Атрибутами таблиці є: Id, Name, Surname, Patronymic, Photo, Sex, Phone, Address (Таблиця 2.3).

Таблиця 2.3 – Атрибути таблиці «Driver»

№ п/п	Назва	Тип даних	Опис
1	Id	int	Первинний ключ. Унікальний ідентифікатор визначення.
2	Name	nvarchar(50)	Текст. Ім'я.

3	Surname	nvarchar(50)	Текст. Прізвище.
4	Patronymic	nvarchar(50)	Текст. По батькові.
5	Photo	nvarchar(255)	Текст. Посилання на фото водія.
6	Sex	int	Вторинний ключ. Посилання на запис із таблиці «Sex». Використовується для зв'язку з статю.
7	Phone	nvarchar(50)	Текст. Номер телефона
8	Address	nvarchar(255)	Текст. Адреса проживання.

Таблиця «Sex» призначена для збереження даних про стать водія/власника.

Атрибути таблиці є: Id, Name (Таблиця 2.4).

Таблиця 2.4 – Атрибути таблиці «Sexes»

№ п/п	Назва	Тип даних	Опис
1	Id	int	Первинний ключ. Унікальний ідентифікатор визначення.
2	Name	nvarchar(50)	Текст. Назва статі.

Таблиця «Vehicles\_type» призначена для збереження даних про тип машини.

Атрибути таблиці є: Id, (Таблиця 2.5).

Таблиця 2.5 – Атрибути таблиці «Vehicles\_type»

№ п/п	Назва	Тип даних	Опис
1	Id	int	Первинний ключ. Унікальний ідентифікатор визначення.
2	Name	nvarchar(50)	Текст. Назва типу машини.

## 2.6 Підготовка робочих вхідних даних для системи

Для забезпечення якісної роботи системи "NumberFaceIdentifier" необхідно підготувати робочі вхідні дані, які включають зображення номерних знаків та облич. Основним джерелом таких даних є спеціалізовані дата сети, які забезпечують різноманітність та реалістичність вхідних даних, що важливо для тренування та тестування системи.

Дані для роботи системи формуються з декількох джерел:

- публічні дата сети: Відкриті та доступні для завантаження дата сети, які містять зображення номерних знаків та облич. Ці дата сети часто використовуються для тренування моделей машинного навчання та перевірки їх точності;

- власні зображення: Зображення, зібрані спеціально для цього проєкту, включаючи фотографії номерних знаків та облич, зроблені в різних умовах освітлення та з різних кутів;

- комбіновані джерела: Поєднання публічних та власних зображень для створення більш різноманітного та повного набору даних.

Використані публічні дата сети:

1. Stanford Cars Dataset [20] це дата сет який містить 16,185 зображень автомобілів, які можна використовувати для виділення та розпізнавання номерних знаків.

2. LFW (Labeled Faces in the Wild) Dataset [21] це дата сет який містить понад 13,000 зображень облич 5,749 різних осіб, знятих у реальних умовах.

Власні зображення для проєкту були зібрані за допомогою пошуку в відкритих джерелах. Ці зображення дозволяють перевірити роботу системи в реальних умовах, відмінних від умов які є в публічних дата сетах.

Використання публічних та власних дата сетів забезпечує різноманітність та реалістичність даних, що підвищує точність та ефективність системи. Поєднання зображень із різних джерел дозволяє створити повний та

збалансований набір даних, необхідний для успішного тренування та тестування програмного продукту.

## **2.7 Особливості використання спеціалізованих програмних компонентів**

У процесі розробки програмного забезпечення "NumberFaceIdentifier" використовувалися спеціалізовані бібліотеки EmguCV та Tesseract. Ці бібліотеки забезпечують необхідну функціональність для розпізнавання номерних знаків та облич на зображеннях, що є ключовою частиною системи.

EmguCV [22] – це обгортка для OpenCV, яка дозволяє використовувати потужні можливості бібліотеки комп'ютерного зору OpenCV у середовищі .NET, включаючи C#. Вона забезпечує зручний інтерфейс для роботи з обробкою зображень, виявленням об'єктів, розпізнаванням облич та іншими задачами.

EmguCV використовується для попередньої обробки зображень, фільтрації шуму, виявлення контурів та інших операцій, які підвищують точність розпізнавання номерних знаків та облич.

EmguCV планується використовувати для виконання всіх необхідних операцій з обробки зображень, таких як перетворення у відтінки сірого, порогове значення, виявлення контурів, та морфологічні перетворення. Це дозволяє підготувати зображення для подальшого розпізнавання тексту та облич.

Tesseract[23] – це оптичний розпізнавач символів (OCR), який дозволяє витягувати текст із зображень. Ця бібліотека забезпечує високу точність розпізнавання тексту та підтримує багато мов.

Tesseract використовується для розпізнавання тексту на зображеннях номерних знаків. Це досягається шляхом передачі обробленого зображення у функцію розпізнавання, яка повертає текст.

Tesseract планується використовувати для розпізнавання тексту з номерних знаків, які попередньо обробляються за допомогою EmguCV. Після

обробки зображення EmguCV, Tesseract дозволяє витягнути текстову інформацію, яка є ключовою для ідентифікації номерних знаків.

Використання бібліотек EmguCV та Tesseract дозволяє забезпечити високу точність та ефективність у процесі розпізнавання номерних знаків та облич у програмному продукті "NumberFaceIdentifier". EmguCV використовується для попередньої обробки зображень та підготовки їх до розпізнавання, тоді як Tesseract відповідає за безпосереднє витягування тексту з зображень. Такий підхід забезпечує комплексне та ефективне рішення для поставлених завдань.

## 2.8 Оцінювання ефективності розроблених методів

Оцінювання ефективності розроблених методів автоматичного створення бази даних водіїв та номерів автомобілів на основі зображень буде проведено в рамках трьох досліджень: визначення номера, визначення обличчя та визначення номера й обличчя водія одночасно. Для цього використовувалася методика "рейтингового оцінювання" [24] з бальною системою.

Для оцінки результатів роботи системи буде використовуватися бальна система, яка передбачає присвоєння результату однієї з наступних оцінок:

- 0 балів: не вдалося знайти номер/обличчя;
- 1 бал: знайдено номер/обличчя, але з проблемами;
- 2 бали: знайдено правильно номер/обличчя, але неправильно виведено текст номера;
- 3 бали: все виконано правильно.

Перше дослідження «Визначення номера»:

– кількість зображень для проведення дослідження буде від 20 до 40 штук;

– по кожному зображенні буде визначатися, чи вдалося знайти номер та правильно розпізнати текст, і результати фіксуватимуться за трьох бальною системою.

Друге дослідження «Визначення обличчя»:

- кількість зображень для проведення дослідження буде від 20 до 40 штук;
- по кожного зображені буде визначатися, чи вдалося знайти обличчя, і результати фіксуватимуться за двох бальною системою.

Третє дослідження «Визначення номера та обличчя»:

Кількість зображень:

- кількість зображень для проведення дослідження буде від 20 до 40 штук;
- по кожного зображені буде визначатися, чи вдалося знайти номер, чи вдалося знайти обличчя та, чи правильно розпізнався текст номеру, і результати фіксуватимуться за трьох бальною системою.

## **2.9 Висновки до розділу 2**

У цьому розділі розроблено метод автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями, було розроблено моделі та методи для автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями. Створено та автоматизовано обробку потоків даних, спроектовано функціональну структуру методів системи та розроблено архітектуру системи та взаємозв'язок компонентів.

З метою подальшої роботи було виконані наступні завдання.

- Розроблено архітектури нейронної мережі: Проведено розробку та оптимізацію архітектури нейронної мережі для розпізнавання тексту на номерних знаках автомобілів та розпізнавання.
- Спроектовано базу даних програмної реалізації: Спроектовано та реалізовано базу даних для збереження інформації про водіїв та номери автомобілів, а також результати розпізнавання номерних знаків та облич.

– Підготовлено робочі вхідні дані для системи: Проведено підготовку та очищення вхідних даних, необхідних для тренування та тестування нейронної мережі.

– Підготовлена методика для оцінювання ефективності розроблених методів: Створено методику та дано її опис для того щоб показати як буде проводитися оцінювання ефективності.

– Розроблено застосунок: Реалізовано застосунок. Застосунок забезпечує зручний інтерфейс для взаємодії з користувачем та виконання сценаріїв.

Також після повної реалізації застосунка буде проведене функціональне та модульне тестування розробленого застосунку для перевірки коректності його роботи та виявлення можливих помилок.

Ці завдання спрямовані на подальше вдосконалення системи та перевірку її ефективності перед впровадженням у практику.

## **Розділ 3 Програмна реалізація методу автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями**

### **3.1 Визначення шляхів дослідження та засобів створення програмного забезпечення**

Для визначення шляхів дослідження та засобів створення програмного забезпечення було розглянуто кілька аспектів, таких як методологія розробки, вибір технологій, інструментів та мов програмування. Далі наведено чотири кроки, які були виконані для визначення цих шляхів.

1. Вибір технологій: Був проведений аналіз технологічного стека для написання програми. Першою ціллю було визначення платформи для написання програми та мова програмування, другою дією було вирішити яким редактором коду та в якому середовищем програмування буде створюватися програма, останньою дією було визначити елементи для роботи з базою даних.

2. Розробка: Під час розробки було використано вибрану мову програмування та інструменти розробки для створення програмного коду, який відповідає вимогам та функціональності, визначеним на етапі аналізу вимог. Оскільки програма містить в собі графічний інтерфейс користувача, то також було створено елементи інтерфейсу для взаємодії з користувачем які відповідають вимогам та функціям, які були визначеними на етапі аналізу вимог

3. Тестування: Буде проведено модульне, інтеграційне та системне тестування з використанням Unit Test або Test Case. Модульне тестування буде відбуватися під час створення кожного модуля. Інтеграційне тестування буде проводитися коли всі модулі які відповідають за велику функцію будуть написані та перевіренні. Системне тестування буде проведено в кінці розробки програми.

4. Дослідження: Буде проведено дослідження яке міститиме в собі ретельний аналіз вимог до програмного забезпечення, вибір технологій та інструментів для реалізації, а також визначення методів тестування. Кожен етап досліджень буде детально описаний, включаючи кроки, алгоритм обробки

результатів та критерії оцінки ефективності. Буде проведено три дослідження перше про точність визначення номера, друге про точність визначення обличчя, третє про точність визначення номера та обличчя.

### **3.2 Вибір засобів розробки програмної реалізації запропонованого методу**

Перед тим як розпочати розробку програмного продукту, був проведений аналіз варіантів щодо визначення платформи, середовища програмування, мови програмування, СКБД (Система управління базами даних), мови запитів, графічного рушія та редактора програмного коду.

Зараз як платформу для реалізації програми використовується персональний комп'ютер. Він забезпечує необхідну обчислювальну потужність, гнучкість налаштування та можливість швидкої реакції на зміни в програмі.

Обрання персонального комп'ютера як платформи створило сприятливі умови для комфортної та продуктивної розробки програми, водночас забезпечуючи необхідні ресурси та середовище для реалізації поставлених завдань.

В теперішній час після аналізу доступних варіантів мов програмування, які розглядалися для написання програми, та вибір яких був звужений до двох основних альтернатив: C# [25] та C++ [26], програма пишеться з допомогою мови програмування C#.

Оскільки після уважного зважання усіх факторів, та рішення було обрано C# через мій попередній досвід роботи з бібліотеками для машинного зору цією мовою програмування, а також завдяки доступності повної та точної документації для використовуваних бібліотек.

Програма створюється в середовищі програмування MS(Microsoft) Visual Studio [27] та в його редакторі коду для написання програм, такий вибір був зроблений на основі кількох вагомих переваг:

- MS Visual Studio є інтегрованим середовищем розробки, призначеним для мови C# та платформи .NET;
- воно надає широкий набір інструментів та можливостей, які полегшують роботу розробників;
- редактор коду в MS Visual Studio є потужним і зручним інструментом, який підтримує автодоповнення коду, відстеження помилок та інші корисні функції, що полегшують процес написання програм.

Для створення інтерфейсу на час написання програми використовується графічний Windows Forms [28] який був обраний через свою простоту та зручність у використанні та оскільки програма не передбачає використання складних графічних ефектів, Windows Forms надає достатньо функціональності для створення потрібного інтерфейсу без надмірної складності. А оскільки Windows Forms є стандартним для платформи .NET [29], що спрощує інтеграцію з іншими компонентами програми та забезпечує сумісність з іншими рішеннями на платформі .NET, то зміна на інший графічний рушій не планується.

Для запитів в базу даних використовується SQL [30] це обґрунтовується простотою та ефективністю. А оскільки SQL є стандартною мовою запитів для взаємодії з реляційними базами даних, а також вона відома своєю легкістю вивчення та застосування своєю оптимізацією та ефективністю роботи з базами даних з таких причин вона має менший вплив на ресурси системи під час виконання запитів, що зробило її ідеальним вибором для цієї програми.

Під час створення програмної реалізації продукту, після аналізу доступних варіантів СКБД, які розглядалися для використання, щоб зберегти дані роботи програми, та вибір між яких був звужений до двох основних альтернатив: Access [31] та MSSQL [32], т теперішній час програма зберігає дані з використанням СКБД MSSQL.

Щодо такого вибору системи керування базами даних, був проведений аналіз між Access і MSSQL це було важливим кроком у процесі розробки оскільки Access, як проста реляційна СКБД, може бути зручною для невеликих проєктів. Однак, для великих та складних програм, як моя, функціональності

Access виявилася недостатньо, MSSQL являє собою розвинуту СКБД, яка надає більший функціонал, підтримку високої завантаженості та різноманітні можливості для оптимізації та розширення бази даних, а також вона виявилася достатньою функціональною для моєї програми.

Отже, під час написання програми використовувався персональний комп'ютер як платформа для написання, мова програмування C# та її рідна для неї платформа .NET, редактором коду та середовищем програмування виступав MS Visual Studio, за роботу з базою даних відповідає мова запитів SQL та СКБД MSSQL, за графіку відповідає Windows Forms.

### 3.3 Структура та функціональне призначення програмних складових системи

На основі розробленої структури програми було створено діаграму класів [33], що використовувалася під час створення програми.

Розроблена діаграма зображена на рисунку 3.1 також нижче даний опис що робить кожен клас:

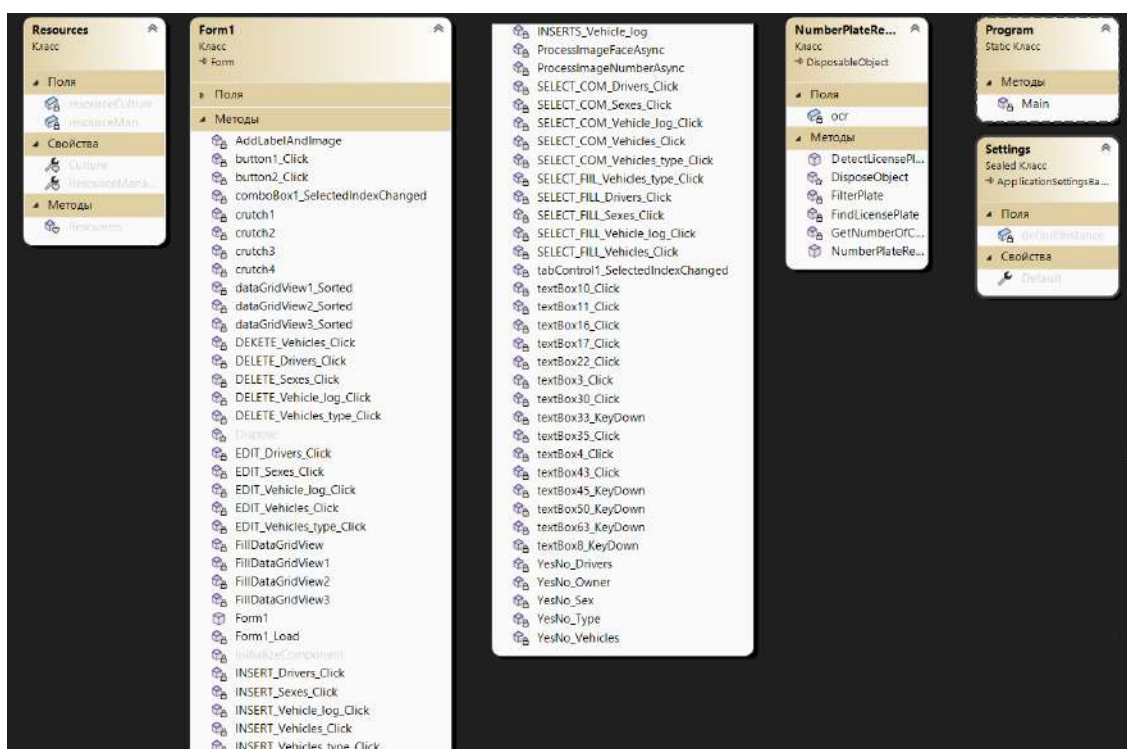


Рисунок 3.1 – Діаграма класів програмної реалізації методу

Форма Form1 являє собою головну форму для програми вона використовується для прийняття даних від користувача, виводу даних та роботи з базою даних в ній знаходяться такі методи як:

- метод AddLabelAndImage: Цей метод додає на головну форму поля для виводу тексту та зображень. Його викликається при ініціалізації форми або при необхідності додати нові елементи на форму;

- метод button1\_Click: Він відповідає за відкриття діалогового вікна провідника для вибору файлів, зокрема фотографій для подальшого розпізнавання. Обраний файл потім відображається на формі;

- метод button2\_Click: Цей метод ініціює розпізнавання номерного знака та обличчя на обраній фотографії, викликаючи відповідні асинхронні методи ProcessImageNumberAsync та ProcessImageFaceAsync;

- метод comboBox1\_SelectedIndexChanged: Він викликається при зміні значення в комбінованому списку, який відповідає за вибір точності пошуку та розпізнавання облич на фотографії;

- методи crutch1, crutch2, crutch3 та crutch4: Ці методи використовуються для відображення фотографій в DataGridView. Вони обробляють дані та відображають зображення у відповідних стовпцях;

- методи dataGridView1\_Sorted, dataGridView2\_Sorted та dataGridView3\_Sorted: Вони викликаються при сортуванні даних у DataGridView, щоб забезпечити правильне відображення даних у відсортованому порядку;

- методи DELETE\_(Назва таблиці)\_Click: Ці методи видаляють вибраний запис з відповідної таблиці у базі даних;

- методи EDIT\_(Назва таблиці)\_Click: Вони використовуються для редагування вибраного запису в таблиці у базі даних;

- методи INSERT\_(Назва таблиці)\_Click: Ці методи додають нові дані в відповідну таблицю у базі даних;

- методи `SELECT_COM_(Назва таблиці)_Click`: Вони виконують користувацький SQL-запит до відповідної таблиці та відображають результати на формі;

- методи `SELECT_FILL_(Назва таблиці)_Click`: Вони відображають усі дані з відповідної таблиці на формі;

- метод `tabControl1_SelectedIndexChange`: Він викликається при зміні вибраної вкладки у `TabControl` і виконує метод `SELECT_FILL_(Назва таблиці)_Click` відповідно до обраної таблиці;

- методи `textBox(Номер текс-бокса)_Click` та `textBox(Номер текс-бокса)_KeyDown`: Вони використовуються для взаємодії з текстовими полями на формі, зокрема для відкриття файлів та завантаження даних при зміні тексту;

- методи `YesNo_Vehicles`, `YesNo_Drivers`, `YesNo_Sex`, `YesNo_Owner` та `YesNo_Type`: Вони виконують перевірку наявності записів у відповідних таблицях за вказаним ідентифікатором та виводять відповідне повідомлення, якщо запис не знайдено.

Клас `NumberPlateRecognizer` являвся класом для роботи з фото та розпізнаванням тексту. Також в ньому знаходяться головні методи:

- метод `FilterPlate`: Цей метод призначений для видалення цифрового шуму з вирізаного фото з номером. Він використовується для очищення зображення перед розпізнаванням тексту, щоб покращити точність розпізнавання;

- метод `GetNumberOfChildren`: Ця функція перевіряє, чи є в потенційній області номерного знака хоча б один символ. Вона допомагає відфільтрувати непотрібні області та скорочує час розпізнавання;

- метод `FindLicensePlate`: Цей метод знаходить всі області на фото, де можуть знаходитися автомобільні номери. Крім того, він розпізнає текст на знайдених номерах, використовуючи інші методи, і зберігає результати для подальшого використання;

- метод `DetectLicensePlates`: Цей метод взаємодіє з інтерфейсом користувача, з якого отримує дані про фотографії та передає їх для

розпізнавання. Він також виконує перепідготовку даних, яка може бути необхідною перед викликом методу FindLicensePlate.

### 3.4 Особливості реалізації програмних складових системи

Програмні модулі є важливими осередками будь-якої програми, оскільки вони забезпечують функціональну роздільність та модульність коду.

Найголовнішими з таких модулів в моїй програмній реалізації методу автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями є:

- розпізнавання тексту;
- розпізнавання обличчя;
- вивід даних та створення полів для цього;
- запуск обробки фото;
- вибір та відкриття фото, запуск обробки фото;
- створення запису з фото машини, фото лиця та номером після визначення їх;
- робота з даними бази даних.

На рисунку 3.2 показано сторінка форми програми на сторінці визначення даних.

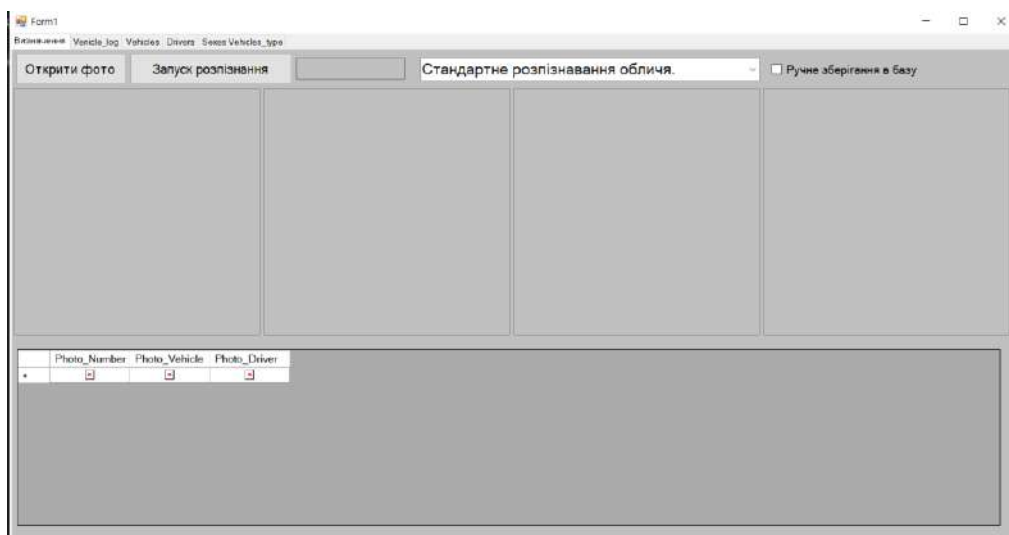


Рисунок 3.2 – Форма програми на сторінці визначення даних

Кнопка «Відкрити фото» виконує відкриття та завантаження фото для визначення: Ця функція викликається при натисканні на кнопку "Вибрати файл". Вона очищає зображення на всіх PictureBox і відкриває діалогове вікно для вибору файлу. Обране зображення потім відображається на першому PictureBox. Якщо вибраний файл не є зображенням, виводиться повідомлення про помилку.

Кнопка «Запуск розпізнавання» запускає код для визначення номера та лица за допомогою наступного програмного коду: Ця функція викликається при натисканні на кнопку "Розпізнати". Вона викликає асинхронні методи для обробки зображення номерного знака та обличчя. Після обробки зображення вона пропонує користувачеві створити запис в базі даних, якщо відмічено відповідний чек-бокс. Після цього вона вибирає останній запис з бази даних та відображає його у DataGridView.

Галочка «Ручне зберігання в базу» міняє спосіб збереження даних в базу даних: Ця функція викликається при зміні вибраної вкладки на TabControl. Залежно від обраної вкладки вона викликає відповідні функції для заповнення DataGridView даними з відповідної таблиці бази даних.

Метод ProcessImageFaceAsync виконує обробку зображення для пошуку облич на ньому. Ось опис його кроків:

1. Завантаження зображення з файлу, вказаного в аргументі image, у форматі Bitmap.
2. Перетворення зображення у формат Image<Bgr, byte>, який потрібний для використання методом DetectMultiScale класифікатора.
3. Використання класифікатора для виявлення облич на зображенні за допомогою методу DetectMultiScale. Результатом є масив прямокутників (Rectangle[]), які представляють знайдені обличчя.
4. Якщо обличчя знайдено (кількість прямокутників більше ніж 0), вибирається перше обличчя (faces[0]) і витягується відповідна область з оригінального зображення.

5. Створення окремого зображення обличчя (faceBitmap) з витягнутої області.

6. Малювання прямокутника на оригінальному зображенні, щоб виділити обличчя. Це робиться за допомогою методу DrawRectangle, де вказується колір та товщина лінії.

7. Виведення зображення обличчя на другий PictureBox (pictureBox2).

8. Якщо обличчя не знайдено, встановлення фотографії "0000.jpg" з каталогу "Photo\_Driver" як тимчасове зображення для відображення в другому PictureBox.

На рисунку 3.3 показано сторінка форми програми на сторінці редагування даних.

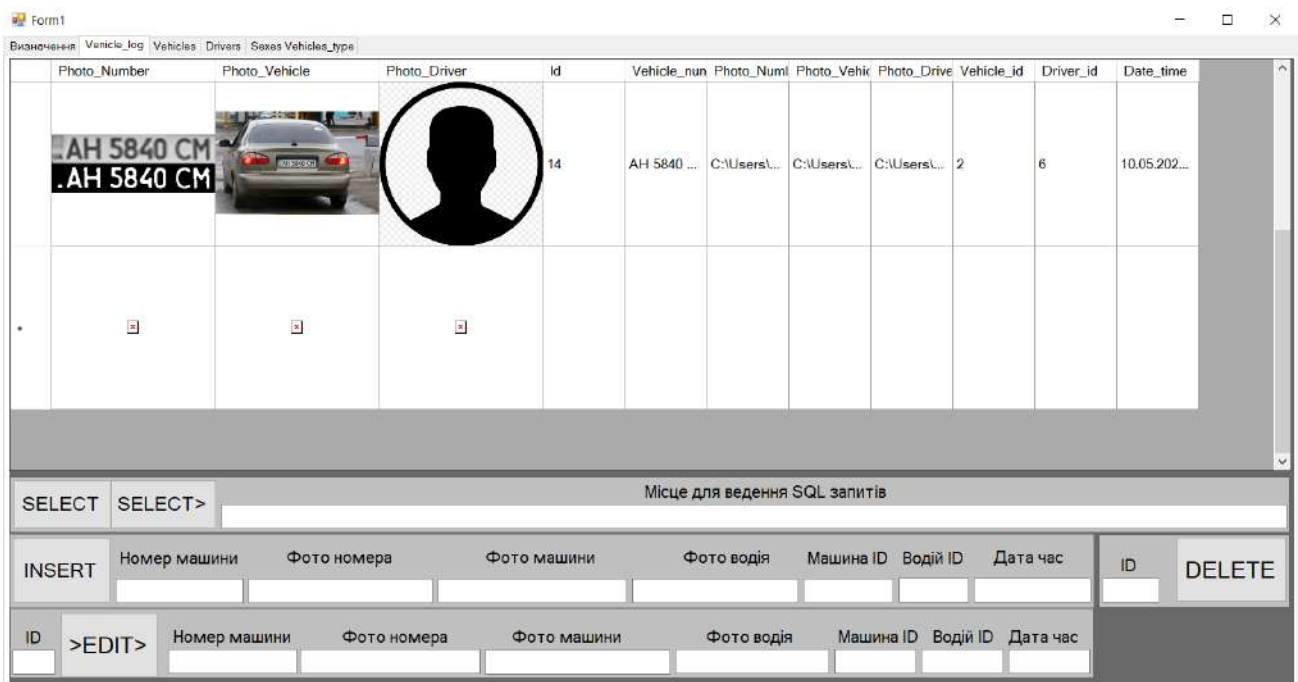


Рисунок 3.3 – Форма програми на сторінці редагування даних

Кнопка «SELECT» : Її функція викликається при натисканні кнопки із завантаженням записів з таблиці Vehicle\_log. Вона виконує запит до бази даних, витягуючи всі дані з таблиці Vehicle\_log, та відображає ці дані у DataGridView за допомогою функції FillDataGridView.

Кнопка «SELECT» : Її функція викликається при натисканні кнопки для виконання користувацького SQL-запиту до таблиці Vehicle\_log. Вона перевіряє, чи користувач ввів запит, а потім виконує цей запит за допомогою SqlDataAdapter і відображає результати у DataGridView.

Кнопка «INSERT» : Її функція викликається при натисканні кнопки для додавання нового запису до таблиці Vehicle\_log. Вона перевіряє наявність ідентифікаторів транспортного засобу та водія у відповідних таблицях, перевіряє коректність введених даних і додає новий запис до таблиці Vehicle\_log.

Кнопка «EDIT» : Її функція викликається при натисканні кнопки для редагування наявного запису в таблиці Vehicle\_log. Вона перевіряє наявність ідентифікаторів транспортного засобу та водія у відповідних таблицях, перевіряє коректність введених даних та оновлює відповідний запис в таблиці Vehicle\_log.

Кнопка «DELETE» : Її функція викликається при натисканні кнопки для видалення запису з таблиці Vehicle\_log. Вона перевіряє коректність введеного ідентифікатора запису та виконує видалення відповідного запису з таблиці Vehicle\_log.

Функція textBox(Номер текс-боксу)\_Click відповідає за обробку події натискання на текстовому полі textBox(Номер текс-боксу). Коли користувач натискає на це поле, викликається відкрите діалогове вікно вибору файлу за допомогою класу OpenFileDialog. Після вибору файлу і натискання кнопки "ОК" у діалоговому вікні, шлях до вибраного файлу зберігається в змінній selectedFile. Далі цей шлях присвоюється властивості Text текстового поля textBox(Номер текс-боксу)\_, щоб відобразити його у вікні програми.

В класі NumberPlateRecognizer являвся класом для роботи з фото та розпізнаванням тексту в ньому реалізовано чотири методи:

- метод FilterPlate являється методом який прибирає цифрові шуми з вирізаного фото з номером;
- метод GetNumberOfChildren який перевіряє чи є в можливому регіоні де є номер хоча б якийсь символ;

– метод `FindLicensePlate` який знаходить всі області на фото де є автомобільні номери, а також розпізнає текст на номерах;

– метод `DetectLicensePlates` який взаємодіє з класом `NumberPlateRecognizer` і отримувати дані які повертає клас після роботи, а також виконує пре підготовка даних перед запуском метода `FindLicensePlate`.

Метод `FilterPlate` призначений для фільтрації номерного знака, щоб покращити точність подальшого розпізнавання тексту. Далі наведено опис його кроків.

1. Створення порожнього чорно-білого зображення (`thresh`) за допомогою функції `Threshold`, яке слугуватиме маскою для фільтрації.

2. Застосування порогової фільтрації до вихідного зображення (`plate`), щоб отримати чорно-біле зображення, де елементи, які перевищують певний поріг інтенсивності, відображаються як білі, а решта як чорні.

3. Визначення розміру вихідного зображення номерного знака.

4. Створення порожнього зображення-маски (`plateMask`) того ж розміру, що і номерний знак.

5. Застосування детектора країв `Canny` до вихідного зображення, щоб знайти контури на зображенні номерного знака.

6. Пошук контурів на зображенні за допомогою функції `FindContours`. Кожен знайдений контур представлений як список точок, які утворюють його форму.

7. Перевірка кожного контуру: якщо висота прямокутника, обмеженого контуру, перевищує половину висоти зображення, то знижуємо розміри прямокутника та застосовуємо його як маску для виключення зображення з областей вище номерного знака.

8. Застосування маски до порогованого зображення, щоб відфільтрувати контури, які перебувають вище номерного знака.

9. Застосування операції ерозії та дилатації до фільтрованого зображення для видалення шуму та з'єднання дискретних елементів.

10. Повернення фільтрованого зображення для подальшого використання.

### 3.5 Тестування програмної реалізації метода та вимоги до розгортання

Тестування являється сукупністю дій, які містить початкові дані, що перевіряються з еталонними результатами, їх використовують для виявлення помилок у програмі.

Основну увагу при тестуванні необхідно приділити дотриманню принципів тестування для гарантування ефективності тестування. У більшості випадках тестування програм зводиться до підбору тестів та порівняння отриманих результатів з еталонними.

Для тестування гри було вибрано метод тестування за допомогою тест-кейсів[34].

Перший тест-кейс перевіряє можливість запуску програми (таблиця 3.1).

Таблиця 3.1 – Тест-кейс TS0001

<b>Тест-кейс ID:</b> TS0001	<b>Пріоритет:</b> 1	<b>Створено:</b> 15.05.2024, Скрипнюк О.Ю.
<b>Назва:</b> Запуск програми		
<b>Вхідні дані:</b> Відкрита папка з exe файлом програми		
<b>Кроки</b>	<b>Очікуваний результат</b>	
<ol style="list-style-type: none"> <li>Натиснути два рази правою кнопкою миші на «NumberFaceIdentifier.exe»</li> <li>Натиснути кнопку «ОК» на повідомленні</li> </ol>	<ol style="list-style-type: none"> <li>З'явилося повідомлення про успішне підключення</li> <li>Відкрилося вікно програми, програма запустилася</li> </ol>	

## Результат виконання тест-кейсу: пройдено успішно

В папці програми потрібно натиснути два рази правою кнопкою миші на «NumberFaceIdentifier.exe» (рисунок 3.4) після чого має відкритися повідомлення про успішне підключення в ньому треба натиснути кнопку «ОК», має відкритися вікно програми, що означає що програма запустилася (рисунок 3.5).

Имя	Дата изменения	Тип	Размер
concr140.dll	18.10.2023 3:41	Расширение при...	320 КБ
cvextern.dll	17.10.2023 15:20	Расширение при...	43 603 КБ
Database.mdf	15.05.2024 13:07	Файл "MDF"	8 192 КБ
Database_log.ldf	15.05.2024 13:07	Файл "LDF"	8 192 КБ
Emgu.CV.Bitmap.dll	18.10.2023 4:37	Расширение при...	24 КБ
Emgu.CV.dll	18.10.2023 3:59	Расширение при...	871 КБ
haarcascade_frontalface_alt.xml	22.04.2024 16:34	Microsoft Edge HT...	661 КБ
haarcascade_frontalface_alt_tree.xml	22.04.2024 15:28	Microsoft Edge HT...	2 627 КБ
haarcascade_frontalface_alt2.xml	22.04.2024 16:29	Microsoft Edge HT...	528 КБ
haarcascade_frontalface_default.xml	22.04.2024 16:35	Microsoft Edge HT...	909 КБ
Microsoft.Bcl.AsyncInterfaces.dll	18.10.2022 19:19	Расширение при...	27 КБ
msvc140.dll	18.10.2023 3:41	Расширение при...	565 КБ
msvc140_1.dll	18.10.2023 3:41	Расширение при...	35 КБ
msvc140_2.dll	18.10.2023 3:41	Расширение при...	261 КБ
msvc140_atomic_wait.dll	18.10.2023 3:41	Расширение при...	49 КБ
msvc140_codecvt_ids.dll	18.10.2023 3:41	Расширение при...	31 КБ
<b>NumberFaceIdentifier.exe</b>	15.05.2024 13:09	<b>Приложение</b>	<b>353 КБ</b>
NumberFaceIdentifier.exe.config	25.04.2024 14:27	Исходный файл С...	2 КБ
NumberFaceIdentifier.pdb	15.05.2024 13:09	Program Debug D...	116 КБ

Рисунок 3.4– Exe файл який потрібно запустити

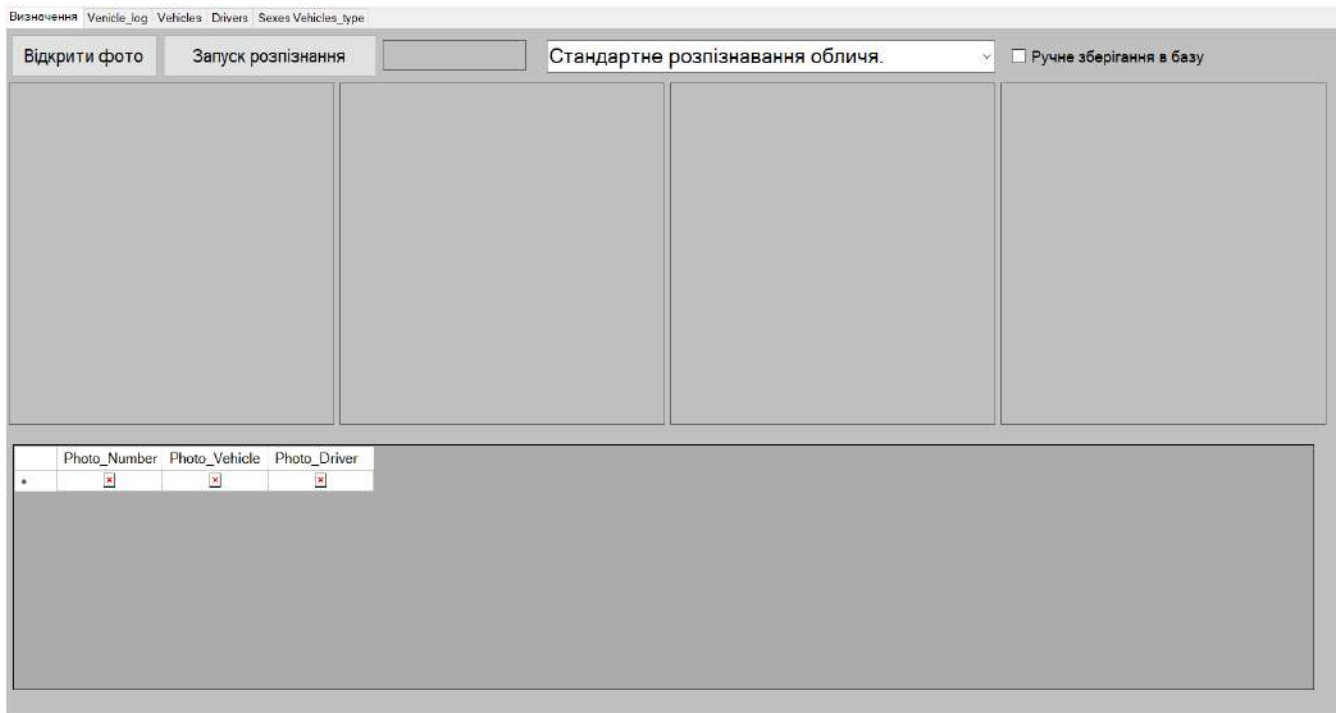


Рисунок 3.5 – Вікно програми

Другий тест-кейс перевіряє можливість відкриття фото (таблиця 3.2).

Таблиця 3.2 – Тест-кейс TS0002

<b>Тест-кейс ID:</b> TS0002	<b>Пріоритет:</b> 1	<b>Створено:</b> 15.05.2024, Скрипнюк О.Ю.
<b>Назва:</b> Відкриття файлу		
<b>Вхідні дані:</b> Програма запущена		
<b>Кроки</b>	<b>Очікуваний результат</b>	
3. Натиснути кнопку «Відкрити фото» 4. Побачити що відкрився провідник файлів 5. Відкрити файл фото	3. Кнопка «Відкрити файл» спрацювала 4. Провідник відкрився і в ньому видно файли 5. Файл фото відкрився і воно відобразилося зліва в програмі	
<b>Результат виконання тест-кейсу:</b> пройдено успішно		

На формі програми на пункті визначення потрібно натиснути кнопку «Відкрити фото» (рисунок 3.6) після чого має відкритися провідник (рисунок 3.7) в ньому треба відкрити файл фото, фото яке було відкрито має відобразитися в лівій частині програми (рисунок 3.8).

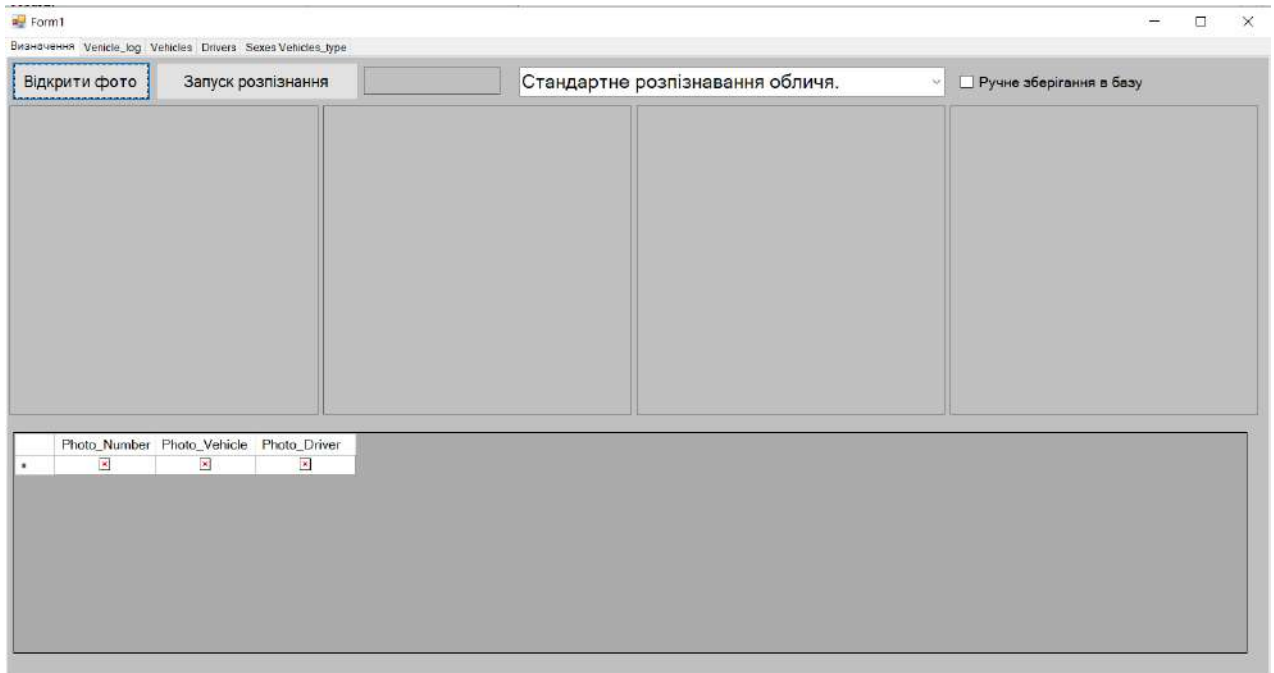


Рисунок 3.6 – Кнопка «Відкрити фото» на формі пошуку номера

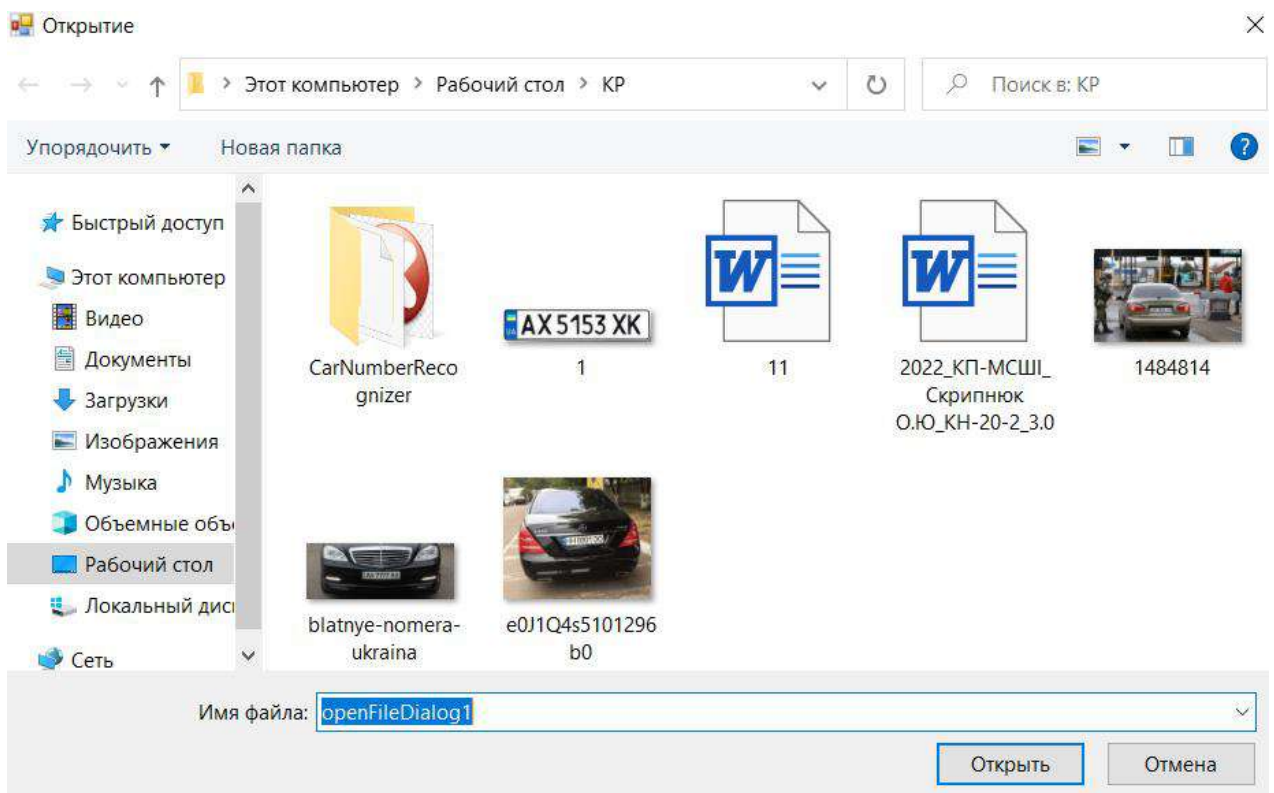


Рисунок 3.7 – Провідник для вибору файлів

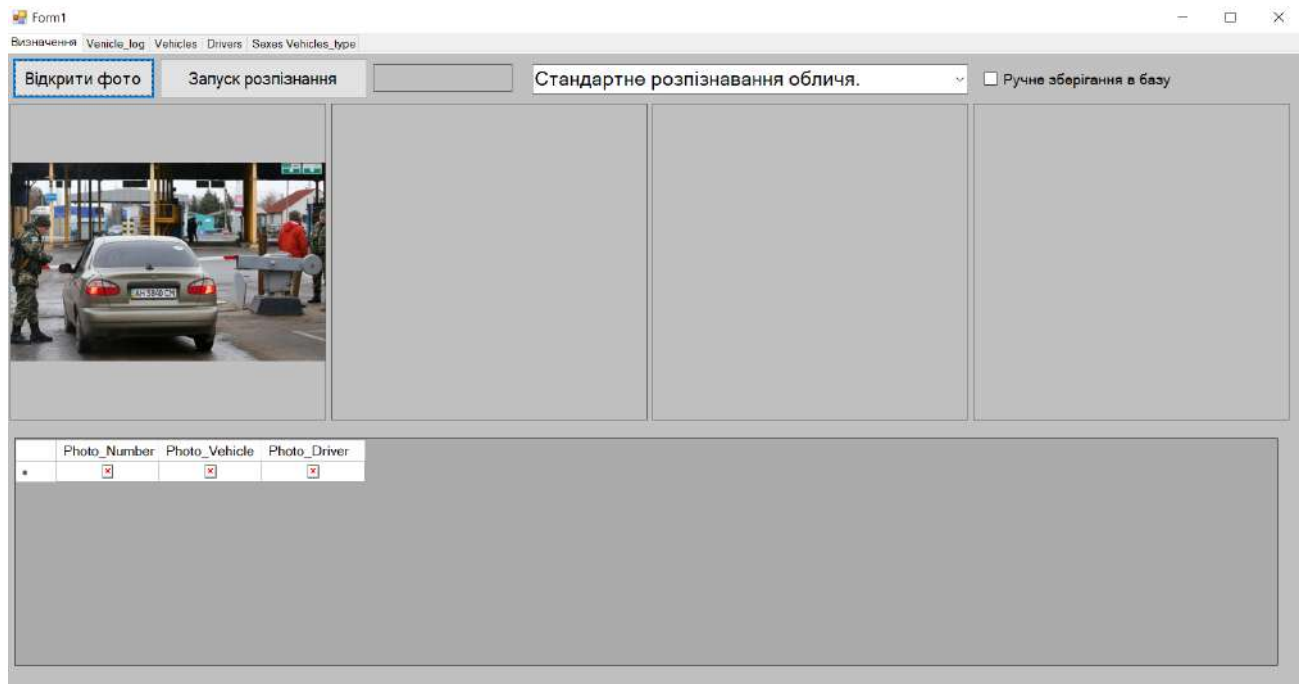


Рисунок 3.8 – Правильне відображення фото

Третій тест-кейс перевіряє можливість пошуку номера та обличчя (таблиця 3.3).

Таблиця 3.3 – Тест-кейс TS0003

<b>Тест-кейс ID:</b> TS0003	<b>Пріоритет:</b> 1	<b>Створено:</b> 15.05.2024, Скрипнюк О.Ю.
<b>Назва:</b> Пошук номера та обличчя		
<b>Вхідні дані:</b> Програма запущена виконані дії з тест-кейса TS0002		
<b>Кроки</b>	<b>Очікуваний результат</b>	
<ol style="list-style-type: none"> <li>Натиснути кнопку «Запуск розпізнавання».</li> <li>Побачити що програма вивела номер, фото машини, фото водія та</li> </ol>	<ol style="list-style-type: none"> <li>Кнопка «Запуск розпізнавання» спрацювала.</li> <li>Справа від кнопки з'явився номер машини, після фото яке було відкрите з'явиться обрізане фото машини далі обличчя водія та фото номера.</li> <li>А) Галочка «Ручне зберігання в базу» не</li> </ol>	

<p>фото номера.</p> <p>3. Залежно від стану галочки «Ручне зберігання в базу» буде виконано дві різні дії.</p>	<p>стоїть, програма сама за пише в базу дані.</p> <p>Б) Галочка «Ручне зберігання в базу» стоїть, програма покаже повідомлення «Бажаєте створити запис в базу даних».</p> <p>4. Якщо галочка «Ручне зберігання в базу» не стоїть або в повідомленні «Бажаєте створити запис в базу даних» було натиснуто «ТАК».</p> <p>А) Якщо номер існує в базі машин, то програма створить новий запис в базу даних.</p> <p>Б) Якщо номера нема в базі, то з'явиться повідомлення «Транспортний засіб з вказаним номером не знайдено. Бажаєте його створити?».</p>
<p><b>Результат виконання тест-кейсу:</b> пройдено успішно</p>	

На формі програми на пункті визначення потрібно натиснути кнопку «Запуск розпізнавання» (рисунок 3.9), після чого мають вивестися номер, фото машини, фото водія та фото номера (рисунок 3.10), якщо галочка «Ручне зберігання в базу» не стоїть, програма сама за пише в базу дані, але якщо галочка «Ручне зберігання в базу» стоїть, програма покаже повідомлення «Бажаєте створити запис в базу даних» (рисунок 3.11), якщо номер існує в базі машин, то програма створить новий запис в базу даних. Якщо номера нема в базі, то з'явиться повідомлення «Транспортний засіб з вказаним номером не знайдено. Бажаєте його створити?» (рисунок 3.12).

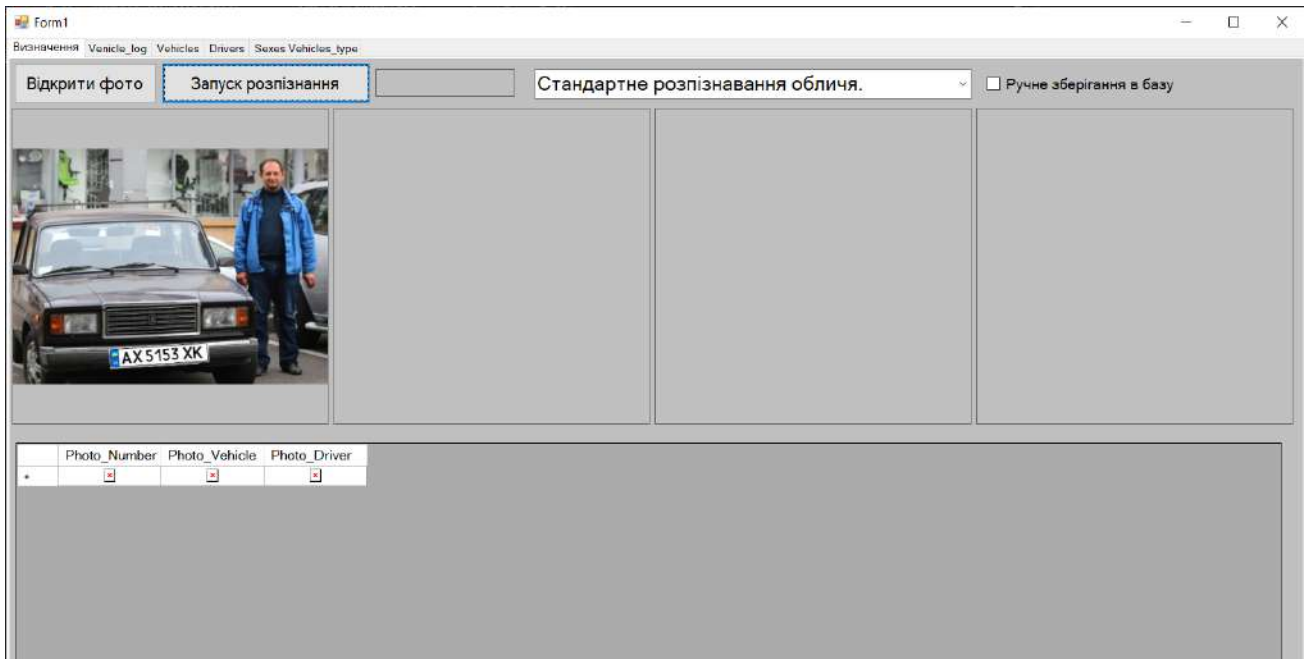


Рисунок 3.9 – Кнопка «Запуск розпізнавання»

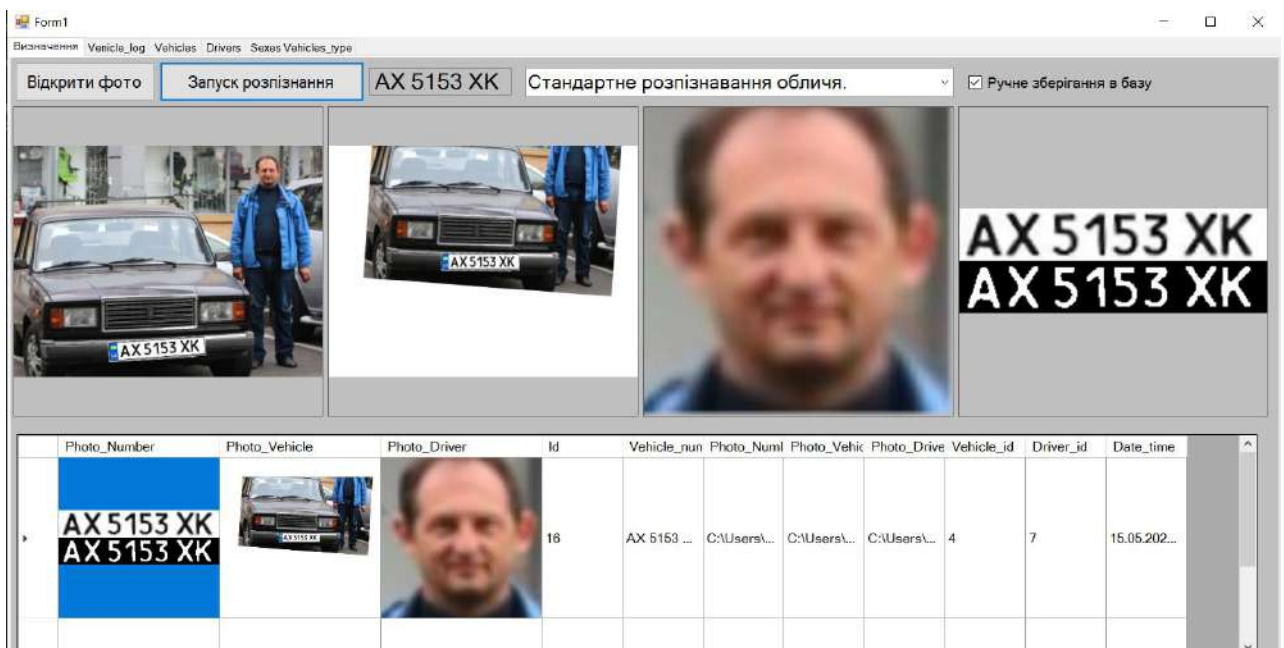


Рисунок 3.10 – Правильне відображення номера, фото машини, фото водія та фото номера.

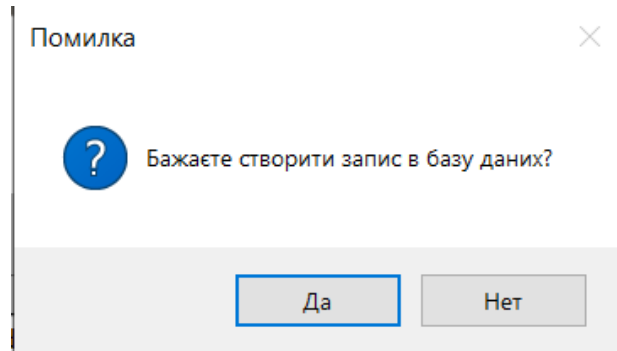


Рисунок 3.11 – Повідомлення про «Ручне зберігання в базу»

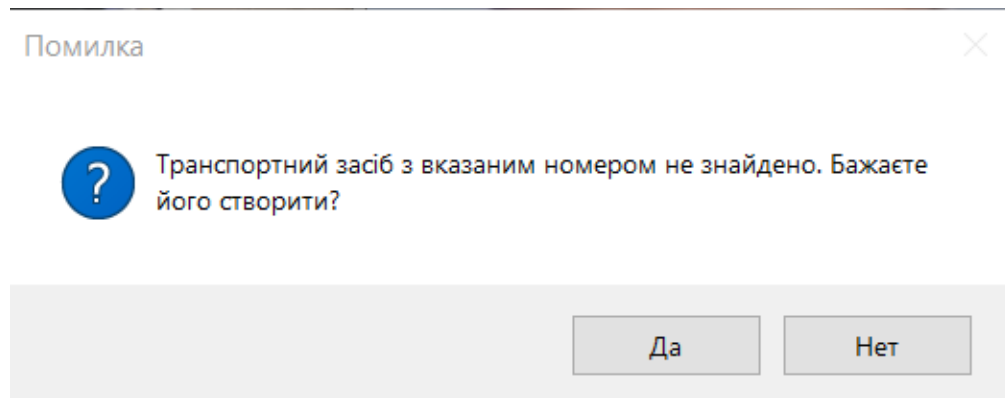


Рисунок 3.12 – Повідомлення що «Транспортний засіб з вказаним номером не знайдено. Бажаєте його створити?»

В таблиці 3.4 описані технічні та програмні вимоги до встановлення та запуску програми

Таблиця 3.4 – Вимоги до системи

Підтримувані операційні системи	<p>Windows 11[35] версії 21H2 та вище: Домашня, Pro, та Корпоративна.</p> <p>Windows 10[36] версії 1903 та вище: Домашня, Pro, для освітніх закладів та корпоративні випуски LTSC.</p> <p>Windows Server 2022[37]: Standard та Datacenter.</p> <p>Windows Server 2019[38]: Standard та Datacenter.</p> <p>Windows Server 2016[39]: Standard та Datacenter.</p>
Обладнання	Двох ядерний процесор із тактовою частотою не нижче 2 ГГц. Рекомендується використовувати як мінімум

	<p>процесор новіше чим Pentium G 2020.</p> <p>4 ГБ ОЗУ; рекомендується 8 ГБ ОЗУ.</p> <p>Місце на жорсткому диску: мінімум 10 Гб вільного місця.</p> <p>Відеоадаптер з можливістю видавати картинку з мінімальною роздільною здатністю 1280 на 720 пікселів, рекомендується 1920 на 1080 пікселів.</p> <p>Монітор з мінімальною роздільною здатністю 1280 на 720 пікселів, рекомендується 1920 на 1080 пікселів.</p> <p>Клавіатура та мишка.</p>
Підтримувані мови	Українська
Вимоги до програмних засобів	Для запуску програми потрібна платформа .NET Framework[40] 4.8 або вище.

### 3.6 Експериментальні дослідження ефективності розроблених методів

Для визначення та дослідження ефективності розроблених методів проведемо експериментальне дослідження та аналіз ефективності запропонованого підходу на основі результатів застосування програмного продукту. Нижче представлено аналіз кожного етапу дослідження, а також відповідні статистичні дані у вигляді діаграм.

У першому дослідженні було оцінено ефективність розпізнавання номерних знаків автомобілів. Для цього було використано набір зображень, на яких програма мала знайти та розпізнати номери. Результати показали, що з можливих трьох балів програма набрала в середньому 2.43 бали, що складає 81% від максимально можливого результату. На рисунку 3.13 наведена відповідна діаграма, яка ілюструє ці результати.

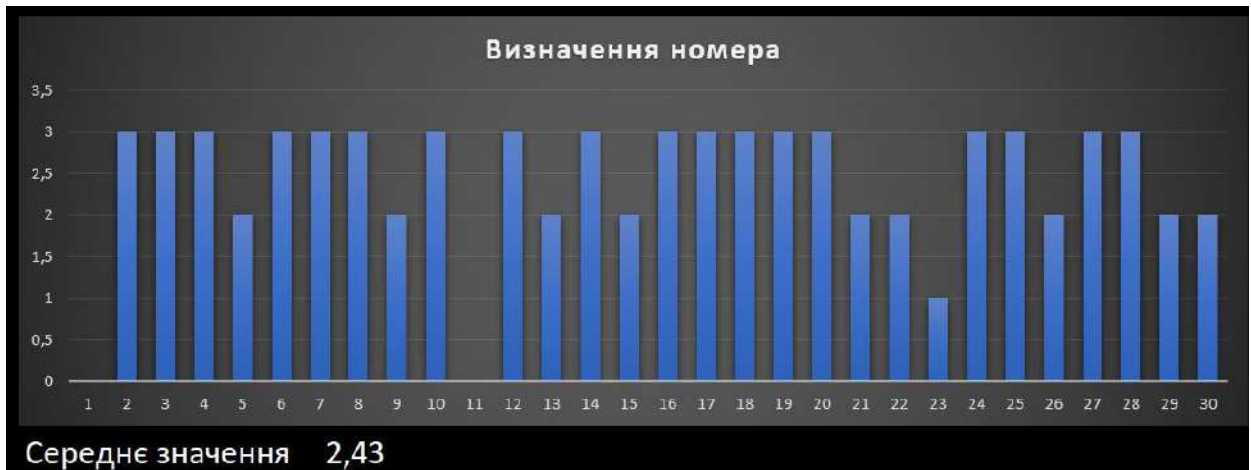


Рисунок 3.13 – Результати дослідження визначення номера

У другому дослідженні було оцінено ефективність розпізнавання обличчя водія. Результати цього дослідження також були представлені у вигляді діаграми. Середній бал склав 2.35, що дорівнює 78% від максимально можливого результату. На рисунку 3.14 наведена діаграма, яка ілюструє ці результати.

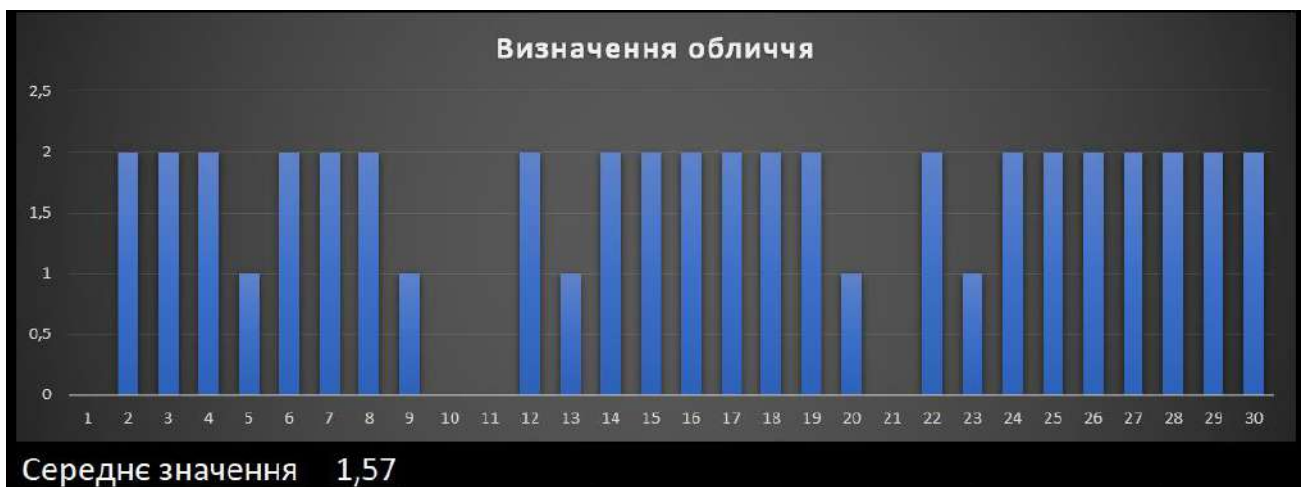


Рисунок 3.14 – Результати дослідження визначення обличчя

У третьому дослідженні оцінювалася ефективність одночасного розпізнавання номерних знаків та обличчя водія. Результати показали, що середній бал склав 2.43, що дорівнює 81% від максимально можливого результату. На рисунку 3.15 наведена діаграма, яка ілюструє ці результати.

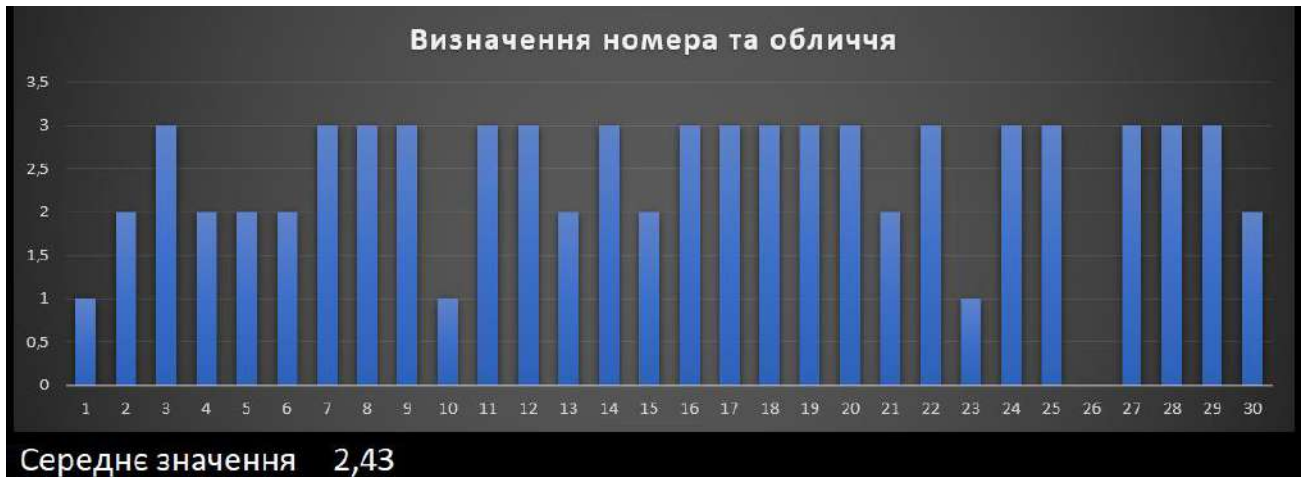


Рисунок 3.15 – Результати дослідження визначення номера та обличчя

Аналіз результатів вимірювань, представлених у вигляді графіків та середніх значень, дозволяє зробити висновки про ефективність методів для розробленого програмного продукту. За результатами дослідження стало видно, що методи які були створені для програми здатні знаходити номери, але іноді виникають проблеми з визначенням тексту номера.

Також стало зрозуміло, що метод розпізнавання обличчя водія має певні проблеми у визначенні обличчя, хоча результати були задовільними. При одночасному визначенні номера та обличчя результати програми виявилися кращими, ніж очікувалося, хоча деякі проблеми все ж з'явилися.

Результати експериментальних досліджень підтверджують, що розроблений метод демонструє високу ефективність, але потребує подальшого вдосконалення для підвищення точності розпізнавання тексту номерів та облич.

### 3.7 Висновки до розділу 3

На основі спроектованих методів було розроблено програмну реалізацію метод автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями яка дозволяє практично реалізувати завдання які було поставлено на основі структури та функціонального призначення елементів.

Програмна реалізація реалізованої програми складається з кількох важливих модулів, кожен з яких відповідає за конкретні функції. Особливості

реалізації програмних складових системи включають розпізнавання тексту та обличчя, вивід даних, покращення фото, роботу з базою даних та інші

Програмну реалізацію методу було протестовано за допомогою тест-кейсів, було успішно перевірено функціональність програми. Кожен з тест-кейсів, які були розглянуті, пройшов успішно, підтверджуючи правильну роботу програми відповідно до очікуваних результатів.

Було проведено дослідження що показало, що програмний продукт «NumberFaceIdentifier» виявився ефективним у визначенні номера автомобіля та тексту на ньому під час дослідження він отримав 81% правильних результатів, однак він має певні проблеми у визначенні обличчя водія під час дослідження він отримав 78.5%, що впливає на точність результатів дослідження у визначенні номера автомобіля та одночасного розпізнання обличчя водія під час такого дослідження він отримав 81% що виявилось більшим результатом чим очікувалося.

## Загальні висновки

У рамках виконання кваліфікаційна робота бакалавра було досягнуто поставленої мети було розроблено метод що дозволяє спростити формування бази даних водіїв та номерів автомобілів за допомогою застосування інтелектуальних систем. Також було створено програмну реалізацію методу для автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями що відповідає поставленій меті кваліфікаційної роботи бакалавра, а саме спрощення формування бази даних водіїв та номерів автомобілів за допомогою застосування інтелектуальних систем.

Для реалізації поставленої мети було використано сучасні методи та засоби, включаючи нейронні мережі, оптичне розпізнавання символів та методи обробки зображень, основні результати виконання кваліфікаційної роботи:

Для розробки методу було проведено глибокий аналіз предметної області, проведено аналіз методів розпізнавання тексту на номерних знаках та обличчя водія. Оцінено наявні рішення та технології, виявлено їхні сильні та слабкі сторони. Також була проведена автоматизація обробки даних автоматизовано основні етапи обробки зображень, включаючи збір, первинну обробку, виявлення контурів та підготовку до розпізнавання. Що дозволило значно підвищити ефективність системи.

Під час створення програмної реалізації методу автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями було розроблено метод розпізнавання тексту на зображеннях номерних знаків, який враховує різні умови освітлення та кути зйомки для забезпечення високої точності та метод детектування та ідентифікації обличчя водіїв на зображеннях, що дозволяє точно визначати особистість водія в різних умовах зйомки.

Було створено програмну реалізацію розроблених методів детектування обличчя водіїв та розпізнавання тексту на номерних знаках. Реалізація включає використання спеціалізованих бібліотек та інструментів для обробки зображень та OCR. Проведено тестування та оптимізація програмної реалізації методу,

проведено навчання та тестування нейронної мережі на великому наборі даних, що дозволило досягти високої точності розпізнавання символів на номерних знаках та облич водіїв.

Були проведені дослідження ефективності. Проведені експериментальні дослідження показали високу точність розпізнавання номерних знаків та облич. Точність розпізнавання номерних знаків склала понад 81%, а облич водіїв – близько 78.5%. Система продемонструвала стабільну роботу у різних умовах, включаючи різні рівні освітлення та різні кути зйомки.

Розроблена програма показала має високий потенціал для впровадження в реальні умови, зокрема для використання в системах контролю дорожнього руху, паркінгах, митницях та інших подібних сферах. Система може бути використана для автоматизації процесів контролю та ідентифікації автомобілів у різних установах та організаціях.

Можливості та шляхи вдосконалення:

- інтеграція з інформаційними системами, розширення функціонала для підтримки нових типів зображень та поліпшення алгоритмів розпізнавання для підвищення точності;
- використання більш сучасних моделей нейронних мереж та алгоритмів машинного навчання для обробки зображень та розпізнавання символів;
- додавання механізму самонавчання системи на основі нових даних, що надходять у процесі її експлуатації.

Таким чином, виконання даної кваліфікаційної роботи дозволило досягти поставленої мети та створити ефективну реалізацію методу автоматичного створення бази даних водіїв та номерів автомобілів, яка має високий потенціал для практичного використання та подальшого вдосконалення.

## Перелік посилань

1. Wikipedia. Deep Learning. URL: [https://en.wikipedia.org/wiki/Deep\\_learning](https://en.wikipedia.org/wiki/Deep_learning)
2. Аналіз методів комп'ютерного зору в задачах ідентифікації осіб у відеопотоці. URL: <http://www.tsatu.edu.ua/kn/wp-content/uploads/sites/16/analiz-metodiv-kompjuternoho-zoru-v-zadachah-identyfikaciyi-osib-u-videopotoci.pdf>
3. GitHub. OpenALPR. URL: <https://github.com/openalpr/openalpr>
4. GitHub. Tesseract OCR. URL: <https://github.com/tesseract-ocr/tesseract>
5. Wikipedia. Convolutional Neural Network. URL: [https://en.wikipedia.org/wiki/Convolutional\\_neural\\_network](https://en.wikipedia.org/wiki/Convolutional_neural_network)
6. EAI. A Secure Image Steganography System using Chaos Based Steganographic Algorithm. URL: <https://publications.eai.eu/index.php/sis/article/view/2055>
7. ACM. A Comprehensive Survey of Machine Learning for Networking: Evolution, Applications and Research Opportunities. URL: <https://dl.acm.org/doi/abs/10.1145/3439726>
8. MDPI. Sensors. URL: <https://www.mdpi.com/1424-8220/19/12/2809>
9. Springer. Image Processing for Computer Graphics. URL: [https://link.springer.com/chapter/10.1007/978-1-4684-2106-4\\_22](https://link.springer.com/chapter/10.1007/978-1-4684-2106-4_22)
10. ResearchGate. Database Meets Artificial Intelligence: A Survey. URL: [https://www.researchgate.net/publication/341427551\\_Database\\_Meets\\_Artificial\\_Intelligence\\_A\\_Survey](https://www.researchgate.net/publication/341427551_Database_Meets_Artificial_Intelligence_A_Survey)
11. OpenALPR. URL: <https://www.openalpr.com/>
12. Officer. OpenALPR Version 2.6. URL: <https://www.officer.com/command-hq/technology/traffic/lpr-license-plate-recognition/product/21031080/openalpr-technology-inc-openalpr-version-26>
13. Rekor. URL: <https://www.rekor.ai/>

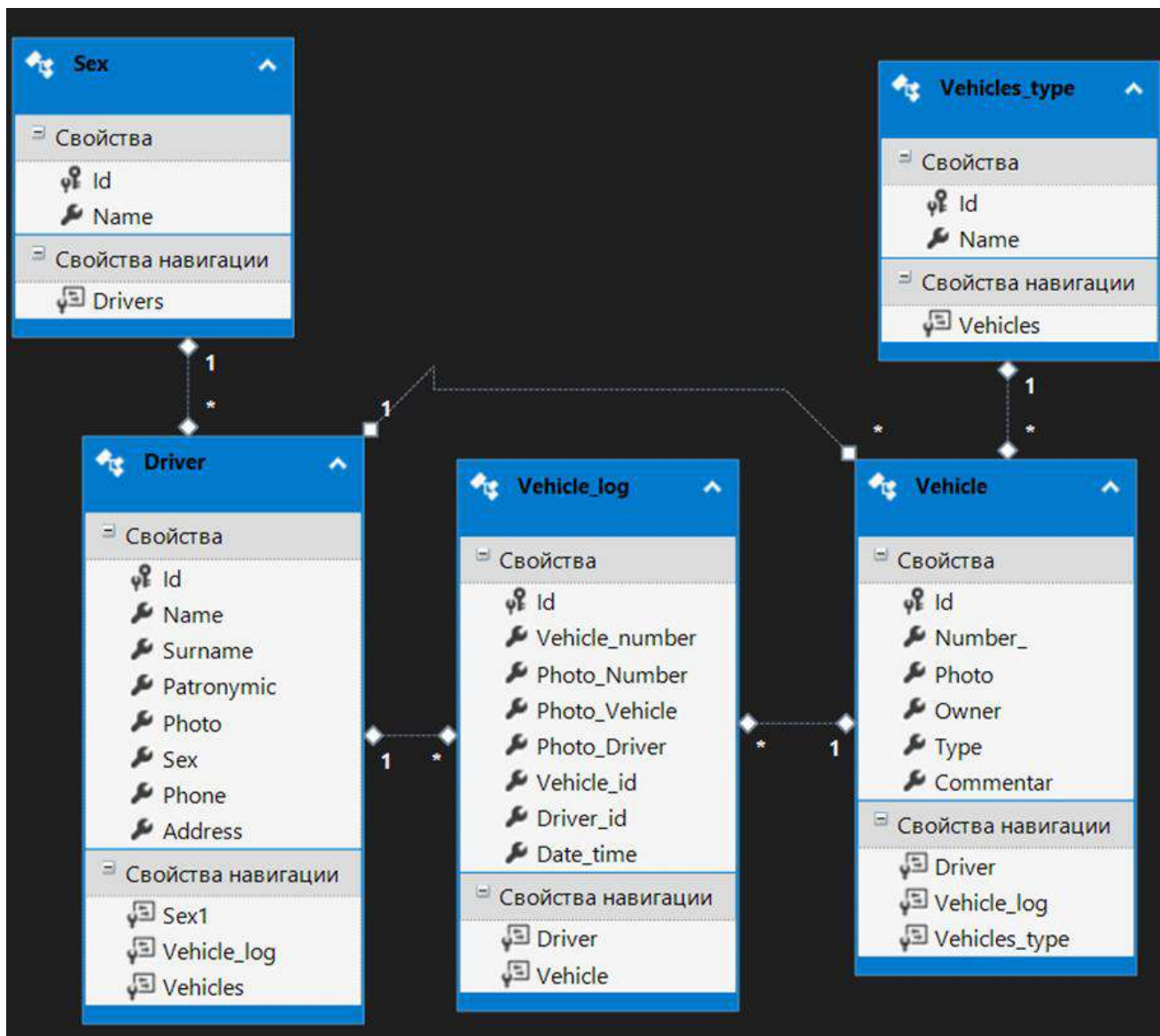
14. Rekor. URL: [https://cdn.prod.website-files.com/61329f1ed969e70039c05b5e/643ee7c4d4dbbe58baf207e7\\_alpr-demo.jpg](https://cdn.prod.website-files.com/61329f1ed969e70039c05b5e/643ee7c4d4dbbe58baf207e7_alpr-demo.jpg)
15. Amazon Rekognition. URL: <https://aws.amazon.com/rekognition/>
16. Cloud Academy. Amazon Rekognition Video Feature Announcement. URL: <https://cloudacademy.com/blog/amazon-rekognition-video-feature-announcement/>
17. OpenCV. URL: <https://opencv.org/>
18. PyImageSearch. OpenCV Face Recognition. URL: <https://pyimagesearch.com/2018/09/24/opencv-face-recognition/>
19. Viso. OpenCV. URL: <https://viso.ai/computer-vision/opencv/>
20. Kaggle. Stanford Cars Dataset. URL: <https://www.kaggle.com/datasets/jessicali9530/stanford-cars-dataset>
21. Labeled Faces in the Wild. URL: <https://vis-www.cs.umass.edu/lfw/>
22. Emgu CV. URL: [https://www.emgu.com/wiki/index.php/Main\\_Page](https://www.emgu.com/wiki/index.php/Main_Page)
23. Wikipedia. Tesseract. URL: <https://uk.wikipedia.org/wiki/Tesseract>
24. Wikipedia. Rating Scale. URL: [https://en.wikipedia.org/wiki/Rating\\_scale](https://en.wikipedia.org/wiki/Rating_scale)
25. Wikipedia. C# (Programming Language). URL: [https://en.wikipedia.org/wiki/C\\_Sharp\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/C_Sharp_(programming_language))
26. Wikipedia. C++. URL: <https://en.wikipedia.org/wiki/C++>
27. Microsoft. Visual Studio. URL: <https://visualstudio.microsoft.com/>
28. Microsoft. WinForms Overview. URL: <https://learn.microsoft.com/uk-ua/dotnet/desktop/winforms/overview/?view=netdesktop-8.0>
29. Wikipedia. .NET Framework. URL: [https://uk.wikipedia.org/wiki/.NET\\_Framework](https://uk.wikipedia.org/wiki/.NET_Framework)
30. Wikipedia. SQL. URL: <https://en.wikipedia.org/wiki/SQL>
31. Microsoft. Access. URL: <https://www.microsoft.com/uk-ua/microsoft-365/access>
32. Wikipedia. Microsoft SQL Server. URL: [https://en.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](https://en.wikipedia.org/wiki/Microsoft_SQL_Server)

33. Wikipedia. Діаграма класів. URL: [https://uk.wikipedia.org/wiki/Діаграма\\_класів](https://uk.wikipedia.org/wiki/Діаграма_класів)
34. Wikipedia. Test Case. URL: [https://en.wikipedia.org/wiki/Test\\_case](https://en.wikipedia.org/wiki/Test_case)
35. Microsoft. Windows 11. URL: <https://www.microsoft.com/uk-ua/windows/windows-11>
36. Wikipedia. Windows 10. URL: [https://uk.wikipedia.org/wiki/Windows\\_10](https://uk.wikipedia.org/wiki/Windows_10)
37. Microsoft. Windows Server. URL: <https://www.microsoft.com/uk-ua/windows-server>
38. Microsoft. Evaluate Windows Server 2019. URL: <https://www.microsoft.com/en-us/evalcenter/evaluate-windows-server-2019>
39. Microsoft. Evaluate Windows Server 2016. URL: <https://www.microsoft.com/en-us/evalcenter/evaluate-windows-server-2016>
40. Wikipedia. .NET Framework. URL: [https://uk.wikipedia.org/wiki/.NET\\_Framework](https://uk.wikipedia.org/wiki/.NET_Framework)

# ДОДАТКИ

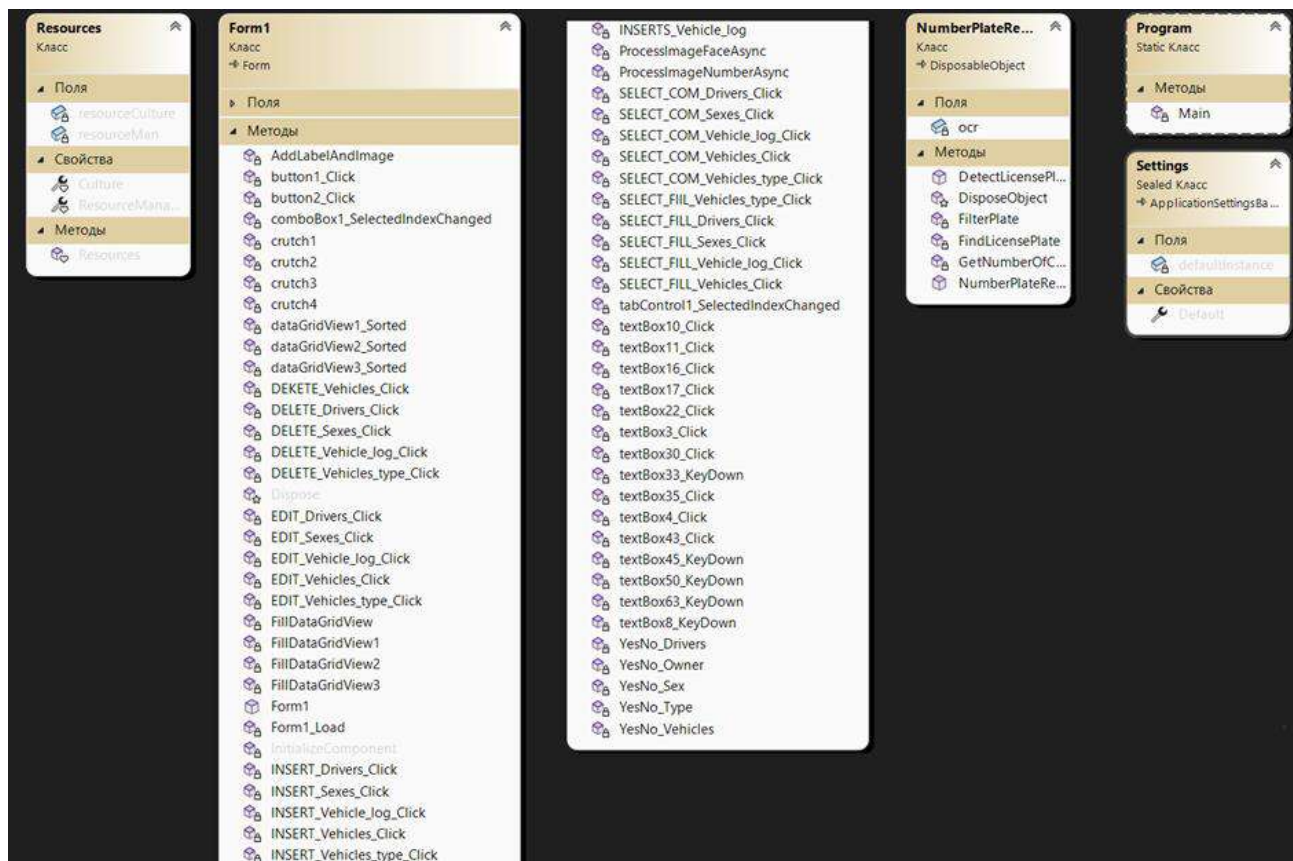
## Додаток А

## Структура бази даних методу з автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями



## Додаток Б

### Розгорнута структура класів методу з автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями



# Актуальність

Розвиток автомобільної індустрії та широке впровадження штучного інтелекту у сферу транспорту прогнозується, як фактор, що може вплинути на можливість заміни людей автоматизованими системами на дорогах та пропускних пунктах. Зокрема, зростання швидкості, з якою автомобілі рухаються, ставить перед собою завдання забезпечення безпеки на дорогах, що може бути складним завданням для людських операторів. Отже, розробка продукту для автоматичного розпізнавання номерів та водіїв стає актуальною.

Актуальність цієї проблематики відображається у багатьох аспектах дорожньої безпеки та ефективного управління транспортним потоком.

По-перше, забезпечення безпеки на дорогах є однією з головних завдань держави та органів управління. Автоматизовані системи розпізнавання номерів та водіїв можуть сприяти у вчасному виявленні порушень правил дорожнього руху та запобігати аварійним ситуаціям.

По-друге, ефективний контроль дорожнього руху є важливим для забезпечення плавного руху транспорту та запобігання заторів. Застосування технологій розпізнавання номерів та водіїв може допомогти в ідентифікації транспортних засобів, їх швидкості та маршрутів, що сприятиме більш ефективному управлінню дорожнім рухом.

По-третє, оптимізація використання дорожнього простору є ключовим завданням для міст та мегаполісів, де дефіцит місця на дорогах є проблемою. Впровадження систем автоматичного розпізнавання може сприяти більш ефективному розподілу транспортних засобів та зменшенню заторів, що відобразиться на загальній продуктивності та комфортності дорожнього руху.

КВАЛІФІКАЦІЙНА РОБОТА

БАКАЛАВРА

## МЕТОД АВТОМАТИЧНОГО СТВОРЕННЯ БАЗИ ДАНИХ ВОДІІВ ТА НОМЕРІВ АВТОМОБІЛІВ ЗА ЗОБРАЖЕННЯМИ

**Виконав:**

студент 4 курсу, групи КН-20-2

**Скрипнюк Олександр Юрійович**

**Керівник:**

професор кафедри КН

**Манзюк Едуард Андрійович**



# Мета і задачі роботи

**Метою кваліфікаційної роботи бакалавра** є спрощення формування бази даних водіїв та номерів автомобілів за допомогою застосування інтелектуальних систем, основні задачі для досягнення цієї мети включають:

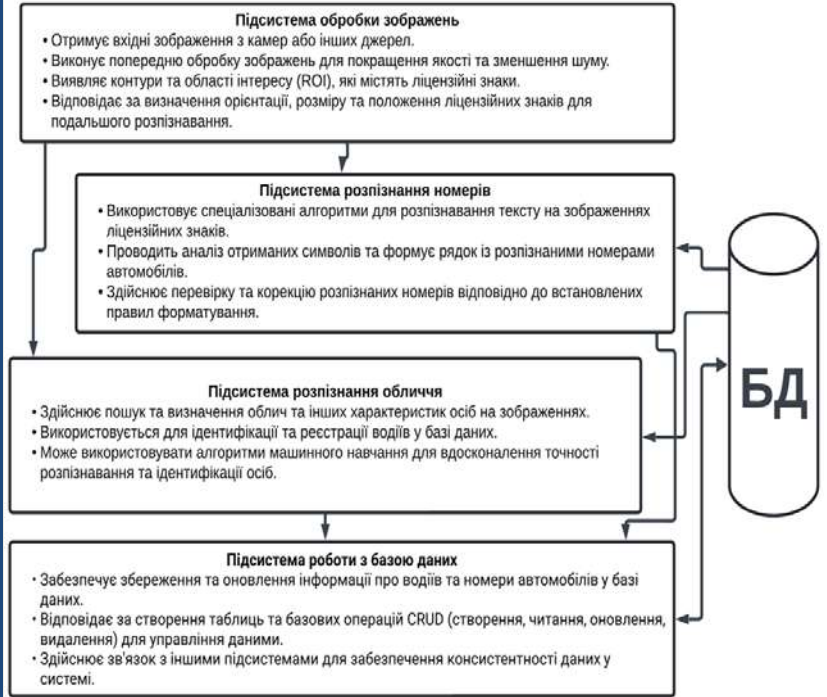
- проведення аналізу предметної області щодо методів розпізнавання тексту на номерних знаках, обличчя водія та виконати аналіз наявних рішень щодо подібних задач;
- розробка методу визначення текстової інформації на номерних знаках автомобілів;
- розробка методу детектування та визначення обличь водіїв автомобілів;
- виконання програмної реалізації розроблених методів детектування обличь водіїв та текстової інформації на номерних знаках автомобілів;
- проведення експериментальних досліджень розроблених методів та розробленої програмної реалізації.

# Об'єкт та предмет дослідження

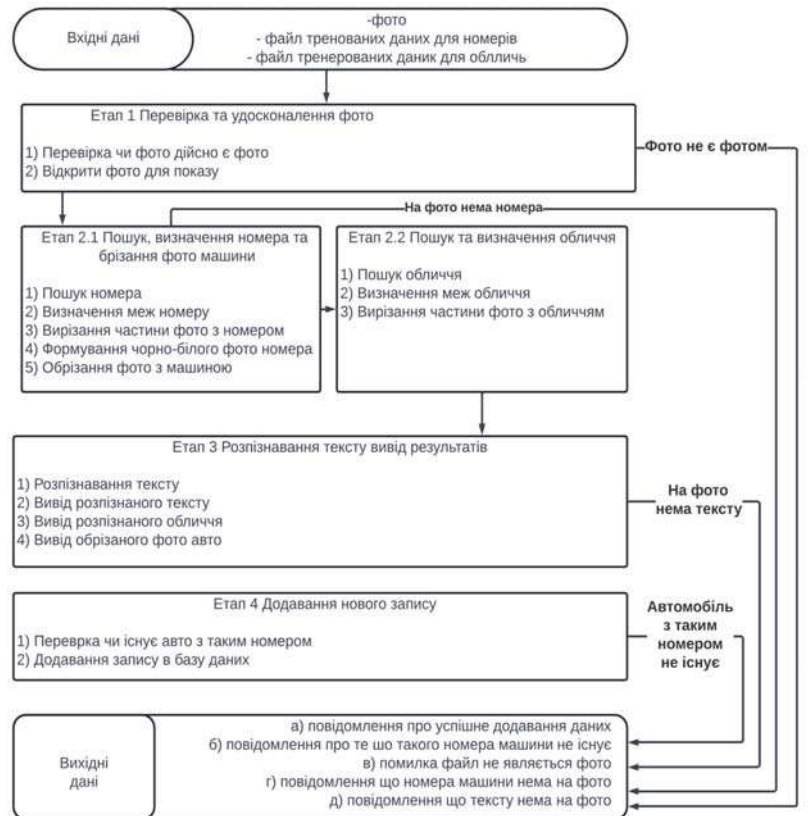
**Об'єкт дослідження** – процес автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями.

**Предмет дослідження** – методи та технології машинного навчання для розпізнавання та порівняння.

# Схема проєктної архітектури методу автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями



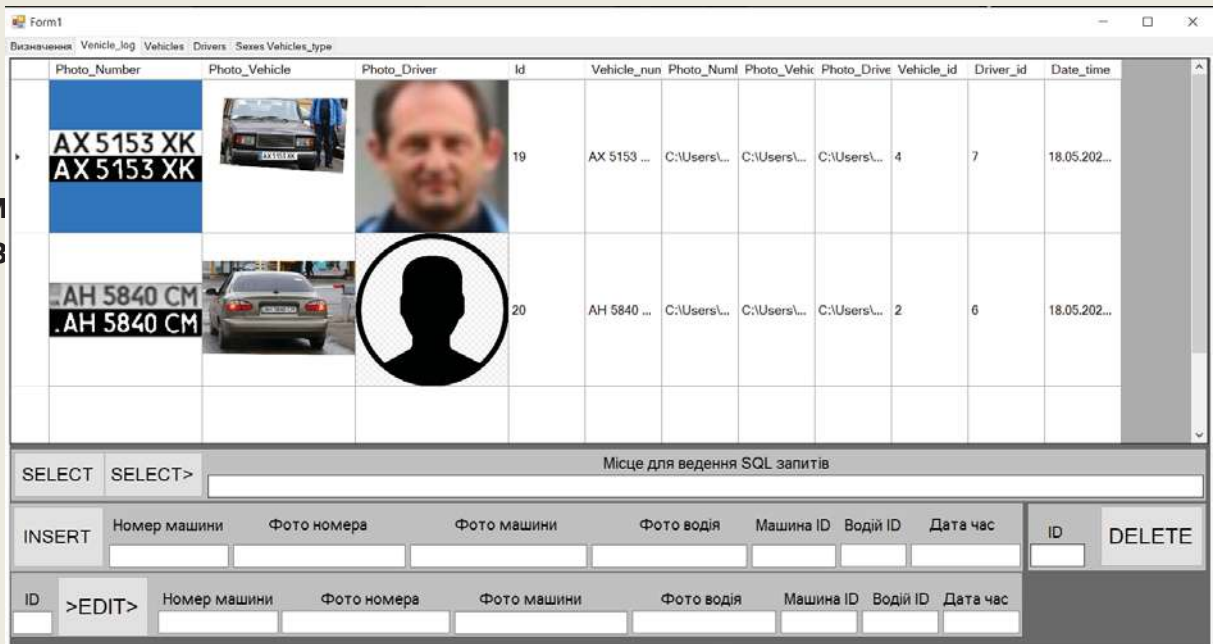
# Схема методу автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями.





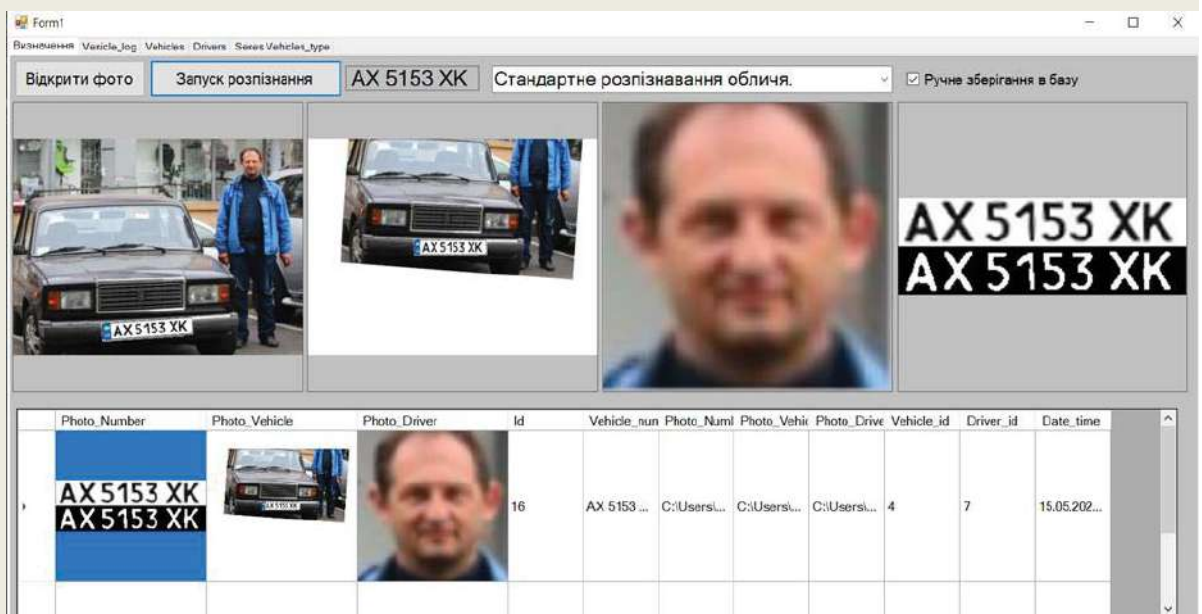
# Програмна реалізація методу автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями

Підсистема роботи з базою даних



# Програмна реалізація методу автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями

Підсистема обробки зображень, розпізнання номерів та підсистема розпізнання обличчя.



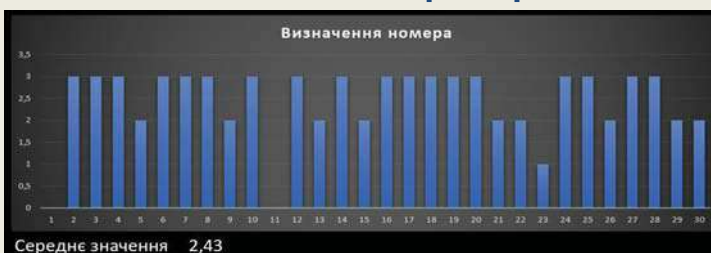
# Висновки

Метою кваліфікаційна робота бакалавра було розробити метод що дозволяє спростити формування бази даних водіїв та номерів автомобілів за допомогою застосування інтелектуальних систем.

Для досягнення поставленої мети в рамках роботи були поставлені та виконані наступні завдання:

- проведено аналіз предметної області щодо методів розпізнавання тексту на номерних знаках, обличчя водія та виконати аналіз наявних рішень щодо подібних задач. Оцінено наявні рішення та технології, виявлено сильні та слабкі сторони;
- розроблено метод визначення текстової інформації на номерних знаках автомобілів який враховує різні умови освітлення та кути зйомки для забезпечення високої точності;
- розроблено метод детектування та визначення обличчя водіїв автомобілів що дозволяє точно визначати обличчя водія в різних умовах зйомки;
- виконано програмну реалізацію розроблених методів детектування обличчя водіїв та текстової інформації на номерних знаках автомобілів. Реалізація включає використання спеціалізованих бібліотек та інструментів для обробки зображень та OCR;

## Експериментальні дослідження ефективності розробленого метода

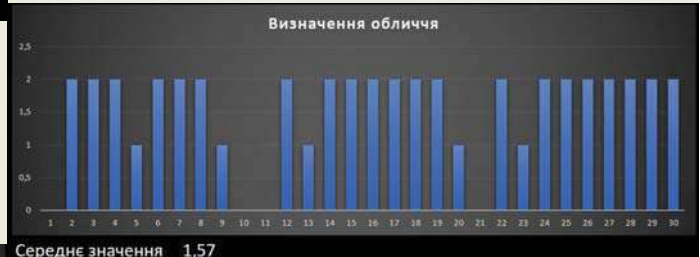


Ефективність розпізнавання номерних знаків автомобілів.

**Результати:** програмна реалізація методу набрала в середньому 2.43 бали, що складає 81%

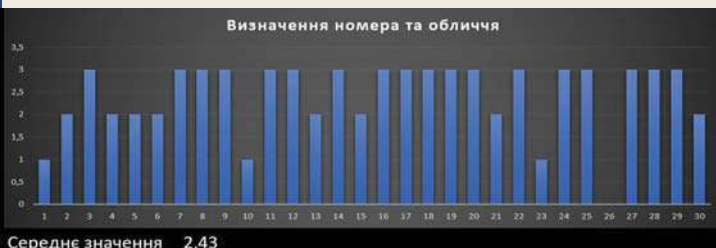
Ефективність розпізнавання обличчя водіїв.

**Результати:** програмна реалізація методу набрала в середньому 1.57 бали, що складає 78.5%



Ефективність одночасного розпізнавання номерних знаків автомобілів та обличчя водіїв.

**Результати:** програмна реалізація методу набрала в середньому 2.43 бали, що складає 81%



# ДЯКУЮ ЗА УВАГУ!

## Висновки

- проведено експериментальне дослідження розроблених методів та розробленої програмної реалізації. Проведені експериментальні дослідження показали високу точність розпізнавання номерних знаків та облич, точність розпізнавання номерних знаків склала понад 81%, облич водіїв – близько 78.5%, а точність одночасного розпізнавання номерних знаків та обличчя водія склала понад 81%.

Також система продемонструвала стабільну роботу у різних умовах, включаючи різні рівні освітлення та різні кути зйомки.

Розроблена програма показала високий потенціал для впровадження в реальні умови, зокрема для використання в системах контролю дорожнього руху, паркінгах, митницях та інших подібних сферах. Система може бути використана для автоматизації процесів контролю та ідентифікації автомобілів у різних установах та організаціях.

Програмна реалізація методу, під час виконання даної кваліфікаційної роботи, дозволило досягти поставленої мети, а саме – спрощення формування бази даних водіїв та номерів автомобілів за допомогою застосування інтелектуальних систем та створити ефективну реалізацію методу автоматичного створення бази даних водіїв та номерів автомобілів, яка має високий потенціал для практичного використання та подальшому вдосконаленні.

## Anti-Plagiarism v-15.257

**Максимальне співпадіння з одним документом 4.0%**

Словники перевірки: en\_US, ru\_RU, ua\_UA. **Помилки в документах: 8%**

ID: 129191 Назва: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА на тему Метод автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями Додано в БД: 2024-06-09 Автора: Олександр СКРИПНІЮК Керівники: Едуард МАНЗІЮК Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	72577	1104	4895 (7%)	81 (7%)

### Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

Ім'я користувача:  
Кафедра КН

ID перевірки:  
1016338626

Дата перевірки:  
09.06.2024 17:46:14 EEST

Тип перевірки:  
Doc vs Internet + Library

Дата звіту:  
09.06.2024 17:51:08 EEST

ID користувача:  
100005671

Назва документа: КН-20-2 Скрипнюк\_ЗАПИСКА

Кількість сторінок: 71 Кількість слів: 11719 Кількість символів: 91940 Розмір файлу: 3.03 MB ID файлу: 1016139735

## 6.84% Схожість

Найбільша схожість: 2.57% з джерелом з Бібліотеки (ID файлу: 1016133152)

5.28% Джерела з Інтернету

501

Сторінка 73

5.19% Джерела з Бібліотеки

145

Сторінка 77

## 0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

## 0% Вилучень

Немає вилучених джерел

**РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНИХ НАУК  
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Метод автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями

Автор: студент групи КН-20-2 Олександр СКРИПНІЮК

Спеціальність: 122 – Комп'ютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: д.т.н., професор кафедри Манзюк Е.А.

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	<b>відповідає</b>
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

**Підтвердження:**

Запозичення, виявлені в роботі Олександра СКРИПНІЮКА, не є плагіатом, оскільки: запозичення розміщені в розділі огляду існуючих підходів, не описують безпосередньо авторську роботу і не стосуються її результатів; усі запозичення фрагментарні; до запозичень входять фрагменти програмного коду, що не мають авторства і містять поширені конструкції; серед запозичень знаходяться загальновідомі терміни, скорочення.

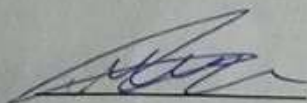
Обсяг запозичень, визначений системами виявлення збігів/ідентичності/схожості, складає:

- за системою Anti-Plagiarism: 4%;

- за системою Unichesk: 6,84%.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості є допустимим.

Керівник роботи



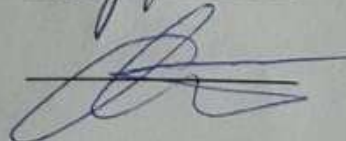
Едуард МАНЗЮК

Гарант ОП



Олександр МАЗУРЕЦЬ

Завідувач кафедри КН



Олександр БАРМАК



**ВІДГУК НАУКОВОГО КЕРІВНИКА  
на кваліфікаційну роботу бакалавра**

студента гр. КН-20-2 Олександра СКРИПНЮКА

за темою Метод автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями

**1. Актуальність теми**

Розвиток автомобільної індустрії та впровадження штучного інтелекту у транспортну сферу може суттєво змінити баланс між людськими операторами та автоматизованими системами на дорогах та пропускних пунктах. Зокрема, збільшення швидкості руху транспортних засобів ставить перед суспільством завдання гарантування безпеки на дорогах, що стає дедалі складнішим завданням для людини. Таким чином, розробка систем для автоматичного розпізнавання номерних знаків та водіїв набуває особливої важливості. Значущість цієї проблеми виявляється у різних аспектах забезпечення безпеки дорожнього руху та ефективного керування транспортними потоками. Автоматизовані системи розпізнавання номерів та водіїв можуть відігравати важливу роль у своєчасному виявленні порушень правил дорожнього руху, а також у запобіганні аварійним ситуаціям.

**2. Відповідність роботи предметній області Стандарту спеціальності 122 Комп'ютерні науки**

Згідно зі стандартом, досліджувані об'єкти та сфера діяльності включають математичні, інформаційні та імітаційні моделі реальних явищ, об'єктів, систем і процесів, а також методи і технології для отримання, зберігання, обробки, передачі та використання інформації. Основна мета роботи полягає в розробці методу автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями. Для розв'язання цієї задачі застосовано математичні моделі, методи та алгоритми, що дозволяють вирішувати теоретичні та прикладні проблеми, які виникають при створенні методів машинного навчання. Результати виконання кваліфікаційної роботи бакалавра відповідають стандартам бакалавра за спеціальністю 122 – Комп'ютерні науки.

**3. Професійні та особистісні якості бакалавра**

Під час роботи над кваліфікаційною роботою бакалавра Олександр Скрипнюк виявив високий рівень знань та навичок, вчасно виконуючи всі поставлені завдання. У процесі написання пояснювальної записки та розробки методу він продемонстрував свої професійні компетенції та навчальні досягнення. Олександр засвоїв професійні навички в галузі "Комп'ютерні науки".

**4. Ступінь самостійності під час виконання кваліфікаційної роботи**

Одержані в роботі результати є наслідком особистої діяльності студента, який самостійно виконував усі поставлені задачі.

**5. Ступінь оволодіння методами дослідження**

При реалізації кваліфікаційної роботи показав належний рівень компетентностей та володіння необхідними методами, методиками та технологіями предметної області комп'ютерних наук.

**6. Повнота та якість розкриття теми роботи**

Тема роботи повною мірою обґрунтована та розкрита належним чином. Проведено аналіз відомих досліджень відповідно до обраної теми. Поставлені завдання, реалізовані та розроблено програмне забезпечення для реалізації запропонованого метода.

**7. Логічність, послідовність, аргументованість, літературна грамотність викладення матеріалу**

Структура роботи та послідовність викладення логічні та відповідають поставленій меті. Викладення матеріалу послідовне, аргументоване, літературно грамотне.

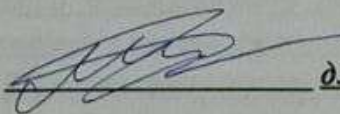
**8. Можливість практичного застосування кваліфікаційної роботи бакалавра, окремих її частин**

Розроблений у роботі метод може бути використаний в автоматичних системах регулювання та виявлення порушень правил дорожнього руху.

**9. Висновок про можливість допуску кваліфікаційної роботи бакалавра до захисту, на яку оцінку заслуговує робота**

Враховуючи належний рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка «відмінно».

Керівник



д.т.н., професор каф. КН Едуард МАНЗЮК



## РЕЦЕНЗІЯ

### на кваліфікаційну роботу бакалавра

студента *гр. КН-20-2 Скрипнюка Олександра Юрійовича*

за темою: Метод автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями

#### 1. Актуальність обраної теми

Виявлення та розпізнавання номерних знаків і облич водіїв є важливою задачею комп'ютерного зору, яка стала ключовим завданням для багатьох практичних застосувань, таких як пропускні пункти, платні автомобільні дороги, запобігання аварійних ситуацій, а також для контролю руху транспортних засобів правоохоронними органами. Тому розробка систем штучного інтелекту для автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями є актуальним завданням, яке має потенціал значно поліпшити безпеку та ефективність дорожнього руху.

#### 2. Повнота розкриття мети та завдань роботи

Мета та завдання, сформульовані в кваліфікаційній роботі, розкрито належним чином. Проведено ґрунтовний аналіз предметної області, визначено актуальність розробки та досліджено відомі методи. Кроки розробленого методу детально описано, а його реалізацію в програмній системі виконано успішно. Крім того, отримані експериментальні результати підтвердили ефективність практичного застосування розробленого методу, що підкреслює його значущість у вирішенні важливих завдань у відповідній галузі.

#### 3. Зміст кожного розділу роботи

Записка кваліфікаційної роботи бакалавра містить три розділи. У першому розділі проведено аналіз предметної області, досліджено відомі роботи та визначено актуальність теми, що підкреслює важливість подальших досліджень у даній області. У другому розділі детально представлено метод автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями, описано кроки його розробки та принципи реалізації. Третій розділ присвячено практичній реалізації розробленого методу та експериментальній перевірці його ефективності, що дозволяє зробити висновки про його придатність для використання у реальних умовах.

#### 4. Оцінка розробленої інформаційної системи, її практична цінність

Розроблений метод автоматичного створення бази даних водіїв та номерів автомобілів за зображеннями є важливим кроком у вдосконаленні систем спостереження та безпеки на дорогах. Ця технологія дозволяє ефективно визначати машини та водіїв, що має безліч застосувань у сферах спостереження та контролю машин на дорогах та пропускних пунктах. Вона також може бути використана для автоматичного виявлення порушень дорожнього руху та оптимізації транспортних потоків, що сприяє підвищенню ефективності та безпеки управління дорожнім рухом.

5. Якість оформлення кваліфікаційної роботи бакалавра

Оформлення кваліфікаційної роботи бакалавра є на високому рівні. Вона включає всі необхідні розділи, таблиці, графіки та посилання на використану літературу. Чітка структура, логічне та послідовне викладення матеріалу, та правильна організація тексту сприяє сприйняттю та оцінці роботи.

6. Недоліки кваліфікаційної роботи бакалавра

Рекомендовано вдосконалити метод додавши можливість здійснювати опрацювання декількох машин на зображенні одночасно, що дозволить підвищити швидкість та ефективність обробки великого обсягу даних у реальному часі.

7. Загальний висновок (допускається чи не допускається до захисту), та оцінка на яку заслуговує кваліфікаційна робота.

Враховуючи рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка «відмінно».

Рецензент К.Т.Н., ррччй каф КііЕ Ніхеноржс А.О.



12.06.2024