

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Комп'ютерна система розпізнавання дорожніх знаків»

08-54.БКР.014.00.000 ПЗ

Виконав: студент 4 курсу, групи ІСП-216
спеціальності 123 — «Комп'ютерна інженерія»
освітня програма — «Системне програмування»

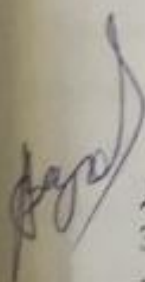
Стадник О.В.

Керівник phd, ст. вик. каф. ОТ

Обертюх М.Р.

Рецензент к.т.н., доц. каф. МБІС

Гришак А. В.


Допущено до захисту
Завідувач кафедри ОТ
д.т.н., проф. Азаров О. Д.

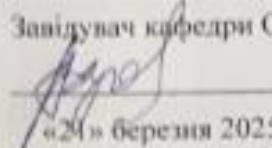
«19» 06 2025 р.

Вінниця ВНТУ – 2025 рік

Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки
Освітньо—кваліфікаційний рівень бакалавр
Галузь знань — 12 Інформаційні технології
Спеціальність — 123 Комп'ютерна інженерія Освітня
програма — Системне програмування

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ проф., д.т.н.

 Азаров О. Д.
«24» березня 2025 року

З А В Д А Н Н Я НА БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

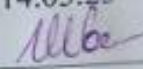
Стаднику Олександру Володимировичу

- 1 Тема роботи: «Комп'ютерна система розпізнавання дорожніх знаків», керівник роботи, PhD, ст. вик. каф. ОТ кафедри ОТ Обертюх Максим Романович, затверджена наказом вищого навчального закладу від «20» березня 2025 року №97.
- 2 Строк подання студентом роботи 13.05.2025
- 3 Вихідні дані до роботи: технології розробки — мова програмування Python, бібліотека OpenCV для роботи із зображеннями, бібліотека numpy для роботи із масивами даних, середовище розробки — Visual Studio.
- 4 Зміст розрахунково-пояснювальної записки: вступ, визначення параметрів вхідних зображень для системи розпізнавання знаків, методи параметричної класифікації дорожніх знаків, аналіз форми дорожніх знаків за допомогою, ідентифікація дорожніх знаків у системах розпізнавання, порівняльний аналіз інструментів, висновки.
- 5 Перелік графічного матеріалу: структурна схема комп'ютерної системи, блок-схема модуля попередньої обробки інформації.

Перелік ілюстративного матеріалу: результат виконання роботи по
результат розпізнавання знаку обмеження швидкості.

6 Консультанти розділів роботи наведені в таблиці 1.

Таблиця 1 — Консультанти роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	phd,ст. вик. каф. ОТ Обертюх М. Р.	21.03.25 	13.05.25 
Нормоконтроль	ас. каф. ОТ Швець С. І.	14.05.25 	30.05.25

7. Дата видачі завдання 21.03.2025 р.

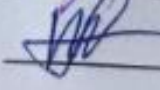
8. Календарний план приведений в таблиці 2.

Таблиця 2 — Календарний план

№ з/п	Назва етапів дипломної роботи	Строк виконання	Підпис
1	Постановка задачі розробки системи розпізнавання знаків	21.03.25	
2	Аналіз існуючих підходів визначення основних параметрів дорожніх знаків	21.03.25— 30.03.25	
3	Аналіз методів попередньої обробки вхідного зображення	21.03.25— 30.03.25	
4	Визначення методів класифікації та розпізнавання дорожніх знаків	31.03.25— 13.04.25	
5	Розробка комп'ютерної програми для моделювання задачі розпізнавання дорожніх знаків та її тестування	14.04.25— 27.04.25	
6	Оформлення пояснювальної записки та ілюстративного матеріалу	28.04.25— 11.05.25	
7	Перевірка якості виконання бакалаврської кваліфікаційної роботи та усунення недоліків	13.05.25	

Студент

Керівник роботи

Олександр СТАДНИК

phd,ст. вик. каф. ОТ Обертюх

АНОТАЦІЯ

УДК 004.42

Стадник О.В. Комп'ютерна система розпізнавання дорожніх знаків. Бакалаврська кваліфікаційна робота зі спеціальності 123 «Комп'ютерна інженерія», освітня програма «Системне програмування». Вінниця: ВНТУ, 2025. 90 с.

На укр. мові. Бібліогр.: 16, рис.: 22, табл.: 4.

У бакалаврській кваліфікаційній роботі розроблено комп'ютерну систему розпізнавання дорожніх знаків, що забезпечує надійне виявлення та ідентифікацію знаків у реальних дорожніх умовах для підвищення безпеки руху та підтримки автоматизованих систем керування транспортними засобами.

Проведено аналіз теоретичних основ і алгоритмів розпізнавання, включаючи огляд літератури, визначення параметрів вхідних зображень і характеристик типів дорожніх знаків відповідно до правил дорожнього руху України.

Реалізовано програмну систему на мові Python з використанням спеціалізованих бібліотек і фреймворків глибокого навчання, проведено порівняльний аналіз інструментів розробки та тестування, яке підтвердило високу точність розпізнавання.

Ключові слова: розпізнавання дорожніх знаків, параметрична класифікація, перетворення Хафа, попередня обробка зображень.

ABSTRACT

Stadnyk O.V.

Computer system for recognizing road signs. Bachelor's thesis in the specialty 123 "Computer Engineering," educational program "System Programming." Vinnytsia: VNTU, 2025. 90 p.

In Ukrainian. Bibliography: 16, 23 figures, 4 appendices.

This bachelor's thesis develops a computer system for recognizing road signs, which provides reliable detection and identification of signs in real road conditions to improve traffic safety and support automated vehicle control systems.

An analysis of the theoretical foundations and recognition algorithms was carried out, including a review of the literature, determination of the parameters of input images and characteristics of road sign types in accordance with the traffic rules of Ukraine.

A software system was implemented in Python using specialized libraries and deep learning frameworks, and a comparative analysis of development and testing tools was conducted, which confirmed the high accuracy of recognition.

Keywords: traffic sign recognition, parametric classification, Hough transform, image preprocessing.

ЗМІСТ

ВСТУП.....	4
1 ТЕОРЕТИЧНІ ОСНОВИ ТА АЛГОРИТМИ РОЗПІЗНАВАННЯ	
ДОРОЖНІХ ЗНАКІВ.....	6
1.1 Огляд літератури систем та методів розпізнавання знаків	6
1.2 Визначення параметрів вхідних зображень для системи	
розпізнавання знаків	7
1.3 Типи дорожніх знаків в правилах дорожнього руху України	10
1.4 Основні параметри дорожніх знаків	17
1.5 Аналіз характеристик типів дорожніх знаків та методи їх	
розпізнавання.....	20
2 МЕТОДИ ПАРАМЕТРИЧНОЇ КЛАСИФІКАЦІЇ ДОРОЖНІХ	
ЗНАКІВ.....	26
2.1 Отримання вхідного зображення.....	26
2.2 Виявлення дорожніх знаків	27
2.3 Попередня обробка зображень.....	28
2.4 Аналіз форми дорожніх знаків.....	32
2.5 Аналіз форми дорожніх знаків за допомогою	
перетворення Хафа.....	35
2.6 Аналіз непрямо отриманих ознак у системах розпізнавання	
дорожніх знаків.....	38
2.7 Формування класифікаторів на основі обчислених ознак для розпізнавання	
дорожніх знаків.....	39
2.8 Ідентифікація дорожніх знаків у системах розпізнавання	41
2.9 Методи верифікації та ідентифікації дорожніх знаків.....	46
3 ПРОГРАМНА РЕАЛІЗАЦІЯ КОМП'ЮТЕРНОЇ СИСТЕМИ	
РОЗПІЗНАВАННЯ ДОРОЖНІХ ЗНАКІВ.....	49
3.1 Визначення сучасних програмних інструментів для розробки	
системи обробки зображень	49

3.1.3	Фреймворки глибокого навчання	53
3.2	Вибір інтегрованого середовища розробки (IDE) та інструментів розробки.....	54
3.3	Порівняльний аналіз інструментів та обґрунтування вибору Python для проекту	56
3.4	Попередня обробка вхідного зображення	59
3.5	Класифікація дорожнього знаку за його за геометричними ознаками	65
3.6	Результати роботи та тестування програми	68
ВИСНОВКИ		73
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ		74
ДОДАТОК А Технічне завдання		76
ДОДАТОК Б Протокол перевірки кваліфікаційної роботи		78
ДОДАТОК В Графічна частина		82
ДОДАТОК Г Ілюстративна частина		85
ДОДАТОК Д Лістинг програмного забезпечення		87

ВСТУП

Актуальність роботи викликана тим, що розробка інтелектуальних транспортних засобів, спрямована на зменшення кількості помилкових рішень водіїв і дорожньо-транспортних пригод, є однією з пріоритетних тем сучасних автомобільних технологій. У процесі створення таких транспортних засобів досліджується низка наукових і технічних завдань, серед яких особливе місце посідає розпізнавання дорожніх знаків. Системи призначені для своєчасного інформування або попередження водіїв про обмеження, небезпеки чи інші відомості, які передаються дорожніми знаками. Оскільки дорожні знаки розроблені таким чином, щоб привертати увагу водіїв завдяки своїм стандартизованим кольорам і формам, обробка цих характеристик є ключовим елементом функціонування таких систем.

У роботі розроблено систему розпізнавання дорожніх знаків, яка здатна виявляти та ідентифікувати знаки навіть за наявності візуальних артефактів, спричинених несприятливими погодними умовами, частковим перекриттям або іншими зовнішніми факторами.

Розроблений у цій роботі алгоритм складається з трьох основних етапів: попередня обробка, сегментація та ідентифікація. Етап попередня обробка передбачає обробку знака на зображенні для покращення якості зображення. На етапі сегментації виконується виділення областей зображення, що відповідають стандартизованим кольорам дорожніх знаків, з урахуванням впливу змін освітлення в навколишньому середовищі. На завершальному етапі ідентифікації визначається тип знака шляхом порівняння отриманих характеристик із базою еталонних шаблонів.

Складність алгоритму зосереджена переважно на етапі ідентифікації, оскільки аналіз часткових сегментів для визначення цілісної форми знака вимагає обчислень і використання методів комп'ютерного зору. Для подолання цих викликів застосовуються сучасні підходи, такі як аналіз контурів, методи машинного навчання та обробка просторових зв'язків між фрагментами зображення. Крім того,

для підвищення стійкості системи до візуальних артефактів використовуються техніки попередньої обробки, такі як нормалізація яскравості та фільтрація шумів, що дозволяють мінімізувати вплив зовнішніх факторів. Таким чином, розроблена система забезпечує надійне розпізнавання дорожніх знаків, сприяючи підвищенню безпеки дорожнього руху та ефективності роботи автоматизованих систем керування транспортними засобами.

Метою роботи є розширення функціональних можливостей комп'ютерної системи для ефективного розпізнавання дорожніх знаків, яка забезпечує надійне виявлення та ідентифікацію знаків у складних умовах видимості для підвищення безпеки дорожнього руху та підтримки автоматизованих систем керування транспортними засобами.

Для досягнення мети були поставлені наступні **завдання**:

- розробити алгоритму сегментації зображень для виділення областей дорожніх знаків;
- визначити метод, який ефективно ізолює регіони зображення, що містять дорожні знаки, на основі їхніх стандартизованих характеристик, таких як колір і форма, з урахуванням змін освітлення та шумів;
- розробити алгоритм відновлення цілісної форми дорожніх знаків на основі фрагментарних даних, використовуючи методи комп'ютерного зору та аналіз просторових зв'язків;
- розробити моделі ідентифікації дорожніх знаків;
- реалізувати алгоритм класифікації, який точно визначає тип знака шляхом порівняння виділених ознак із базою еталонних шаблонів;
- перевірити працездатність системи на практичних прикладах.

Об'єктом дослідження є процес розробки комп'ютерних систем розпізнавання дорожніх систем.

Предметом дослідження є методи фільтрації, сегментації, виділення контурів та розпізнавання дорожніх знаків на зображеннях.

1 ТЕОРЕТИЧНІ ОСНОВИ ТА АЛГОРИТМИ РОЗПІЗНАВАННЯ ДОРОЖНІХ ЗНАКІВ

Огляд літератури систем та методів розпізнавання знаків

Розпізнавання та виявлення дорожніх знаків є важливою складовою сучасних систем допомоги водієві. Такі системи підвищують безпеку, розпізнаючи знаки та класифікуючи їх. У [1] запропоновано поділ процесу на два етапи: виявлення (локалізація знаків за кольором, формою та розміром) та класифікація (нейронна мережа типу багатошаровий перцептрон). Це забезпечило високу точність розпізнавання та прийнятний час обробки. Аварії часто трапляються через неуважність або нерозпізнавання знаків. У [2] запропоновано систему, що поєднує одноракурсне виявлення з 3D-трекінгом, що дозволяє відстежувати кілька знаків одночасно.

Дорожні знаки регулюють рух транспорту і попереджають про небезпеки. У [3] розглянуто метод виявлення пошкоджених знаків із застосуванням SIFT і порівняння гістограм, що дозволяє локалізувати пошкодження та визначити їхню площу. У [4] розглянуто розпізнавання знаків в умовах зовнішнього освітлення, яке ускладнюється затіненням або зміною положення об'єкта. Використано метод опорних векторів (SVM), що показав стійкість до складних умов. Обидва тексти стосуються розробки та аналізу систем розпізнавання дорожніх знаків (TSR). У центрі уваги — обробка візуальних даних для ідентифікації знаків, навіть за умов поганої видимості, зумовленої погодою чи перешкодами. Представлені алгоритми використовують сегментацію за кольором та аналіз форми, зокрема, метод "лінійної гістограми" для реконструкції частково закритих знаків. Дослідження також розглядають класифікацію знаків, порівнюючи ефективність підходів на основі шаблонного зіставлення та нейронних мереж.

Для автоматизованих транспортних засобів у [5] було запропоновано алгоритм, що поєднує градієнт орієнтованої гістограми та SVM. Завдяки інтеграції візуальних механізмів досягнуто точність 98% і 5% хибних спрацьовувань. У [6] представлено трьохетапну систему, що враховує яскравість, колір та форму, з

використанням методу кореляції з корекцією країв, що забезпечило ефективне виявлення та розпізнавання знаків із низьким рівнем хибних спрацювань. У [7] досліджено використання кольорових і в градація сірого кольору зображень для розпізнавання знаків обмеження швидкості. Автори відзначили, що колірні зміни при різному освітленні можуть знижувати точність розпізнавання. У згаданих джерелах висвітлюються підходи до отримання та підготовки початкових зображень дорожніх знаків для завдань комп'ютерного зору, зокрема для їх класифікації та розпізнавання за допомогою нейронних мереж. Цей етап, відомий як збір та попередня обробка даних, є критично важливим для досягнення високої продуктивності та точності моделей.

Розпізнавання дорожніх знаків є ключовим елементом систем допомоги водієві (ADAS), спрямованим на підвищення безпеки дорожнього руху. У дослідженнях розглядаються різні методи виявлення та класифікації знаків, зокрема на основі кольору, форми, розміру, контрасту й текстурних ознак. Використовуються такі підходи, як нейронні мережі, метод опорних векторів (SVM), гістограма орієнтованих градієнтів, SIFT, AdaBoost, Hough-перетворення та інші. Ефективність систем значно залежить від умов освітлення, типу зображень (кольорове чи сіре), а також країни походження набору даних. Попри високі показники точності в деяких випадках (до 98%), проблеми з хибними спрацюваннями й адаптацією до різних умов залишаються актуальними. Загалом, дослідження свідчать про високий потенціал розвитку ADAS через вдосконалення алгоритмів розпізнавання дорожніх знаків.

1.2 Визначення параметрів вхідних зображень для системи розпізнавання знаків

Один зі шляхів отримання даних включає використання існуючих великих наборів зображень, які вже були зібрані та попередньо оброблені. Наприклад, в одному дослідженні використовується датасет Traffic Signs Preprocessed dataset, який базується на німецькому бенчмарку German Traffic Sign Recognition Benchmark (GTSRB). Цей датасет містить значну кількість зображень (понад 104

тисячі), розподілених за 43 різними класами дорожніх знаків, що представляють поширені категорії, такі як обмеження швидкості, попередження та заборони. Використання таких різноманітних та об'ємних датасетів забезпечує міцну основу для навчання надійних моделей розпізнавання. Етап збору даних передуює подальшим етапам обробки та моделювання.

Інший підхід, детально описаний в одному з джерел, демонструє процес зчитування зображень безпосередньо з файлів та подальшу роботу з ними. Для цього використовується програмна бібліотека OpenCV, а саме функція `cv2.imread`, яка дозволяє завантажити зображення за вказаним шляхом до файлу. Після завантаження зображення може бути виконано конвертацію колірного простору, наприклад, переведення з формату BGR (Blue-Green-Red), який використовується за замовчуванням в OpenCV, до формату RGB (Red-Green-Blue) за допомогою функції `cv2.cvtColor`.

Паралельно зі зчитуванням самих зображень, для завдань, що включають визначення місцезнаходження об'єктів (наприклад, обмежувальних рамок) та їх класифікацію, необхідна відповідна метадані або анотації. У розглянутому випадку, ці анотаційні дані, що містять інформацію про тип дорожнього знака та координати його обмежувальної рамки на зображенні (`xmin`, `ymin`, `xmax`, `ymax`), зберігаються в окремих XML-файлах. Ці файли парсяться (аналізуються) за допомогою інструментів, таких як `xml.etree.ElementTree`, для вилучення необхідних даних для кожного зображення. В результаті формується структурований набір даних (наприклад, у вигляді таблиці), який зв'язує зображення з його анотаціями, готуючи його для подальшого використання в процесі навчання.

Таким чином, першим кроком у розробці систем розпізнавання дорожніх знаків, як це представлено у джерелах, є збір та підготовка візуальних даних. Це може бути здійснено шляхом використання попередньо існуючих датасетів або шляхом програмного зчитування файлів зображень (наприклад, за допомогою OpenCV) та відповідних файлів анотацій (наприклад, XML), що дозволяє отримати як самі зображення, так і інформацію про об'єкти, які на них містяться. Ці методи

забезпечують необхідну вхідну інформацію для роботи алгоритмів машинного навчання, таких як згорткові нейронні мережі.

У роботах, що стосуються розпізнавання дорожніх знаків за допомогою нейронних мереж, підготовка початкового зображення є критично важливим етапом для досягнення високої точності. Хоча джерела не визначають єдиний набір "оптимальних" параметрів зображення, вони описують методи та характеристики, які сприяють ефективному розпізнаванню.

Стандартизація розміру та формату. Для коректної роботи нейронних мереж, зокрема згорткових, всі вхідні зображення повинні мати однаковий розмір. В одному з досліджень зображення масштабувалися до розміру 300x300 пікселів ($IM_SIZE = 300$). При зміні розміру зображення відповідно трансформуються координати обмежувальних рамок об'єктів, щоб вони відповідали новому масштабу. Зображення, як правило, представлені у колірному просторі, наприклад, RGB, хоча може виконуватися конвертація колірного простору під час зчитування.

Якість зображення та ефективність розпізнавання залежить від якості даних. Низька роздільна здатність, складна фоновіа структура (сторонні об'єкти, схожі на шуканий) та різні спотворення, спричинені умовами зйомки (невдалі ракурси, погані погодні умови, різні кути та умови освітлення, шумові перешкоди, афінні та проєкційні спотворення), істотно знижують надійність сучасних методів. Для покращення якості зображень застосовуються методи попередньої обробки, такі як фільтрація зашумлених зображень та корекція яскравості і контрасту.

Нормалізація даних перед подачею зображень у нейронну мережу їх необхідно нормалізувати, наприклад, використовуючи статистику великих наборів даних, як-от ImageNet. Це допомагає стабілізувати процес навчання.

Різноманітність та кількість даних та використання великих та різноманітних наборів даних (датасетів) є фундаментальним для навчання надійних моделей. Аугментація даних (перевороти, повороти, обрізання) застосовується для штучного збільшення обсягу та різноманітності навчальної вибірки, що допомагає моделі краще узагальнювати і зменшує ризик перенавчання.

Таким чином, хоча конкретні "оптимальні" числові параметри (як-от роздільна здатність) можуть варіюватися залежно від задачі та моделі, ключовими аспектами для ефективного розпізнавання є стандартизація розміру та формату, висока якість зображень, їх нормалізація, а також значна кількість та різноманітність навчальних даних, досягнута в тому числі за допомогою аугментації. Згорткові нейронні мережі навчаються виявляти характерні ознаки об'єктів на основі цих підготовлених зображень.

1.3 Типи дорожніх знаків в правилах дорожнього руху України

Попереджувальні дорожні знаки (див. рис. 1.1) відіграють ключову роль у забезпеченні безпеки дорожнього руху. Їхнє основне призначення полягає в інформуванні водіїв про наближення до ділянок дороги, що потенційно можуть становити загрозу для транспортного засобу, інших учасників дорожнього руху або пішоходів. Такі знаки попереджають про зміну дорожніх умов, небезпеку або особливості інфраструктури, що вимагають підвищеної уваги.

У відповідь на отриману інформацію водій повинен оперативно оцінити ситуацію та вжити необхідних заходів задля зниження рівня ризику. До таких дій належать зменшення швидкості, приведення транспортного засобу у стан готовності до екстреної зупинки або підвищення зосередженості, зокрема при наближенні до узбіч, перехресть, пішохідних переходів чи інших небезпечних зон.

Попереджувальні знаки, відповідно до національного стандарту ДСТУ 4100:2021, мають впізнавану трикутну форму з червоною окантовкою. Основне тло полотна виконується білим кольором, а піктограми — у чорному. Таке поєднання кольорів сприяє високій помітності знаків у різних погодних та освітлювальних умовах. Виняток становлять окремі знаки, що повідомляють про перетин залізничних колій або вказують напрямок повороту: вони можуть мати дещо іншу графічну побудову чи елементи дизайну, зокрема наявність декількох червоних смуг або змінену орієнтацію зображення.

Важливо підкреслити, що попереджувальні знаки не накладають заборон чи прямих обмежень на маневри водія, а лише сигналізують про потребу змінити

поведінку з урахуванням можливої небезпеки. Незважаючи на відсутність юридичної заборони, ігнорування цих знаків часто призводить до аварійних ситуацій. Саме тому їх слід розглядати як обов'язковий елемент ефективної інформаційно-запобіжної системи на автошляхах.



Рисунок 1.1 — Попереджувальні дорожні знаки

Отже, попереджувальні дорожні знаки виконують не лише інформаційну, а й профілактичну функцію. Їх правильне сприйняття та своєчасна реакція водія є передумовою зменшення кількості дорожньо-транспортних пригод та підвищення загального рівня безпеки на автошляхах.

Заборонні дорожні знаки (рис. 1.2) виконують регуляторну функцію, чітко встановлюючи заборону на здійснення певних маневрів, таких як обгін, поворот, зупинка, стоянка, в'їзд, розворот тощо. Вони мають обов'язковий характер і є юридично значущими – порушення їхніх вимог розцінюється як пряме порушення Правил дорожнього руху України та тягне за собою адміністративну відповідальність, зокрема у вигляді штрафів.

Ця категорія також охоплює знаки, що скасовують раніше встановлену заборону. Візуально всі заборонні знаки мають круглу форму з білим фоном. Заборонні знаки позначаються червоною окантовкою, а ті, що скасовують заборону, – чорними елементами. У символіці використовуються переважно червоний, чорний і синій кольори, що забезпечує їх чітке сприйняття за будь-яких умов освітлення.



Рисунок 1.2 — Заборонні дорожні знаки

Зазвичай ці знаки встановлюються безпосередньо перед небезпечною ділянкою або перехрестям: у межах населених пунктів – на відстані не більше 25 метрів, поза межами — до 50 метрів. Дія заборони припиняється або відповідним знаком «кінець обмеження», або після перетину перехрестя. Водії повинні добре знати й розрізняти ці знаки, оскільки вони безпосередньо впливають на безпеку дорожнього руху та правову відповідальність.

Знаки пріоритету (рис. 1.3) відіграють важливу роль у регулюванні черговості проїзду на перехрестях, перетинах доріг, а також на вузьких ділянках проїжджої частини, де ширина полотна не дозволяє одночасний рух в обох напрямках. Їх головна функція — надання водіям чітких вказівок щодо порядку переваги в русі за відсутності світлофорного регулювання. Найпоширенішими

представниками цієї групи є знаки «Головна дорога», «Дати дорогу», «Проїзд без зупинки заборонено», «Перевага зустрічного руху» тощо.



Рисунок 1.3 — Знаки пріоритету

Ці знаки вирізняються серед інших тим, що не мають уніфікованої форми: вони можуть бути трикутними, круглими, квадратними або ромбоподібними. Їхня кольорова гама включає білий, жовтий, червоний, чорний та синій кольори — для чіткого сприйняття у різних дорожніх умовах. Встановлюються безпосередньо перед відповідною ділянкою — початком головної дороги, перехрестям, виїздом із другорядної дороги. Знак «Кінець головної дороги» інформує про завершення пріоритету. Знання цих знаків є критично важливим для безпечної та відповідальної поведінки на дорозі.

Наказові, дорожні (рис. 1.4) знаки є важливою складовою системи регулювання дорожнього руху. Вони встановлюють обов’язкові правила поведінки для водіїв, зокрема вказують необхідність здійснення певного маневру — повороту, розвороту, руху прямо або об’їзду. Ігнорування таких приписів трактується як порушення Правил дорожнього руху України та тягне за собою адміністративну відповідальність у вигляді штрафу.

Приклади: 4.1 «Рух прямо», 4.3 «Рух праворуч», 4.6 «Рух тільки для вантажних автомобілів», 4.8 «Об’їзд перешкоди праворуч або ліворуч».

Окрім регламентації маневрів, наказові знаки також позначають окремі смуги руху для пішоходів чи велосипедистів. Наприклад, дорожня інфраструктура може передбачати пішохідну або велосипедну доріжку, де рух дозволений виключно відповідним учасникам.



Рисунок 1.4 — Наказові дорожні знаки

Знаки, що регулюють запровадження або скасування особливих режимів дорожнього руху, виконують комплексну функцію, вони одночасно мають інформативний і дозвільний характер називаються інформаційно-дозвільними знаками (рис 1.5). Їх основне призначення — повідомити учасників дорожнього руху про введення специфічних умов пересування на певній ділянці дороги, а також вказати на дозвіл здійснення окремих дій у межах визначеної зони.

До цієї групи належать дорожні знаки, що позначають автомагістралі, житлові зони, пішохідні переходи, велосипедні та пішохідні доріжки, зупинки громадського транспорту, початок і кінець населеного пункту. Такі знаки суттєво впливають на поведінку водіїв, змінюючи допустимі режими руху, швидкість, типи дозволеного транспорту тощо. Зовні ці знаки мають переважно прямокутну або квадратну форму з синім або зеленим фоном. Вони є обов'язковими до сприйняття і правильного розуміння, оскільки забезпечують ефективну орієнтацію в дорожній інфраструктурі та сприяють безпеці всіх учасників руху.

Приклади: 5.1 «Автомагістраль», 5.3 «Початок населеного пункту», 5.5 «Кінець населеного пункту».

Інформаційно-вказівні знаки є невід'ємною складовою дорожньої інфраструктури, головна мета яких — надання учасникам дорожнього руху орієнтовної, географічної або сервісної інформації. Ці знаки повідомляють про місце розташування населених пунктів, вулиць, річок, об'єктів інфраструктури (музеїв, готелів, автозаправних станцій, СТО), пішохідних переходів, зупинок громадського транспорту, а також встановлюють або рекомендують певний режим

руху.

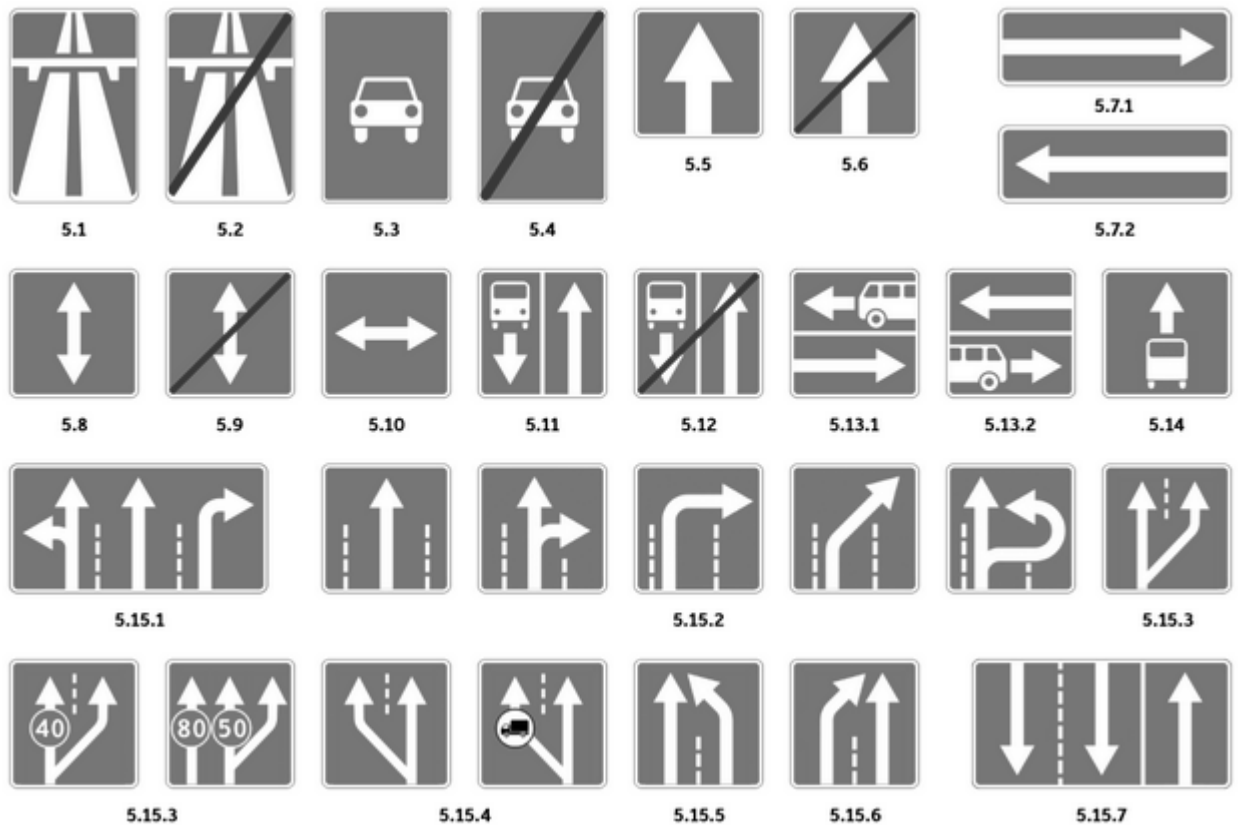


Рисунок 1.5 — Інформаційно-дозвільні знаки

Інформаційно-вказівні знаки (рис. 1.6) не є заборонними чи наказовими, однак мають важливе значення для навігації, особливо у незнайомій місцевості чи під час подорожей.

Вони сприяють своєчасному прийняттю рішень водіями, що позитивно впливає на безпеку та комфорт дорожнього руху.

Зовні такі знаки, як правило, мають прямокутну або квадратну форму. Фон знаків — синій, білий або зелений, залежно від їхнього призначення. Тимчасові інформаційні знаки виділяються жовтим фоном для підвищення помітності.

Знаки сервісу (рис.1.7) відіграють ключову роль у забезпеченні інформаційної підтримки водіїв і пасажирів під час пересування, особливо на міжміських та міжнародних маршрутах. Основне призначення цих знаків полягає в інформуванні про наявність поблизу об'єктів обслуговування, таких як автозаправні

станції, СТО, медичні установи, пункти зв'язку, готелі, кемпінги, ресторани, туалети, місця для стоянки, зони відпочинку, туристичні об'єкти тощо.



Рисунок 1.6 — Інформаційно-вказівні знаки



Рисунок 1.7 — Інформаційно-вказівні знаки

Встановлення знаків сервісу дозволяє водіям завчасно планувати зупинки, обирати оптимальні маршрути з урахуванням наявної інфраструктури та вчасно

отримувати необхідні послуги. Це особливо актуально на автомагістралях і трасах із тривалими відстанями між населеними пунктами.

Зовні ці знаки мають переважно прямокутну форму з синім фоном, на якому розміщено біле графічне зображення, що символізує відповідний об'єкт чи послугу. Їх чітке графічне оформлення сприяє легкому сприйняттю навіть на високих швидкостях руху.

Позначають наявність певних послуг або об'єктів сервісного обслуговування поблизу дороги. Зазвичай мають квадратну форму з синім фоном і білими символами.

Розпізнавання дорожніх знаків є критично важливою складовою безпечного дорожнього руху, як для водіїв, так і для автоматизованих систем допомоги керування (ADAS). Ключовими параметрами, що забезпечують ефективне сприйняття та ідентифікацію знаків, є форма, колір, контур, символи та місце розміщення.

Саме ці характеристики дозволяють швидко та однозначно визначити тип знака, навіть при обмеженій видимості або високій швидкості руху. Наприклад, трикутна форма з червоною окантовкою асоціюється з попередженням, а кругла з червоним обрамленням — із заборonoю.

У правилах дорожнього руху України закладено пріоритет дії знаків у разі конфлікту з іншими засобами регулювання. Найвищий пріоритет мають знаки пріоритету, за ними йдуть наказові та заборонні.

Попереджувальні є інформаційними, але нехтування ними може призвести до порушень або аварій. Визначення параметрів дорожніх знаків для їх розпізнавання є основним завданням для забезпечення безпеки на дорозі.

Основні параметри дорожніх знаків

Основними геометричними ознаками дорожніх знаків, що є важливими для їх розпізнавання, є загальна форма знака, краї та контури, внутрішні візерунки (піктограми), положення та розмір, які зведено в таблицю 1.1.

Дорожні знаки стандартизовані і мають певні форми, які часто корелюють з їхнім типом або функцією.

Таблиця 1.1 — Визначення призначення та пріоритету дорожніх знаків

Критерій	Попереджувальні знаки	Заборонні знаки	Знаки пріоритету	Наказові знаки
Призначення	Попередження про потенційну небезпеку або зміну дорожніх умов	Повна або часткова заборона певних дій (руху, обгону, зупинки)	Визначення черговості проїзду на перехрестях або вузьких ділянках	Вказівка обов'язкового напрямку або дії (рух прямо, поворот тощо)
Форма	Рівносторонній трикутник з вершиною вгору	Коло (рідше прямокутник для касування)	Різна: трикутник, ромб, квадрат тощо	Коло (рідше прямокутник – для окремих знаків)
Колір фону	Білий	Білий	Білий, жовтий, синій	Синій (іноді білий фон для окремих варіантів)
Контур	Червона	Червона (чорна для скасування заборони)	Червона, жовта або відсутня	Відсутня
Колір символів/зображення	Чорний	Чорний, червоний, синій	Чорний, білий, жовтий	Білий
Місце встановлення	Завчасно перед небезпечною ділянкою	Перед зоною дії заборони (25 м у місті, 50 м поза межами)	Безпосередньо перед перехрестям, виїздом чи початком ділянки	Перед ділянкою, де слід виконати дію
Характер дії	Інформаційний, попереджувальний	Обов'язковий до виконання	Обов'язковий до виконання	Обов'язковий до виконання
Юридична сила (пріоритет)	Середній	Високий	Найвищий	Високий
Штраф за невиконання	Немає (опосередковано, якщо є ДТП)	Передбачено адміністративне покарання	Передбачено адміністративне покарання	Передбачено адміністративне покарання

Наприклад, заборонні знаки зазвичай мають круглу форму з яскраво-червоним обведенням. Попереджувальні знаки найчастіше виконані у формі трикутника з червоним обведенням. Розпорядчі знаки можуть мати форму кола або квадрата синього кольору. Знаки пріоритету та знаки особливих приписів також мають специфічні форми, такі як ромб, перевернутий трикутник або прямокутник. Ці стандартизовані форми є ключовими візуальними ознаками для початкової ідентифікації категорії знака.

Розпізнавання об'єктів традиційно включало етапи виявлення країв та кутів на зображенні. Сучасні згорткові нейронні мережі (CNN), хоча й навчаються виявляти ознаки автоматично, на низьких рівнях обробляють основні геометричні компоненти, такі як прямі та криві лінії. Операція згортки в CNN використовує фільтри, які можуть бути налаштовані для виявлення таких ліній та інших базових форм. Техніки, як перетворення відстані (distance transform), працюють на бінаризованих зображеннях або картах країв, аналізуючи просторове розподілення інформації в межах об'єкта. Дескриптори ознак, такі як HOG, аналізують розподіл градієнтних векторів, що тісно пов'язано з орієнтацією та інтенсивністю країв.

Хоча самі піктограми є графічними символами, їхня структура та розташування в межах геометричної форми знака є важливими ознаками. Методи вилучення ознак, такі як HOG, аналізують градієнтні розподіли, що дозволяє захопити деталі внутрішніх малюнків. Перетворення відстані також допомагає розподілити інформацію про піктограму по всій області знака. Фільтри Габора (Gabor wavelets) виявляють особливості на основі орієнтації та частоти, що корисно для аналізу як країв знака, так і структури внутрішніх символів.

Для задач розпізнавання об'єктів важливо визначити місцезнаходження знака на зображенні, що зазвичай робиться за допомогою обмежувальних рамок (bounding box). Координати та розміри цих рамок є геометричними властивостями, які система має передбачити. Методи розпізнавання враховують варіації в положенні, масштабі та орієнтації знаків, часто використовуючи аугментацію даних (обрізання, повороти, перевероти) для збільшення стійкості моделі до таких геометричних трансформацій.

Ці геометричні ознаки разом із кольором дозволяють системам комп'ютерного зору ефективно ідентифікувати та класифікувати дорожні знаки навіть за наявності різних спотворень та ускладнених умов зйомки.

Аналіз характеристик типів дорожніх знаків та методи їх розпізнавання

Розпізнавання дорожніх знаків є ключовою складовою систем забезпечення безпеки дорожнього руху, зокрема в автоматизованих системах допомоги водію та автономних транспортних засобах. Ефективність ідентифікації знаків залежить від їхніх стандартизованих характеристик, таких як форма, колір, контур, символи та місце розташування. Форма дорожніх знаків є стандартизованою та чітко визначеною відповідно до міжнародних норм, таких як Віденська конвенція про дорожні знаки.

Основні типи форм включають:

- круглі знаки (заборонні, наприклад, обмеження швидкості, або наказові, наприклад, обов'язковий напрямок);
- трикутні знаки (попереджувальні, часто з червоною облямівкою);
- прямокутні або квадратні знаки (інформаційні чи вказівні);
- восьмикутні знаки (наприклад, знак «Стоп»).

Форма є первинною ознакою, оскільки шаблони створюються для кожної стандартної форми (наприклад, круг для обмеження швидкості). Метод ефективний в умовах чіткої видимості, але чутливий до поворотів і масштабування, що вимагає додаткового нормування зображення. Дескриптори локальних особливостей (SIFT, Форма опосередковано враховується через аналіз контурів і ключових точок. Наприклад, SIFT фіксує геометричні особливості трикутників або кіл, що робить метод стійким до поворотів і змін масштабу, але потребує обчислювальних ресурсів.

Алгоритм Віюлі-Джонса використовує прямокутні ознаки для виявлення контурів форм, таких як кути трикутника чи межі кола. Метод швидкий, але менш точний для складних форм через спрощені ознаки. Згорткові нейронні мережі (CNN): Форма вилучається автоматично згортковими шарами, які аналізують

просторові структури. CNN є найгнучкішим методом, оскільки адаптується до різних форм без необхідності явного визначення шаблонів. Для простих форм (круг, трикутник) ефективним є порівняння з шаблонами або алгоритм Віоли-Джонса, тоді як для складних умов (зміни ракурсу, часткове перекриття) перевагу мають

Колір є стандартизованою ознакою дорожніх знаків, що відповідає їхньому функціональному призначенню:

- червоний (заборонні або попереджувальні знаки, наприклад, червона облямівка трикутників);
- синій (наказові або інформаційні знаки);
- білий, зелений, жовтий (фони інформаційних або вказівних знаків);
- контрастні комбінації (наприклад, білий текст на червоному фоні для знака «Стоп»);

Вплив на методи розпізнавання при порівнянні з шаблонами — колір використовується для попередньої сегментації зображення (наприклад, виділення червоних областей). Метод залежить від стабільного освітлення, оскільки зміни кольору через тіні чи погоду знижують точність. Детектори локальних особливостей (SIFT, SURF), колір не є первинною ознакою, оскільки детектори фокусуються на градієнтах яскравості. Однак попередня колірна сегментація може зменшити кількість ключових точок для аналізу. Алгоритм Віоли-Джонса — колір опосередковано враховується через контрастність між регіонами (наприклад, червона облямівка на білому фоні). Метод менш чутливий до змін кольору, але потребує попередньої обробки для виділення контрастних областей. Згорткові нейронні мережі (CNN) — колір обробляється через канали RGB у вхідному зображенні. CNN ефективно розпізнає колірні комбінації навіть при змінах освітлення, якщо модель навчена на різноманітних даних. Колір є критичним для попередньої сегментації в усіх методах. Порівняння з шаблонами та Віола-Джонс ефективні в стабільних умовах, тоді як CNN і SIFT/SURF краще справляються зі змінами освітлення.

Контур дорожнього знака визначається чіткими межами, які відокремлюють знак від фону, наприклад, червона облямівка трикутних попереджувальних знаків. Рівні межі кіл або прямокутників чи контраст між контуром і фоном (наприклад, білий фон у червоній облямівці).

Контур є частиною шаблону, що порівнюється з вхідним зображенням. Чіткі контури полегшують зіставлення, але спотворення (наприклад, через дощ чи бруд) знижують ефективність. Детектори локальних особливостей (SIFT, SURF) – контури фіксуються через градієнти яскравості в ключових точках, що робить методи стійкими до часткових спотворень контуру.

Алгоритм Віоли-Джонса — прямокутні ознаки ефективно виявляють чіткі контури, особливо для трикутних чи круглих знаків, але менш точні при розмитих межах. Згорткові нейронні мережі (CNN) — контури вилучаються згортковими фільтрами на ранніх шарах, що забезпечує високу стійкість до деформацій чи шумів. Чіткі контури найкраще обробляються порівнянням з шаблонами та Віолою-Джонсом, тоді як SIFT/SURF і CNN ефективніші в умовах розмитості чи часткового перекриття.

Символи включають текстові написи (наприклад, «50» для обмеження швидкості, «STOP») та піктограми (стрілки, зображення пішоходів). Вони стандартизовані за шрифтом, розміром і розташуванням у межах знака. Порівняння з шаблонами, коли символи є частиною шаблону, що забезпечує точну ідентифікацію при відповідності шрифту та розміру.

Детектори локальних особливостей (SIFT, SURF) — символи аналізуються як набір ключових точок, що дозволяє розпізнавати піктограми навіть при поворотах чи масштабуванні. Алгоритм Віоли-Джонса — менш ефективний для детального аналізу символів через спрощені прямокутні ознаки, але може використовуватися для первинного виявлення знака перед застосуванням OCR (оптичного розпізнавання символів). Згорткові нейронні мережі (CNN) найкраще розпізнають символи завдяки навчанню на різноманітних даних, що включають варіації шрифтів і піктограм. Для символів CNN є найефективнішим методом через

здатність обробляти складні візуальні патерни, тоді як порівняння з шаблонами та SIFT/SURF потребують чітких умов.

Таблиця 1.2 — Залежність характеристики типів дорожніх знаків від застосування методу їх розпізнавання

Параметр	Характеристика	Порівняння з шаблонами		Алгоритм Віолі-Джонса	Згорткові нейронні мережі (CNN)
Форма	Стандартні геометричні форми: круг, трикутник, прямокутник, восьмикутник.	Ефективний для чітких форм у стабільних умовах. Чутливий до поворотів і масштабування, потребує нормування зображення.	Стійкий до змін масштабу та поворотів завдяки аналізу ключових точок і градієнтів. Вимагає значних обчислень.	Виявляє форми через прямокутні ознаки, ефективний для простих форм (круг, трикутник), але менш точний для складних.	Автоматично вилучає форми згортковими шарами, стійкий до спотворень і змін ракурсу.
Колір	Стандартизована палітра: червоний (заборонні, попереджувальні), синій (наказові), білий/зелений/жовтий (інформаційні).	Використовується для попередньої сегментації (наприклад, виділення червоних об'єктів). Чутливий до змін освітлення.	Колір опосередкований враховується через градієнти яскравості. Потребує попередньої сегментації для оптимізації.	Контрастність кольорів аналізується через прямокутні ознаки. Менш чутливий до змін кольору, але потребує обробки.	Ефективно обробляє кольори через RGB-канали, стійкий до змін освітлення при навчанні на різноманітних даних.
Контур	Чіткі межі, що відокремлюють знак від фону (наприклад, червона облямівка трикутника).	Контур є частиною шаблону, ефективний при чітких межах, але чутливий до розмитості чи перекриття.	Аналізує контури через градієнти ключових точок, стійкий до часткових спотворень.	Виявляє контури через прямокутні ознаки, ефективний для чітких меж, але менш точний при розмитості.	Вилучає контури згортковими фільтрами, стійкий до шумів і часткового перекриття.
Символи	Текст (наприклад, «50», «STOP») та піктограми стандартизовані за шрифтом і розташуванням.	Символи порівнюються як частина шаблону, точний при чітких умовах, але чутливий до змін ракурсу.	Аналізує символи як набір ключових точок, стійкий до поворотів і масштабування, але потребує обчислень.	Менш ефективний для детального аналізу символів, потребує OCR для тексту після виявлення знака.	Найкраще розпізнає символи завдяки навчанню на різноманітних даних, шрифтів і піктограм.

Дорожні знаки зазвичай розташовані в передбачуваних зонах (узбіччя, верхня частина зображення, над дорогою). Це контекстуальна ознака, яка залежить від перспективи камери та дорожньої інфраструктури. Вплив на методи розпізнавання. Порівняння з шаблонами — обмеження аналізу до передбачуваних

зон зменшує кількість порівнянь, підвищуючи швидкість. Метод залежить від точного визначення областей інтересу (ROI).

Дескриптори локальних особливостей (SIFT, SURF) — місце розташування використовується для попередньої сегментації, що зменшує кількість ключових точок для аналізу. Алгоритм Віоли-Джонса — спливаюче вікно застосовується до передбачуваних зон, що прискорює виявлення, але потребує правильного вибору масштабу. Згорткові нейронні мережі (CNN) — контекст розташування враховується автоматично через аналіз просторових зв'язків у зображенні, що робить метод менш залежним від ручного визначення зон. Тобто місце розташування оптимізує всі методи шляхом звуження області пошуку, але CNN є найгнучкішим завдяки автоматичному аналізу контексту. Порівняння з шаблонами найкраще для чітких умов і стандартизованих знаків, але чутливе до спотворень і потребує попередньої обробки. SIFT/SURF — стійкі до змін масштабу, поворотів і часткових спотворень, але обчислювально складні, підходять для складних умов. Алгоритм Віоли-Джонса швидкий для первинного виявлення, ефективний у реальному часі, але менш точний при складних формах чи розмитих контурах. Згорткові нейронні мережі (CNN) — найбільш універсальний метод, стійкий до шумів, змін освітлення та перекриття, але вимагає значних даних для навчання та обчислювальних ресурсів.

Вибір методу залежить від умов розпізнавання, доступних ресурсів і вимог до швидкості та точності. Для реального часу в стабільних умовах перевагу мають порівняння з шаблонами та Віола-Джонс, тоді як для складних сценаріїв (зміни освітлення, ракурсу) оптимальними є SIFT/SURF або CNN.

Отже, можна зробити висновок, що параметри дорожніх знаків — форма, колір, контур, символи та місце розташування відіграють ключову роль у виборі методу розпізнавання. Порівняння з шаблонами та алгоритм Віоли-Джонса ефективні для чітких умов і стандартизованих знаків, але чутливі до спотворень. Дескриптори SIFT/SURF забезпечують стійкість до змін масштабу та ракурсу, але вимагають значних обчислювальних ресурсів. Згорткові нейронні мережі є найбільш універсальним методом, оскільки автоматично вилучають і комбінують

усі параметри, забезпечуючи високу точність навіть в умовах шумів, змін освітлення чи часткового перекриття. Вибір методу залежить від компромісу між швидкістю, точністю та доступними обчислювальними ресурсами.

2 МЕТОДИ ПАРАМЕТРИЧНОЇ КЛАСИФІКАЦІЇ ДОРОЖНІХ ЗНАКІВ

2.1 Отримання вхідного зображення

Етап збору даних, є початковим і критично важливим у процесі розпізнавання дорожніх знаків. На цьому етапі здійснюється захоплення вхідних даних за допомогою одного або кількох сенсорів, найчастіше відеокамер, які фіксують послідовність зображень. Якість, надійність і придатність даних для подальшої обробки значною мірою залежать від ефективності цього етапу, оскільки вхідні дані формують основу для всіх наступних операцій розпізнавання. Для підвищення якості даних можуть застосовуватися додаткові сенсори, що забезпечують різноманітність і деталізацію інформації.

У дослідженні, описаному в [4], запропоновано використання двох камер із різними характеристиками. Перша камера, оснащена ширококутним об'єктивом (wide-camera), орієнтована на напрямок руху транспортного засобу та призначена для первинного виявлення потенційних дорожніх знаків. Друга камера, обладнана телеоб'єктивом (tele-camera), має можливість змінювати напрямок відповідно до команд алгоритму, що дозволяє отримувати деталізовані зображення окремих кандидатів на знаки. Ширококутна камера забезпечує широкий огляд для виявлення можливих об'єктів, тоді як телекамера, керуючись прогнозуванням траєкторії руху, фокусується на кожному кандидаті для отримання зображення з вищою роздільною здатністю. Однак використання рухомої телекамери підвищує ймовірність пропуску кандидатів у сценах із кількома знаками, що є обмеженням цього підходу.

Інший підхід, описаний у [14], передбачає використання двох ідентичних камер для створення стереопари зображень. Стереозображення дозволяють застосовувати принципи епіполярної геометрії та попередньо визначені параметри калібрування для оцінки розташування виявленого об'єкта на парному зображенні. За допомогою критерію подібності, заснованого на нормалізованій кореляції, визначається точне положення цільового об'єкта. Подвійна надлишковість цього методу забезпечує високу надійність, оскільки втрата об'єкта можлива лише за

умови його невидимості для обох камер. Такий підхід підвищує стійкість системи до часткових перекриттів або складних умов видимості.

Для забезпечення високої якості даних на етапі збору можуть також використовуватися додаткові сенсори, такі як інфрачервоні камери чи лідари, які дозволяють отримувати інформацію в умовах обмеженої видимості (наприклад, у темряві або за поганої погоди). Вибір сенсорів залежить від вимог до системи, зокрема швидкості обробки, точності та стійкості до зовнішніх факторів.

2.2 Виявлення дорожніх знаків

Етап виявлення дорожніх знаків спрямований на локалізацію потенційних кандидатів на зображенні, отриманому з сенсорів. Основна мета цього етапу полягає в ефективному використанні обчислювальних ресурсів, доступних на момент розробки системи, для забезпечення швидкості та точності обробки. Зі зростанням обчислювальної потужності комп'ютерів сучасні методи виявлення стають дедалі складнішими, що дозволяє досягати кращих результатів. Дослідження, опубліковані в близькі часові періоди, зазвичай застосовують подібні техніки, адаптовані до поточного рівня технологій.

На етапі виявлення ключову роль відіграють стандартизовані характеристики дорожніх знаків, зокрема їхня форма та колір, які регламентуються міжнародними стандартами, такими як Віденська конвенція про дорожні знаки. Ці ознаки є основою для первинного відбору кандидатів. Наприклад, червоні трикутні знаки асоціюються з попередженнями, а сині круглі — з наказовими вказівками. Окрім очевидних характеристик, таких як форма та колір, сучасні алгоритми дедалі частіше використовують методи вилучення непрямих ознак, які не спостерігаються безпосередньо, але підвищують точність розпізнавання.

Алгоритми виявлення дорожніх знаків можна класифікувати за трьома основними підходами.

Аналіз кольору використовується для сегментації зображення за стандартизованими кольорами знаків (червоний, синій, білий). Цей підхід дозволяє швидко звужити область пошуку, але чутливий до змін освітлення.

Аналіз форми базується на виявленні геометричних контурів (круг, трикутник, прямокутник) за допомогою методів, таких як перетворення Хафа. Цей метод ефективний для чітких умов, але потребує додаткової обробки при спотвореннях.

Аналіз непрямо отриманих ознак включає використання дескрипторів локальних особливостей (наприклад, SIFT, SURF) або згорткових нейронних мереж для вилучення складних візуальних патернів, таких як текстури чи градієнти яскравості. Ці методи забезпечують стійкість до шумів і змін ракурсу, але вимагають значних обчислювальних ресурсів.

Для підвищення ефективності виявлення часто застосовується комбінація цих підходів. Наприклад, колірна сегментація може використовуватися для первинного виділення областей інтересу, після чого аналіз форми або непрямих ознак уточнює розташування знака. У сучасних системах згорткові нейронні мережі дедалі частіше стають основним інструментом, оскільки вони здатні автоматично вилучати та комбінувати всі типи ознак, забезпечуючи високу точність навіть у складних умовах.

Попередня обробка зображень

Обробка зображень є ключовою галуззю комп'ютерного зору, що займається аналізом, маніпулюванням та покращенням цифрових зображень. Фундаментальним етапом у цьому процесі є зчитування зображення, після чого часто застосовуються різноманітні фільтри для його підготовки до подальшого аналізу або для покращення візуальної якості. Розглянемо ці процеси детальніше, зосередившись на реалізації за допомогою бібліотеки OpenCV та основних принципах роботи Гауссового розмиву, Медіанного та Білатерального фільтрів.

Фільтрація зображень — це процес зміни або покращення зображення шляхом застосування математичної операції до кожного пікселя на основі його власного значення та значень сусідніх пікселів. Більшість фільтрів є "просторовими" фільтрами, що означає, що операції виконуються безпосередньо над пікселями зображення.

Згортка (Convolution) є фундаментальною операцією для багатьох фільтрів. Вона полягає в застосуванні "ядра" (або "маски", "фільтра") – невеликої матриці чисел – до кожного пікселя зображення. Ядро переміщається по зображенню, і в кожній позиції виконується поелементне множення значень ядра на відповідні значення пікселів у підматриці зображення, що перекривається ядром. Результати множення підсумовуються, а отримане значення стає новим значенням центрального пікселя в результуючому зображенні.

Гауссів розмиття (Gaussian Blur) є одним з найпоширеніших фільтрів для згладжування зображень та зменшення шуму. Він належить до класу лінійних фільтрів. Його ефективність полягає у використанні Гауссової функції (функції нормального розподілу) для обчислення ваг у ядрі фільтра. Пікселі, ближчі до центру ядра, отримують більші ваги, тоді як віддалені пікселі — менші. Це забезпечує плавний перехід і меншу чутливість до різких змін, ефективно розмиваючи дрібні деталі та шум.

Принцип роботи — кожен піксель нового зображення є зваженим середнім значенням пікселів з сусідства вихідного зображення, де ваги визначаються Гауссовою функцією.

Математична складова двовимірної Гауссової функції, центрована у $(0,0)$, має вигляд:

де x, y — координати пікселя відносно центру ядра;

σ (сигма) — стандартне відхилення Гауссового розподілу.

Стандартне відхилення Гауссового розподілу контролює "ширину" Гауссового ядра і, отже, ступінь розмиву. Більше σ означає сильніший розмив. 2

На практиці для дискретних зображень створюється Гауссове ядро (матриця) на основі цієї функції, яке потім застосовується до зображення за допомогою згортки. Мовою пайтон реалізується наступним чином.

Параметри `cv2.GaussianBlur(src, ksize, sigmaX)`: `src` — вхідне зображення; — розмір Гауссового ядра, кортеж (ширина, висота). Обидва значення повинні бути непарними та позитивними (наприклад, (5, 5)); `sigmaX`: — стандартне відхилення Гауссового ядра в напрямку `X`. Якщо 0, воно обчислюється автоматично на основі `ksize`. `sigmaY` (для напрямку `Y`) також може бути вказано, але зазвичай дорівнює `sigmaX`. Застосування — зменшення Гауссового шуму, згладжування зображень, передобробка для виявлення країв (наприклад, для Canny Edge Detector).

Медіанний Фільтр (Median Filter) є нелінійним фільтром, який дуже ефективний для зменшення шуму типу "сіль і перець" (окремі яскраві або темні пікселі). На відміну від Гауссового розмиву, який усереднює значення, медіанний фільтр замінює значення центрального пікселя медіаною всіх пікселів у його сусідстві (вікні). Принцип роботи — ядро (вікно) фільтра переміщається по зображенню; для кожної позиції ядра, значення всіх пікселів, що потрапляють у вікно, збираються в список; цей список сортується; медіанне значення (середнє значення відсортованого списку) стає новим значенням центрального пікселя. Дуже ефективний проти імпульсного шуму ("сіль і перець"). Зберігає краї зображення краще, ніж лінійні фільтри (наприклад, середній фільтр), оскільки він не створює нових, неіснуючих значень пікселів. Він просто вибирає одне з існуючих значень. Але серед недоліків — може трохи розмивати або "сплющувати" тонкі лінії та деталі. Обчислювально дорожчий, ніж лінійні фільтри, через операції сортування.

В мові Python параметри функції `cv2.medianBlur(src, ksize)`: `src`: — вхідне зображення; `ksize` — розмір апертури (вікна). Це одиночне непарне ціле число (наприклад, 3, 5, 7). Це означає, що ядро буде мати розмір `ksize` x `ksize`. Використовують для зменшення шуму "сіль і перець", обробка перед сегментацією або виявленням об'єктів.

Білатеральний фільтр — це потужний нелінійний фільтр, який чудово справляється зі зменшенням шуму, одночасно зберігаючи різкі краї. Він досягає цього, враховуючи дві ваги для кожного пікселя в сусідстві.

Просторова вага (spatial weight) заснована на просторовій відстані пікселів, подібно до Гауссового розмиву. Пікселі, що знаходяться ближче, мають більший вплив.

Діапазонна вага (intensity/range weight) заснована на різниці інтенсивності (кольору) між центральним пікселем і сусідніми пікселями. Пікселі зі схожими значеннями інтенсивності (кольором) мають більший вплив, тоді як пікселі зі значно відмінними значеннями інтенсивності мають менший вплив.

Значення пікселя обчислюється як зважене середнє значень сусідніх пікселів. Вага для кожного сусіднього пікселя є добутком його просторової ваги та діапазонної ваги. Це дозволяє фільтру згладжувати області з подібним кольором/інтенсивністю (зменшуючи шум), але не згладжувати через краї, де є значні перепади інтенсивності.

Математична основа — значення вихідного пікселя I_p у позиції p обчислюється як:

$$I_p = \frac{\sum_{q \in N_p} I_q \cdot W_p}{\sum_{q \in N_p} W_p}$$

де I_q — значення сусіднього пікселя у позиції q ,
 N_p — множина сусідніх пікселів,
 W_p — просторова Гауссова функція, що зменшується з відстанню між p і q ,
 $|I_p - I_q|$ — діапазонна Гауссова функція, що зменшується з різницею інтенсивності між I_p і I_q ,
 W_p — нормалізуючий коефіцієнт, що є сумою всіх ваг.

Високоефективний для зменшення шуму, особливо Гауссового, при збереженні гостроти країв. Зберігає деталі текстури в однорідних областях. Значно повільніший, ніж Гауссів або Медіанний фільтр, через складніші обчислення для кожного пікселя. Вимагає ретельного підбору параметрів `sigmaColor` та `sigmaSpace`.

Застосовують для зменшення шуму в фотографіях, покращення якості зображень для комп'ютерної графіки, передобробка для виявлення об'єктів у складних умовах.

Кожен з розглянутих фільтрів має свої унікальні властивості та найкраще підходить для певних типів задач та шуму. Гауссів розмив є швидким і простим для загального згладжування. Медіанний фільтр чудово справляється з імпульсним шумом, зберігаючи краї. Білатеральний фільтр є найскладнішим, але пропонує найкращий компроміс між зменшенням шуму та збереженням деталей, особливо країв. Вибір фільтра залежить від конкретних вимог застосування та характеристик шуму, присутнього в зображенні.

Аналіз форми дорожніх знаків

Аналіз форми є однією з ключових характеристик для ідентифікації дорожніх знаків на зображеннях, оскільки інформація про колір може бути нестабільною в умовах мінливого освітлення протягом дня, наприклад, через погодні зміни чи тіні. Форма дорожніх знаків, стандартизована міжнародними нормами, такими як Віденська конвенція про дорожні знаки, залишається надійною ознакою навіть за наявності візуальних артефактів. У цьому розділі розглядаються методи аналізу форми, їхнє застосування в задачах розпізнавання дорожніх знаків, а також особливості їхньої реалізації з використанням сучасних підходів комп'ютерного зору та штучного інтелекту.

У дослідженні, описаному в [6], початковим етапом аналізу форми є застосування алгоритму виявлення контурів Кенні (Canny edge detection), який дозволяє точно ідентифікувати межі об'єктів на зображенні. Виявлені контури обробляються за допомогою алгоритму ланцюгового зв'язування ребер (edge chaining) та полігональної апроксимації, запропонованої Павлідісом [7]. Ці методи дають змогу отримати прямі сегменти контурів, які відповідають геометричним компонентам дорожніх знаків. Передбачається, що транспортний засіб рухається в напрямку дороги, а дорожні знаки розташовані ортогонально до цього напрямку. Завдяки цьому припущенню форма знака не зазнає значних перспективних спотворень, що полегшує її аналіз.

Для трикутних дорожніх знаків, які є типовими для попереджувальних знаків,

а

л

г

о

допуск похибки. Цей підхід дозволяє точно виявляти трикутні форми, відфільтровуючи нерелевантні контури. Такий метод аналізу форми є ефективним для стандартизованих геометричних фігур, таких як трикутники, кола чи прямокутники, оскільки він спирається на чітко визначені геометричні характеристики.

Окрім цього, сучасні системи аналізу форми можуть інтегрувати додаткові техніки, такі як перетворення Хафа (Hough Transform), для виявлення специфічних геометричних фігур, або методи машинного навчання для адаптивного визначення контурів у складних умовах. Наприклад, у разі часткового перекриття знака чи наявності шумів алгоритми на основі згорткових нейронних мереж (CNN) здатні ефективно вилучати форму шляхом аналізу просторових зв'язків між контурами. Це забезпечує стійкість до перспективних спотворень і зовнішніх факторів, таких як тіні чи деформації.

Таким чином, аналіз форми відіграє центральну роль у системах розпізнавання дорожніх знаків, оскільки дозволяє надійне виявлення об'єктів навіть за умов нестабільного освітлення чи часткового перекриття. Подальший розвиток методів аналізу форми, зокрема з інтеграцією глибокого навчання, сприятиме підвищенню точності та ефективності таких систем, що є критично важливим для безпеки дорожнього руху та функціонування автоматизованих систем керування.

Аналіз форми є центральним елементом у системах розпізнавання дорожніх знаків, оскільки стандартизовані геометричні характеристики, такі як трикутники, прямокутники чи кола, дозволяють надійно ідентифікувати об'єкти навіть за умов обмеженої видимості. У цьому розділі розглядаються методи виявлення контурів і кутів, які застосовуються для визначення форми дорожніх знаків, із акцентом на алгоритми комп'ютерного зору.

На завершальному етапі аналізу форми перевіряється можливість формування рівностороннього трикутника шляхом комбінації виявлених ребер. Результати кожного етапу обробки, представлені ілюструють послідовність процесу. У дослідженні [8] застосовується метод виявлення кутів на бінарній масці,

отриманій після сегментації за кольором. Виявлення кутів базується на згортці бінарного зображення зі спеціальними масками для ідентифікації верхнього та нижнього лівого кутів трикутного знака. Маска для нижнього правого кута є симетричною до маски нижнього лівого кута.

Алгоритм передбачає сканування зображення для пошуку верхнього кута, після чого визначається область пошуку (рис. 2.1а) для нижнього лівого кута. У разі його виявлення формується область для нижнього правого кута (рисунок 2.1б). Для прямокутних знаків використовуються маски для 90-градусних кутів (рисунок 2.2а). Для кругових знаків після виявлення трьох кутів обчислюється окружність, і якщо більшість точок належить їй, об'єкт класифікується як круговий знак (рисунок 2.2б).

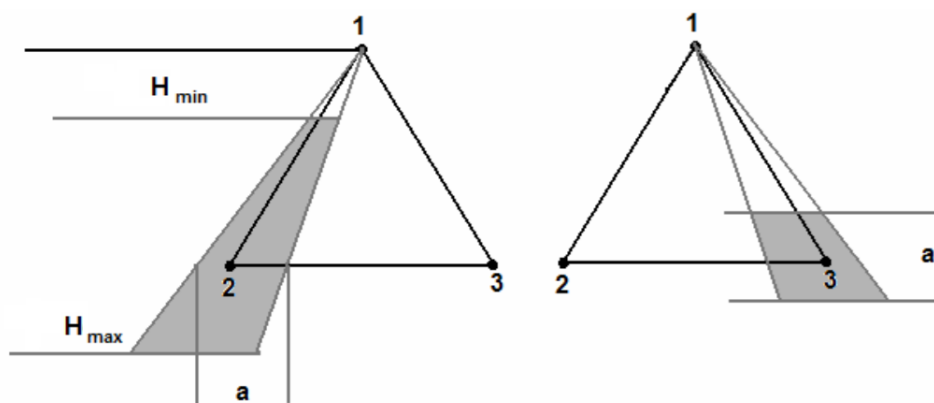


Рисунок 2.1 — Визначення трикутних знаків

Аналіз лінійних сегментів — гістограма ліній використовується для оцінки цілісності сегментів. Якщо довжина лінії (L), обчислена за рівнянням (3.3), перевищує локальний максимум (L'), сегмент вважається перерваним. Для ідентифікації лінії враховується центр обертання ((C_x, C_y)), розташований у центрі описаного прямокутника $Rect1$. Після обертання компоненти на кут (ϕ), точка (X_1, Y_1) трансформується в

де (D) — відстань до лінії.

Рівняння лінії визначається за рівнянням (2.1).

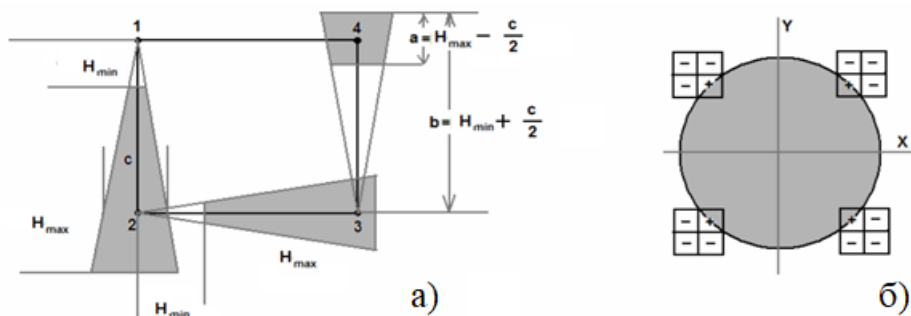


Рисунок 2.2 — Визначення геометрії знаку

Для підвищення точності аналізу форми сучасні системи можуть інтегрувати методи машинного навчання, такі як згорткові нейронні мережі, які автоматично вилучають складні геометричні патерни, або алгоритми на основі перетворення Хафа для точного визначення контурів. Ці методи забезпечують стійкість до шумів, перспективних спотворень і часткового перекриття, що є критично важливим для надійного розпізнавання дорожніх знаків у реальних умовах.

Аналіз форми дорожніх знаків за допомогою перетворення Хафа

Аналіз форми дорожніх знаків є критично важливим етапом у системах комп'ютерного зору, спрямованих на розпізнавання стандартизованих геометричних об'єктів, таких як трикутники, кола чи прямокутники, що використовуються в дорожніх знаках. Одним із ефективних методів для цього є перетворення Хафа (Hough Transform), представлене в [10], яке дозволяє виявляти лінії, криві та інші геометричні структури на зображенні. Цей метод забезпечує надійне виявлення форм навіть в умовах часткового перекриття чи шумів, що робить його цінним інструментом для систем автоматизованої допомоги водію (ADAS) і автономних транспортних засобів.

Перетворення Хафа трансформує точки зображення з декартової системи

к
о
о
р

де ρ — відстань від початку координат до прямої,

θ — кут нахилу нормалі до прямої.

На рисунку 2.2а зображено лінію та її параметри в площині $((x, y))$. У процесі перетворення кожна лінія в декартовій площині відповідає точці в параметричному просторі (ρ, θ) , що дозволяє ефективно виявляти лінійні структури. Цей підхід уможлиблює ідентифікацію геометричних форм, таких як сторони трикутних чи прямокутних знаків, шляхом аналізу накопичених голосів у просторі Хафа.

У дослідженні [9] перед застосуванням перетворення Хафа до зображення застосовується алгоритм витончення (thinning algorithm), який виконується після етапу колірної сегментації. Цей алгоритм зменшує товщину контурів до одного пікселя, що полегшує подальший аналіз. Витончені контури передаються на вхід перетворення Хафа, яке виявляє лінійні сегменти, що відповідають сторонам трикутного знака. Перетворення Хафа є універсальним інструментом, оскільки воно дозволяє виявляти не лише прямі лінії, а й інші криві, геометрія яких може бути параметризована, наприклад, кола чи еліпси, що є важливим для розпізнавання кругових дорожніх знаків. Для кругових знаків параметризація включає центр кола (a, b) і радіус (r) , що описується рівнянням:

Ця властивість робить метод придатним для аналізу різноманітних форм дорожніх знаків.

Для підвищення ефективності аналізу форми за допомогою перетворення Хафа сучасні системи можуть комбінувати цей метод із попередньою обробкою, такою як фільтрація шумів або нормалізація освітлення, щоб зменшити вплив зовнішніх факторів, таких як тіні чи погодні умови. Крім того, інтеграція

перетворення Хафа з методами машинного навчання, наприклад, згортковими нейронними мережами, дозволяє автоматизувати вилучення складних геометричних патернів і підвищувати стійкість до спотворень. Таким чином, перетворення Хафа залишається важливим інструментом у системах розпізнавання дорожніх знаків, забезпечуючи надійне виявлення форм і сприяючи безпеці дорожнього руху.

Перетворення Хафа є надійним інструментом для аналізу форми дорожніх знаків, забезпечуючи стійкість до спотворень і шумів. Інтеграція з сучасними методами глибокого навчання може додатково підвищити точність і ефективність, сприяючи безпеці дорожнього руху.

Для підвищення ефективності кругового перетворення Хафа сучасні системи можуть комбінувати його з іншими методами комп'ютерного зору, такими як попередня сегментація за кольором для звуження областей пошуку або інтеграція із згортковими нейронними мережами для автоматичного вилучення складних геометричних патернів. Крім того, оптимізація обчислень, наприклад, за допомогою адаптивного вибору радіусів чи використання графічних процесорів (GPU), сприяє застосуванню методу в реальному часі. Таким чином, кругове перетворення Хафа залишається потужним інструментом для аналізу кругових форм дорожніх знаків, забезпечуючи надійність і точність у системах розпізнавання.

2.6 Аналіз непрямо отриманих ознак у системах розпізнавання дорожніх знаків

Вибір характерних ознак є визначальним етапом у будь-якій системі розпізнавання, оскільки від нього значною мірою залежить успіх усієї системи. У сучасних системах розпізнавання дорожніх знаків (Traffic Sign Recognition, TSR) техніки аналізу ознак застосовуються не лише на етапі класифікації, а й на етапі виявлення, що відображає сучасні тенденції в галузі комп'ютерного зору та

штучного інтелекту. Окрім безпосередньо видимих характеристик, таких як колір і форма, для ідентифікації дорожніх знаків на зображеннях дедалі частіше використовуються непрямо отримані ознаки, які дозволяють підвищити точність і стійкість системи до складних умов, таких як зміни освітлення, часткове перекриття чи шум.

У дослідженні для виявлення дорожніх знаків застосовуються ознаки типу Хаара (Haar-like features), які є ефективним інструментом у задачах комп'ютерного зору. Ці ознаки формуються шляхом віднімання суми інтенсивностей пікселів у двох різних прямокутних регіонах, виділених на підобласті зображення в градаціях сірого. Приклади вибору таких прямокутних регіонів зображено на рисунку 2.1а. Для зменшення обчислювальної складності при розрахунку ознак Хаара використовуються інтегральні зображення, які дозволяють швидко обчислювати суми пікселів у довільних регіонах. Для заданого зображення I , інтегральне зображення $S(x, y)$ визначається як:

Ця формула, описує суму інтенсивностей усіх пікселів, розташованих лівіше та вище точки (x, y) . Використання інтегрального зображення значно прискорює обчислення, що є критично важливим для систем реального часу. =

Сума інтенсивностей пікселів у регіоні R , позначена як $F(R)$, може бути легко обчислена за допомогою значень інтегрального зображення у чотирьох ключових точках (A, B, C, D) за формулою:

де (A, B, C, D) — вершини прямокутного регіону. =

Цей підхід дозволяє ефективно обчислювати ознаки Хаара, що робить їх придатними для швидкого виявлення дорожніх знаків, особливо в умовах обмежених обчислювальних ресурсів.

Для підвищення точності аналізу непрямо отриманих ознак сучасні системи можуть комбінувати ознаки Хаара з іншими методами, такими як дескриптори локальних особливостей (SIFT, SURF) або згорткові нейронні мережі (CNN). Наприклад, ознаки Хаара можуть використовуватися для первинного виявлення кандидатів на знаки, після чого CNN уточнює класифікацію, враховуючи складніші патерни, такі як текстури чи просторові зв'язки. Крім того, для забезпечення стійкості до змін освітлення чи часткового перекриття можуть застосовуватися методи попередньої обробки, такі як нормалізація яскравості чи фільтрація шумів. Таким чином, аналіз непрямо отриманих ознак, зокрема ознак Хаара, відіграє ключову роль у забезпеченні надійності та точності систем розпізнавання дорожніх знаків, сприяючи безпеці дорожнього руху та ефективності автоматизованих систем керування.

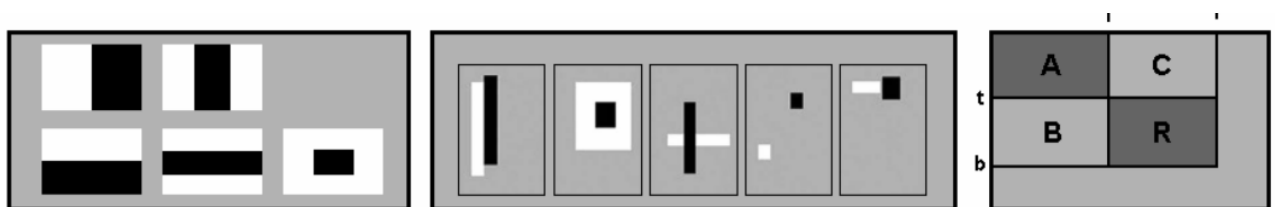
2.7 Формування класифікаторів на основі обчислених ознак для розпізнавання дорожніх знаків

Використання обчислених ознак є ключовим етапом у системах розпізнавання дорожніх знаків, оскільки дозволяє створювати класифікатори для визначення, чи є обрана область зображення дорожнім знаком. У сучасних системах комп'ютерного зору для цього застосовуються слабкі класифікатори, які базуються на раніше описаних ознаках типу Хаара (Haar-like features). Через простоту цих ознак кожен окремий слабкий класифікатор демонструє обмежену ефективність у задачі класифікації. Однак, шляхом об'єднання кількох ефективних слабких класифікаторів за допомогою алгоритму AdaBoost можна сформувати один потужний класифікатор, який значно підвищує точність розпізнавання. У межах цього підходу на кожній ітерації алгоритму AdaBoost обирається найкращий слабкий класифікатор із набору та додається до сильного класифікатора з відповідною вагою, яка залежить від його продуктивності.

У дослідженні [14] застосовується каскадна структура слабких класифікаторів, що є ефективним підходом для обробки зображень у реальному часі. У такій структурі кожне підвікно зображення послідовно проходить через

серію слабких класифікаторів. На кожному етапі, якщо підвікно визначається як потенційний об'єкт (дорожній знак), воно передається до наступного класифікатора. У разі відхилення на будь-якому етапі підвікно виключається з подальшої обробки. Важливою умовою є те, що кожен слабкий класифікатор повинен мати дуже низький рівень помилкових негативів (false negative rate), тоді як рівень помилкових позитивів (false positive rate) може бути вищим, оскільки подальші класифікатори уточнюють результати. Такий підхід забезпечує швидке відсіювання непотрібних регіонів, що є критично важливим для систем реального часу, таких як автоматизовані системи допомоги водію (ADAS).

Крім того, у [14] для класифікації підрегіонів зображення на предмет наявності дорожнього знака використовуються так звані дисоційовані диполі. Ці диполі, як показано на рисунку 2.3, складаються з прямокутних регіонів, розташованих більш вільно порівняно з ознаками Хаара, що дозволяє виловити ширший спектр ознак із вхідного підвікна. На відміну від ознак Хаара, де обчислюється різниця сум інтенсивностей пікселів, для дисоційованих диполів ознаки розраховуються як різниця середніх значень інтенсивностей пікселів у двох регіонах: інгібіторному (чорному) та збуджувальному (білому), як зображено на рис. 2.3. Такий підхід робить дисоційовані диполі нечутливими до масштабування вибраних прямокутників, що підвищує їхню універсальність. У дослідженні Лінхарта і Майдта зазначають, що точність детектора зростає зі збільшенням кількості доступних ознак, оскільки більший набір ознак дозволяє краще описувати складні візуальні патерни.



а) для ознак Хаара б) для дисоційованих диполів в) інтегрального зображення

Рисунок 2.3 — Варіанти вибору прямокутних регіонів

Каскадні класифікатори на основі ознак Хаара та дисоційованих диполів забезпечують швидке й надійне розпізнавання дорожніх знаків.

Ідентифікація дорожніх знаків у системах розпізнавання

Етап ідентифікації є завершальним у процесі розпізнавання дорожніх знаків, спрямованим на класифікацію виявлених об'єктів за їхньою належністю до певних категорій знаків. Цей етап відіграє ключову роль у забезпеченні точності систем розпізнавання, які застосовуються в автоматизованих системах допомоги водію та автономних транспортних засобах. У сучасних системах комп'ютерного зору для ідентифікації використовуються нейронні мережі, які дозволяють обробляти складні візуальні дані та забезпечувати стійкість до змін освітлення, ракурсу чи часткового перекриття.

У дослідженні [15] для класифікації виявлених об'єктів як одного з шести типів дорожніх знаків застосовуються багат шарові перцептрони (MLP). Модуль класифікації складається з шести окремих нейронних мереж, кожна з яких відповідає за ідентифікацію одного типу знака. Кожна мережа має два вихідні нейрони, які визначають, чи належить виявлений об'єкт до відповідного класу. Така модульна структура забезпечує гнучкість і можливість розширення системи для включення нових категорій знаків у майбутньому. Для обробки зображень використовується колірний простір YCbCr, який дозволяє ефективно виділяти колірні характеристики знаків. Під час навчання мережі застосовуються зображення розміром 30×30 пікселів, що забезпечує баланс між обчислювальною складністю та якістю класифікації. Попередня обробка зображень включає нормалізацію та корекцію контрастності для підвищення стійкості до змін освітлення.

У розділі 1 дорожні знаки класифікуються за дев'ятьма категоріями: наказові знаки (сині круги), заборонні знаки (червоні круги), попереджувальні знаки (червоні трикутники), інформаційні знаки (сині квадрати), знаки заборони паркування та зупинки (червоні та сині круги), знаки «Стоп» і «В'їзд заборонено» (червоні заповнені круги), знаки «Роботи в процесі» (червоні та жовті трикутники),

знаки пріоритету (жовті ромби), знаки «Дати дорогу» (червоні перевернуті трикутники).

Оскільки останні дві категорії містять лише по одному знаку, їхня ідентифікація здійснюється безпосередньо на основі результатів етапу виявлення, які враховують форму та колір, без додаткового етапу класифікації.

Перед подачею виявленого підрегіону зображення до нейронної мережі виконується кілька етапів попередньої обробки: видаляються межі знаків із фіксованою облямівкою (наприклад, червона облямівка трикутників), зображення конвертується в градації сірого, змінюється розмір до 50×50 пікселів, а також застосовується розтягнення контрасту для підвищення якості зображення. Ці кроки забезпечують уніфікацію вхідних даних і підвищують точність класифікації, особливо в умовах змін освітлення чи часткового перекриття. Для забезпечення стійкості системи до зовнішніх факторів, таких як погодні умови чи фізичні пошкодження знаків, можуть застосовуватися додаткові методи обробки, наприклад, фільтрація шумів або нормалізація яскравості. Таким чином, використання нейронних мереж у поєднанні з ретельною попередньою обробкою даних забезпечує надійну ідентифікацію дорожніх знаків, сприяючи безпеці дорожнього руху та ефективності автоматизованих систем керування. Підготовка даних і вибір методів ідентифікації є критично важливими для забезпечення точності та надійності систем розпізнавання дорожніх знаків, які відіграють ключову роль у підвищенні безпеки дорожнього руху та функціонуванні автоматизованих систем допомоги водію (ADAS). На рис. 2.4 представлено приклади варіацій, поворотів і зміщень моделей дорожніх знаків, включених до навчального набору даних. Оригінальна модель зображена у верхній частині рисунка [16]. Такі варіації, як повороти, зміщення та масштабування, додаються до навчального набору для підвищення стійкості системи до реальних умов, де дорожні знаки можуть бути частково перекриті, змінені за ракурсом або зазнати впливу зовнішніх факторів, таких як освітлення чи погодні умови. Використання синтетичних зображень із різними трансформаціями дозволяє нейронним мережам

або іншим алгоритмам класифікації ефективніше узагальнювати візуальні патерни, що є важливим для роботи в динамічних дорожніх умовах.

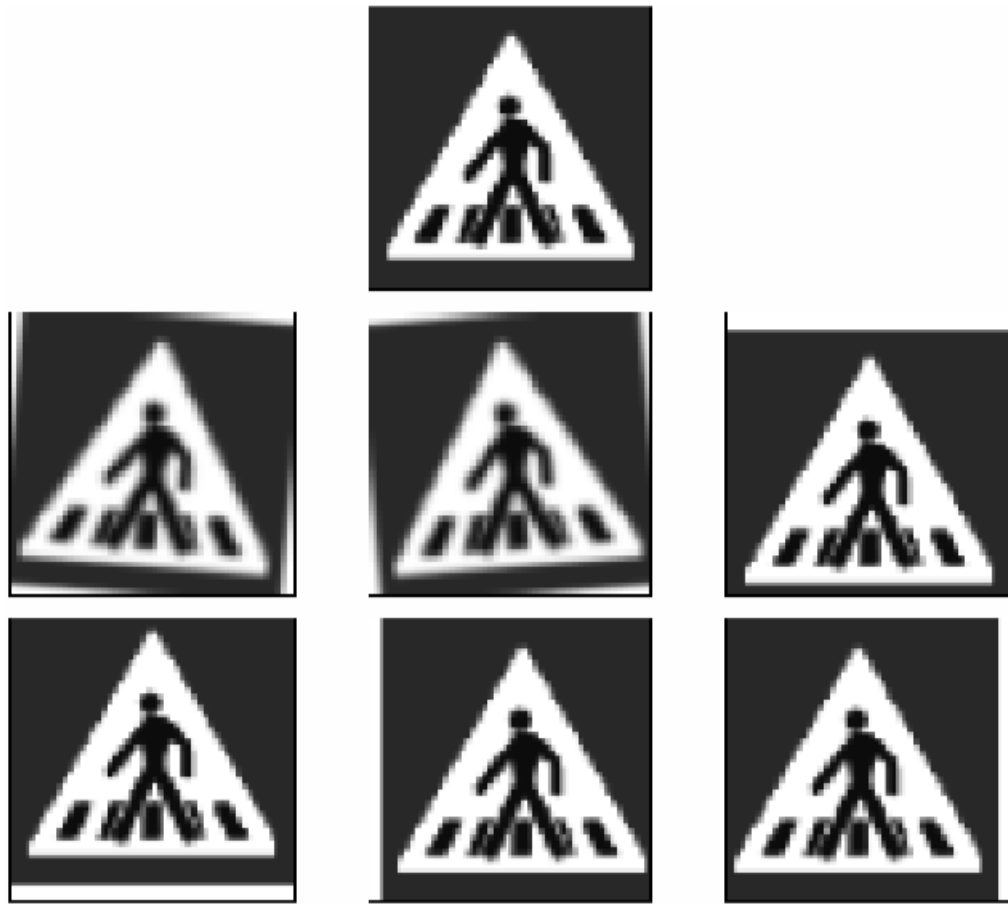


Рисунок 2.4 — Приклади варіацій, поворотів і зміщень моделей дорожніх знаків

У дослідженні [19] для ідентифікації дорожніх знаків використовується метод порівняння з шаблонами, який базується на трансформації відстані (distance transform) для вилучення ознак із виявлених зображень знаків. Трансформація відстані є потужним інструментом у комп'ютерному зорі, який дозволяє оцінити геометричні характеристики об'єктів шляхом обчислення відстані від кожного пікселя до найближчого контурного пікселя на бінарному зображенні. Для пікселя на зображенні R трансформація відстані визначається як:

де B_R — множина контурних пікселів,

— евклідова відстань між пікселями p і q .

Ця формула дозволяє кількісно оцінити близькість кожного пікселя до межі об'єкта, що є корисним для порівняння з еталонними шаблонами.

Для підвищення стійкості методу до змін масштабу та інших спотворень у запропоновано модифіковану версію трансформації відстані:

Ця модифікація враховує експоненціальне зменшення впливу відстані, що робить метод менш чутливим до незначних змін у геометричних характеристиках знака. Такий підхід підвищує точність порівняння з шаблонами, особливо в умовах, коли зображення може бути частково деформованим через перспективу чи шум.

Для забезпечення високої ефективності методу порівняння з шаблонами додатково застосовуються техніки попередньої обробки, такі як нормалізація яскравості, фільтрація шумів і корекція контрастності, що дозволяють мінімізувати вплив зовнішніх факторів.

Оскільки інформаційна частина дорожнього знака, як правило, зосереджена в його центрі, для підвищення точності ідентифікації до кожного пікселя трансформації відстані застосовуються різні вагові коефіцієнти залежно від його відстані до центру регіону. Ці ваги обчислюються за формулою:

де z — радіус описаного кола регіону R ,

— відстань між i -м пікселем і центром регіону.

Такий підхід дозволяє надавати більшу вагу пікселям, розташованим ближче до центру знака, що підвищує точність аналізу його інформаційної частини.

Для оцінки подібності між кандидатом на знак і еталонними зображеннями використовується метрика, визначена формулою:

де H — зображення кандидата,

— i -й елемент навчального набору,

H_F — векторне представлення зваженої трансформації відстані для зображення кандидата H ,

μ_i — середнє значення векторного представлення зваженої трансформації відстані для навчальних зображень i -го класу, =

σ_i — відповідна дисперсія. —

Клас i , який забезпечує мінімальне значення $S(H, R_i)$, визначається як тип кандидата H . Цей підхід дозволяє ефективно порівнювати зображення кандидата з еталонними шаблонами, враховуючи геометричні та просторові характеристики.

Застосування зваженої трансформації відстані забезпечує стійкість до незначних спотворень, таких як повороти чи зміни масштабу, що є типовими для реальних дорожніх умов. Для додаткового підвищення точності можуть використовуватися методи попередньої обробки, такі як нормалізація яскравості чи фільтрація шумів, які мінімізують вплив зовнішніх факторів, наприклад, тіней чи погодних умов.

Методи верифікації та ідентифікації дорожніх знаків

Теорія розпізнавання образів є важливою галуззю інформатики, що об'єднує принципи та методи класифікації й ідентифікації об'єктів, процесів, сигналів і явищ, які описуються скінченним набором характеристик і ознак. Ця дисципліна відіграє ключову роль у розробці систем штучного інтелекту, зокрема в задачах комп'ютерного зору.

Задачі розпізнавання об'єктів на зображеннях виникають у різноманітних сферах сучасної науки та техніки, характеризуючись різними рівнями складності й вимогами до точності. Для їх вирішення застосовуються численні алгоритми, серед яких варто виокремити методи порівняння з шаблонами, використання

дескрипторів локальних особливостей, алгоритм Віолі-Джонса, а також моделі на основі нейронних мереж, зокрема згорткових (Convolutional Neural Networks,

Сучасна теорія розпізнавання образів активно розвивається, однак низка викликів ускладнює досягнення високої ефективності алгоритмів. До таких проблем належать: низька роздільна здатність зображень, що знижує якість вхідних даних, складний фон, який характеризується неоднорідністю кольорів і текстур, варіації ракурсів об'єктів, що спричиняють геометричні та фотометричні спотворення, наявність множини об'єктів на зображенні, кількість яких заздалегідь невідома.

Для подолання цих обмежень необхідні додаткові методи попередньої обробки зображень, що підвищують обчислювальну складність систем. Таким чином, сучасні дослідження зосереджені на розробці алгоритмів, які мінімізують вплив зазначених факторів і забезпечують стабільну точність розпізнавання.

Процес розпізнавання дорожніх знаків логічно поділяється на два основні етапи. Виявлення області розташування дорожнього знака на зображенні. Класифікація та ідентифікація виявленого знака.

На етапі розпізнавання зображення порівнюється з попередньо збереженими шаблонами. У цьому контексті розрізняють два основні методи.

Ідентифікація — порівняння вхідного зображення з множиною шаблонів у базі даних (схема «один до багатьох»). Цей метод відповідає на запитання «що це за об'єкт?» і характеризується вищою обчислювальною складністю та ймовірністю помилок.

Верифікація — порівняння зображення з одним конкретним шаблоном (схема «один до одного»). Цей підхід відповідає на запитання «чи це той самий об'єкт?» і є швидшим та точнішим, але потребує додаткового ідентифікатора, наприклад, унікального коду.

Детерміновані ознаки для розпізнавання дорожніх знаків — це чітко визначені, вимірювані характеристики зображень, які використовуються для ідентифікації та класифікації знаків у системах комп'ютерного зору. Вони

базуються на фіксованих властивостях, що не залежать від випадкових факторів, і зазвичай застосовуються в алгоритмах, таких як порівняння з шаблонами, дескриптори локальних особливостей (SIFT, SURF) або алгоритм Віоли-Джонса. Нижче наведено основні детерміновані ознаки, які використовуються для розпізнавання дорожніх знаків, з акцентом на термінологію штучного інтелекту та алгоритми, як зазначено у вашому запиті.

Геометричні ознаки. Форма знака — дорожні знаки мають стандартизовані форми, які є детермінованими: круглі (наприклад, заборонні чи наказові знаки); трикутні (попереджувальні знаки, часто з червоною облямівкою); квадратні або прямокутні (інформаційні чи вказівні знаки); алгоритми, такі як *hough transform*, використовуються для виявлення цих форм шляхом аналізу контурів.

Розмір і пропорції — співвідношення сторін (наприклад, 1:1 для квадратних знаків) або діаметр кола є фіксованими характеристиками, що полегшують класифікацію.

Кути та орієнтація — кути між сторонами (наприклад, 60° у трикутних знаках) або орієнтація знака відносно горизонту.

Дорожні знаки мають стандартизовану кольорову палітру, визначену міжнародними нормами (наприклад, Віденською конвенцією про дорожні знаки):

- червоний (заборонні або попереджувальні знаки);
- синій (наказові або інформаційні знаки);
- білий, зелений або жовтий (для фону інформаційних знаків).

Аналіз гістограм кольорів у просторах RGB, HSV або Lab дозволяє точно виділяти ці ознаки. Наприклад, червона облямівка трикутних знаків є чіткою детермінованою ознакою, яку можна виявити за допомогою сегментації кольорів.

Текстові та символічні ознаки, шрифт і розмір тексту — дорожні знаки містять стандартизовані написи (наприклад, «STOP», цифри для обмеження швидкості). Алгоритми оптичного розпізнавання символів (OCR) використовуються для аналізу тексту. Піктограми та символи — унікальні графічні елементи (стрілки, зображення пішоходів, велосипедів).

Детерміновані ознаки, такі як геометричні форми, кольорові палітри, текстові елементи, градієнти та прямокутні регіони, є основою для ефективного розпізнавання дорожніх знаків. Вони забезпечують стійкість алгоритмів до змін освітлення, ракурсу чи фонових шумів, що є критично важливим для систем автономного водіння та ADAS.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ КОМП'ЮТЕРНОЇ СИСТЕМИ РОЗПІЗНАВАННЯ ДОРОЖНІХ ЗНАКІВ

.1 Визначення сучасних програмних інструментів для розробки системи обробки зображень

У контексті стрімкого розвитку цифрових технологій та зростання обсягів візуальних даних, обробка зображень посідає одне з центральних місць у багатьох наукових, промислових та комерційних застосунках. Від автоматизованого контролю якості на виробництві та медичної діагностики до систем безпеки та автономних систем, ефективність і надійність обробки зображень є критично важливими. Реалізація таких систем вимагає використання не лише досконалого розуміння теоретичних засад алгоритмів, а й глибокого знання сучасних програмних інструментів та технологій, що забезпечують їхнє функціональне та продуктивне втілення.

Вибір відповідного інструментального та програмного забезпечення визначає архітектуру системи, її розробку, швидкість виконання, здатність до масштабування та подальшої підтримки. Цей розділ магістерської роботи зосереджений на описі сучасних програмних мов, бібліотек, фреймворків та інструментів, що застосовуються для розробки систем обробки зображень. Буде представлено опис їхніх основних характеристик та ролі у процесі розробки, а також розглянуто інтегровані середовища розробки та аспекти розгортання програмних рішень.

Програмні мови слугують фундаментом для побудови будь-якого програмного забезпечення, і їхній вибір для завдань обробки зображень залежить від вимог до продуктивності, швидкості розробки, доступності бібліотек та цільової платформи.

3.1.1 Аналіз можливостей мов програмування

Python є інтерпретованою, високорівневою мовою програмування загального призначення. Вона підтримує множинні парадигми програмування, включаючи об'єктно-орієнтоване, імперативне та функціональне програмування. У сфері

наукових обчислень, зокрема обробки зображень та машинного навчання, Python сформував потужну екосистему. Це досягається завдяки її синтаксичній простоті, що сприяє швидкому написанню коду, та широкому доступу до численних спеціалізованих бібліотек, багато з яких реалізовані на нижчому рівні (наприклад, C/C++), що забезпечує адекватну продуктивність для більшості завдань. Python використовується для швидкого прототипування алгоритмів, розробки дослідницьких інструментів та як бекенд для веб-додатків.

C++ є компільованою мовою програмування загального призначення з сильним статичним типом. Вона надає низькорівневий доступ до пам'яті та апаратних ресурсів, що робить її вибором для систем, де критичними є висока продуктивність, мінімальна затримка та ефективне використання ресурсів. У галузі комп'ютерного зору C++ традиційно використовується для розробки високооптимізованих бібліотек, вбудованих систем, додатків реального часу та програмного забезпечення для пристроїв з обмеженими обчислювальними можливостями. Більшість фундаментальних алгоритмів обробки зображень та комп'ютерного зору спочатку розроблялися та оптимізувалися саме на C++ (рис. 3.1).



Рисунок 3.1 — Логотип мови C++

Julia є динамічною мовою програмування високого рівня, розробленою для високоефективних числових та наукових обчислень. Її архітектура націлена на подолання так званої "проблеми двох мов", коли розробники використовують одну мову для зручності прототипування (наприклад, Python) та іншу для досягнення продуктивності (наприклад, C++). Julia досягає високої продуктивності завдяки компіляції Just-In-Time (JIT), яка оптимізує код під час виконання. Вона інтегрує

функції, які зазвичай зустрічаються в окремих середовищах для наукових обчислень, таких як синтаксис, орієнтований на математичні операції, та вбудовану підтримку паралельних обчислень.

MATLAB є пропрієтарним програмним середовищем та мовою програмування, спеціалізованою для числових обчислень, візуалізації даних та алгоритмічного моделювання. Він широко використовується в інженерних, наукових та дослідницьких дисциплінах, зокрема в обробці сигналів та зображень. MATLAB надає великий набір вбудованих функцій та додаткових "тулбоксів", які містять оптимізовані алгоритми для специфічних завдань. Octave є вільною та відкритою альтернативою MATLAB, що пропонує схожий синтаксис та функціональність. Ці середовища зручні для швидкого дослідження алгоритмів, візуалізації проміжних результатів та верифікації концепцій.

Java є об'єктно-орієнтованою мовою програмування загального призначення, відомою своєю платформовою незалежністю завдяки віртуальній машині Java (JVM). Kotlin – це сучасна статично типізована мова програмування, яка працює на JVM і є повністю сумісною з Java. Обидві мови використовуються для розробки широкого спектру програм, включаючи десктопні, мобільні (Android) та корпоративні веб-додатки. У контексті обробки зображень, Java та Kotlin можуть застосовуватися для створення крос-платформених інструментів, що вимагають надійності та масштабованості, а також для бекенд-систем, які обробляють великі обсяги візуальних даних.

2 Спеціалізовані бібліотеки та фреймворки для обробки зображень

Сучасна розробка програмного забезпечення для обробки зображень спирається на використання спеціалізованих бібліотек та фреймворків, які надають реалізації складних алгоритмів та інструменти для ефективної роботи з візуальними даними.

OpenCV (Open Source Computer Vision Library). OpenCV є крос-платформною бібліотекою комп'ютерного зору з відкритим вихідним кодом, що надає понад 2500 оптимізованих алгоритмів. Вона охоплює широкий спектр функцій, від базових

операцій над зображеннями та відео до складних завдань машинного навчання та глибокого навчання. Бібліотека розроблена переважно на C++, але має інтерфейси для Python, Java, MATLAB та інших мов, що робить її універсальним інструментом для різноманітних проектів (рис. 3.2).

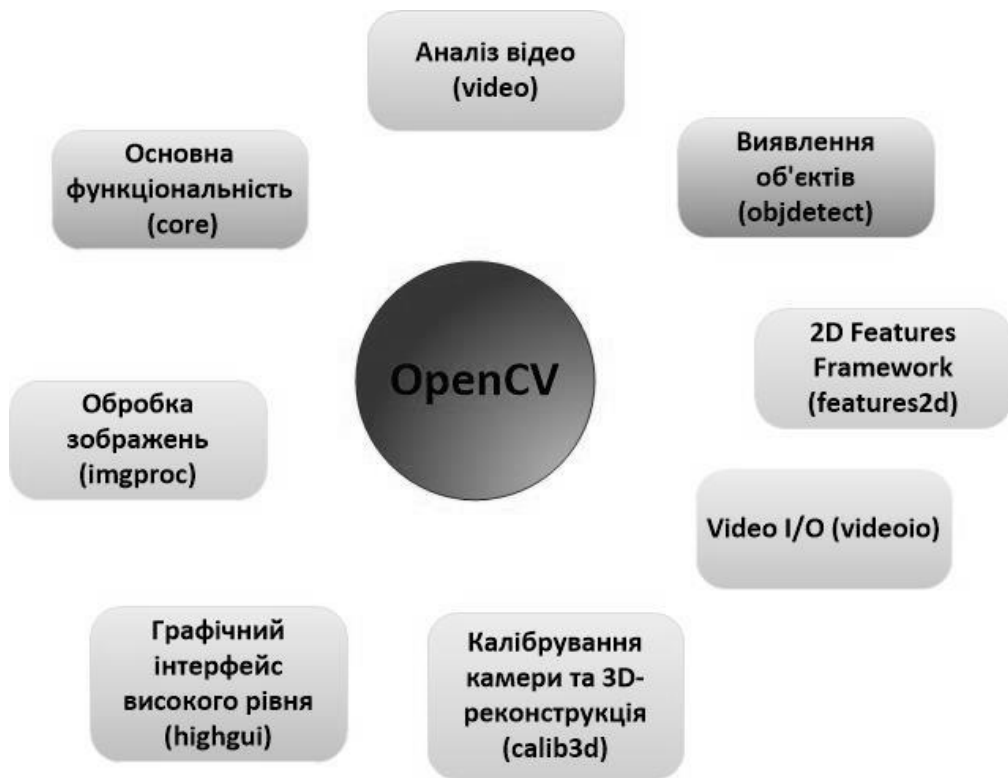


Рисунок 3.2 — Модульна структура бібліотеки OpenCV

Функціонал OpenCV включає операції введення/виведення зображень, перетворення колірних просторів, геометричні перетворення, фільтрацію (включаючи Гауссів розмив, медіанний та білатеральний фільтри), виявлення ознак, сегментацію об'єктів, розпізнавання образів та багато іншого.

NumPy (Numerical Python) є фундаментальним пакетом для наукових обчислень у Python. Він надає потужний об'єкт багатовимірного масиву (ndarray), який є стандартним форматом для представлення зображень та інших числових даних в екосистемі Python. Операції над масивами в NumPy оптимізовані та ефективні, оскільки багато з них реалізовані на C або Fortran.

— це бібліотека, що ґрунтується на NumPy і надає додаткові модулі для широкого спектра наукових та технічних обчислень. Модуль `scipy.ndimage` зокрема, містить

функції для обробки багатовимірних зображень, включаючи різні фільтри, морфологічні операції та перетворення.

Scikit-image є бібліотекою для обробки зображень, реалізованою на Python з використанням NumPy. Вона надає набір алгоритмів для вирішення різноманітних завдань, таких як сегментація зображень, геометричні перетворення, виявлення та опис ознак, морфологічні операції та різні типи фільтрації, scikit-image розроблена з акцентом на чистоту коду та зручність використання, що робить її популярною серед дослідників та розробників.

Pillow (PIL Fork, для Python) є сучасною та активною версією Python Imaging Library (PIL), що забезпечує основні функціональні можливості для роботи з растровими зображеннями. Ця бібліотека використовується для базових операцій, таких як відкриття та збереження зображень у різних форматах, зміна їхнього розміру, обертання, обрізка, зміна кольірних режимів та застосування простих фільтрів. Pillow часто використовується для початкової обробки зображень перед їхнім подальшим аналізом за допомогою більш спеціалізованих бібліотек.

Фреймворки глибокого навчання

Сучасна обробка зображень значною мірою інтегрована з методами глибокого навчання. Фреймворки, такі як TensorFlow (розроблений Google), PyTorch (розроблений Facebook) та Keras (високорівневий API, що працює поверх `ow` або PyTorch), є ключовими інструментами для розробки та розгортання моделей глибокого навчання, зокрема згорткових нейронних мереж (CNNs). Вони надають інструменти для: побудови, тренування та оцінки складних нейронних мереж для класифікації зображень, сегментації, виявлення об'єктів та генерації зображень; ефективних обчислень на графічних процесорах (GPU) завдяки підтримці OpenCL; автоматичного диференціювання, що спрощує оптимізацію моделей. Ці фреймворки часто включають власні утиліти для завантаження, попередньої обробки та аугментації зображень, інтегруючи їх у пайплайни машинного навчання.

CUDA (Compute Unified Device Architecture) від NVIDIA та OpenCL (Open Computing Language) є паралельними обчислювальними платформами та програмними інтерфейсами, що дозволяють розробникам використовувати потужність графічних процесорів (GPU) для виконання загальних обчислювальних завдань. У контексті обробки зображень, GPU-прискорення є критично важливим для завдань, що вимагають високої обчислювальної інтенсивності, таких як обробка відео в реальному часі, застосування складних фільтрів до великих зображень або тренування глибоких нейронних мереж. Багато сучасних бібліотек обробки зображень (включаючи OpenCV та фреймворки глибокого навчання) мають інтегровану підтримку цих технологій, дозволяючи виконувати обчислення на GPU.

3.2 Вибір інтегрованого середовища розробки (IDE) та інструментів розробки

Ефективність процесу розробки значною мірою залежить від якості використовуваних інструментів розробки. Інтегровані середовища розробки (IDE) надають комплексний набір інструментів для написання, налагодження та тестування коду. Для Python популярними IDE є PyCharm (від JetBrains), що пропонує розширену підтримку для наукових обчислень та візуалізації, та Visual iо Code (від Microsoft), який є легким, гнучким редактором з потужними розширеннями для Python, C++ та інших мов. Для розробки на C++ часто використовують Visual Studio (для Windows) (рис. 3.3) або CLion (від JetBrains), що забезпечують потужні засоби налагодження та управління проектами.

Jupyter Notebooks (або JupyterLab) є інтерактивними веб-середовищами, що дозволяють поєднувати код, текст, рівняння та візуалізації в одному документі. Вони є ідеальними для дослідницьких завдань, швидкого прототипування алгоритмів обробки зображень, візуалізації проміжних результатів та представлення результатів експериментів.

Docker та Singularity (для високопродуктивних обчислень) є інструментами для контейнеризації, що дозволяють створювати ізольовані, портативні та відтворювані середовища для розробки та розгортання програмного забезпечення.

Це забезпечує стабільність залежностей та спрощує розгортання додатків на різних системах.

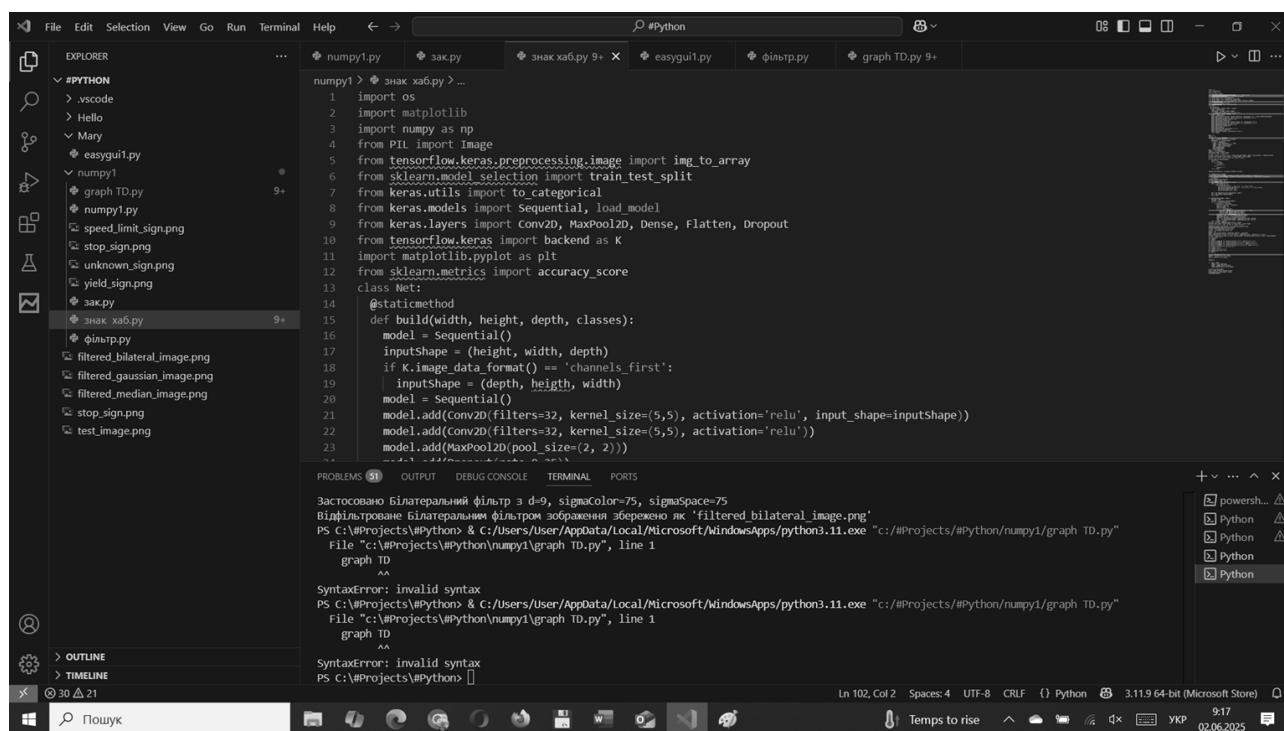


Рисунок 3.3 — Середовище розробки Visual Studio

Git є де-факто стандартом для систем контролю версій. Він дозволяє ефективно керувати змінами у коді, відстежувати історію проекту, співпрацювати з іншими розробниками та забезпечувати цілісність кодової бази.

Після розробки програмного забезпечення для обробки зображень постає питання його розгортання та забезпечення можливості масштабування для вирішення реальних завдань. Вибір стратегії розгортання залежить від цільового використання системи. Desktopні застосунки: для програм, які працюють локально на комп'ютері користувача, можуть використовуватися фреймворки графічного інтерфейсу, такі як PyQt (для Python), Tkinter (для Python) або ж крос-платформенні рішення на базі Electron (для веб-технологій). Зазвичай вимагають пакування всіх залежностей та бібліотек для розповсюдження.

Веб-Застосунки для доступу до функціоналу обробки зображень через веб-браузер розробляються веб-сервіси. Мови та фреймворки, такі як Flask або Django (для Python), Node.js (для JavaScript), Spring (для Java) можуть слугувати основою

для бекенду, який отримує зображення, обробляє їх і повертає результат. Обробка може відбуватися на сервері або в хмарному середовищі. Мобільні застосунки для розробки додатків для смартфонів та планшетів (Android, iOS) використовуються нативні мови (Kotlin/Java для Android, Swift/Objective-C для iOS) або крос-платформенні фреймворки (React Native, Flutter). OpenCV та інші бібліотеки часто мають мобільні SDK або можуть бути інтегровані через нативні інтерфейси.

Вбудовані системи та периферійні пристрої (Edge Devices) для застосунків, що працюють на пристроях з обмеженими ресурсами (наприклад, Raspberry Pi, NVIDIA Jetson, Intel Movidius), необхідна оптимізація коду, використання легких бібліотек та, можливо, апаратних прискорювачів. Такі системи часто використовуються для обробки даних безпосередньо "на краю мережі", зменшуючи затримку та залежність від хмарних сервісів.

Хмарні Обчислення для завдань, що вимагають обробки великих обсягів даних або забезпечення високої доступності та масштабованості, використовуються хмарні платформи, такі як Amazon Web Services (AWS), Microsoft Azure або Google Cloud Platform (GCP). Ці платформи надають обчислювальні ресурси, сховища даних та сервіси для розгортання мікросервісів, бессерверних функцій або контейнеризованих додатків обробки зображень.

Порівняльний аналіз інструментів та обґрунтування вибору python для проекту

Вибір інструментарію для розробки системи обробки зображень (див. табл. 3.1) є компромісом між різними факторами, такими як необхідна продуктивність, швидкість розробки, гнучкість, доступність ресурсів та спеціалізація проекту.

Бібліотеки та фреймворки також мають свої особливості. OpenCV є універсальним рішенням з високою продуктивністю та широким функціоналом, підходящим для багатьох мов. NumPy і SciPy є фундаментальними для всіх Python-проектів, що вимагають роботи з числовими даними, включаючи зображення. scikit-image пропонує більш "Pythonic" реалізації алгоритмів, доповнюючи OpenCV. Pillow є зручним для базових маніпуляцій з файлами зображень.

Фреймворки глибокого навчання, такі як TensorFlow і PyTorch, домінують у сучасних завданнях розпізнавання та аналізу зображень, що ґрунтуються на штучному інтелекті, надаючи потужні засоби для роботи з великими моделями та даними. CUDA/OpenCL є інструментами для оптимізації обчислень на GPU, що є ключовим для продуктивності в інтенсивних завданнях обробки зображень.

Таблиця 3.1 — Порівняльна характеристика засобів розробки

Характеристика / Мова					
Продуктивність	Добра (через оптимізовані C-бібліотеки)	Дуже висока (контроль низького рівня)	Дуже висока (JIT-компіляція)	Середня (для прототипів), висока (для вбудованих функцій)	Середня-висока
Швидкість розробки	Дуже висока (завдяки синтаксису та бібліотекам)	Низька (складність, ручне керування пам'яттю)	Висока (зручний синтаксис)	Дуже висока (спеціалізовані функції)	Середня (багатослівність)
Контроль ресурсів	Обмежений (через інтерпретатор)	Повний (прямий доступ до пам'яті)	Високий (через JIT-компіляцію та низькорівневі можливості)	Обмежений	Обмежений (через JVM)
Типове застосування	Прототипування, дослідження, веб-сервіси,	Системи реального часу, вбудовані системи, бібліотеки	Числові обчислення, наукові дослідження	Швидке прототипування, візуалізація даних	Десктопні/мобільні додатки, корпоративні системи

Для реалізації програми зчитування та фільтрації зображень, опис якої надано в даній магістерській роботі, було обрано мову програмування Python. Цей вибір ґрунтується на кількох ключових факторах, які роблять Python оптимальним рішенням для такого типу проекту.

Швидкість розробки та прототипування — Python дозволяє розробляти та тестувати алгоритми значно швидше порівняно з компільованими мовами, такими

як C++. Це особливо важливо на початкових етапах проекту, де потрібна гнучкість для експериментів з різними фільтрами та параметрами. Простий та лаконічний синтаксис Python мінімізує кількість написаного коду, що зменшує ймовірність помилок та прискорює ітерації розробки.

Багата екосистема бібліотек для обробки зображень та наукових обчислень Python має неперевершену екосистему спеціалізованих бібліотек, які безпосередньо стосуються завдань обробки зображень. Зокрема, OpenCV надає комплексний набір функцій для зчитування, фільтрації (Гаусів розмив, медіанний, білатеральний фільтр) та маніпуляцій із зображеннями. Python API для OpenCV є добре документованим та простим у використанні, дозволяючи доступ до високооптимізованих C++ реалізацій алгоритмів. NumPy — це фундаментальна бібліотека для ефективної роботи з багатовимірними масивами, які є стандартним представленням зображень у цифровому вигляді. Операції NumPy є високопродуктивними, що важливо для обробки великих зображень.

Достатня продуктивність для завдань зчитування та фільтрації окремих зображень (або невеликих потоків), продуктивності в Python (завдяки оптимізованим бекендам бібліотек, як OpenCV) є цілком достатньою. Хоча C++ може запропонувати вищу "чисту" продуктивність для дуже специфічних, ресурсомістких завдань, для загального випадку, що розглядається в даній роботі, Python забезпечує оптимальний баланс між швидкістю розробки та ефективністю виконання.

Враховуючи ці фактори, Python був визнаний найбільш підходящою мовою для розробки програми зчитування та фільтрації зображень, оскільки він дозволяє швидко та ефективно реалізувати необхідний функціонал, використовуючи широкий спектр перевірених та оптимізованих бібліотек.

3.4 Попередня обробка вхідного зображення

Розглянемо блок попередньої обробки зображення з програми розпізнавання дорожніх знаків (рис. 3.4), яка не використовує нейронні мережі (тобто, базується на OpenCV для аналізу кольору та форми).

Мета цього блоку — підготувати вхідне зображення таким чином, щоб полегшити подальше виділення ознак знака, таких як його колір та форма.

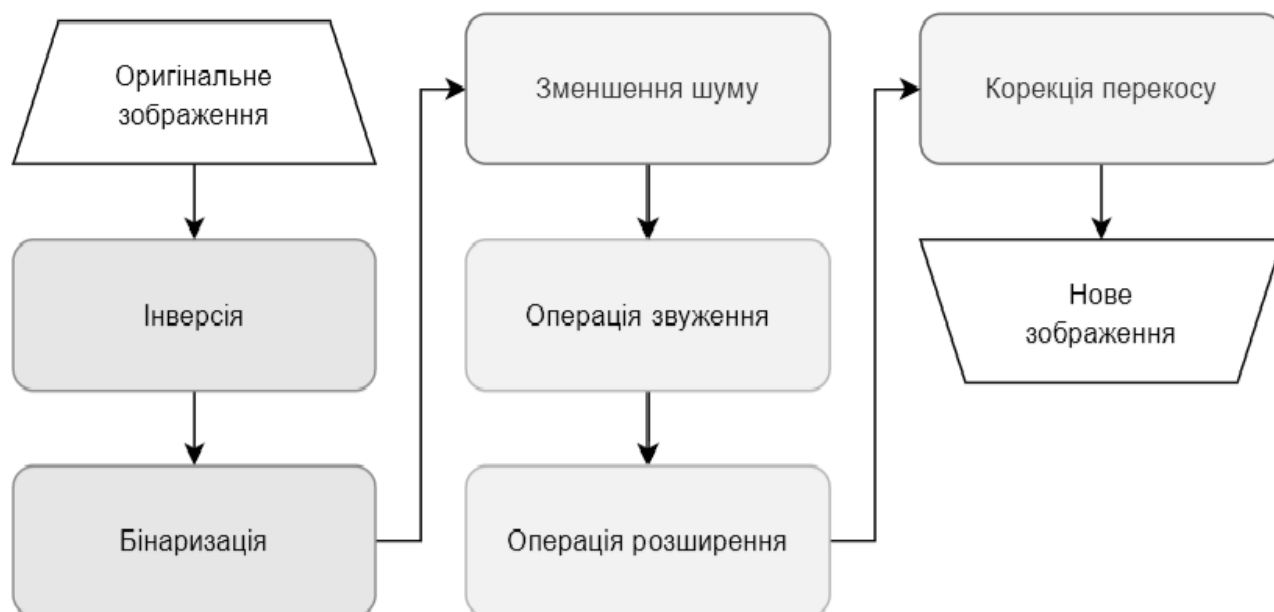


Рисунок 3.4 — Блок-схема попередньої обробки вхідного зображення

Функції, які зазвичай використовуються на цьому етапі, та їх детальний опис. Функція `cv2.imread(image_path)` — завантаження зображення з файлу. Функція читає зображення за вказаним шляхом `image_path` і повертає його у вигляді багатовимірної масиви `NumPy`. Якщо шлях неправильний або файл пошкоджений/не є зображенням, функція поверне `None`. Для кольорових зображень масив зазвичай має три канали (BGR — синій, зелений, червоний, у такому порядку в `OpenCV`).

Функція `cv2.resize(img, (processed_width, processed_height))` — зміна розміру зображення до фіксованих значень. Обробка зображень однакового розміру спрощує подальші кроки, оскільки параметри (наприклад, розмір ядра для розмиття, порогові значення) можуть бути більш універсальними. Функція приймає вхідне зображення `img` та кортеж `(processed_width, processed_height)`, що задає новий бажаний розмір. У нашому коді ми обчислювали `processed_height` пропорційно, щоб зберегти співвідношення сторін, змінюючи лише ширину до

Лістинг 3.1 — Попередня обробка зображення із фільтрацією

```
# Зчитуємо зображення

# Перевіряємо, чи успішно прочитано зображення

print(f"Помилка: Не вдалося прочитати зображення за шляхом

filtered_img = None # Ініціалізуємо змінну для відфільтрованого
зображення

# Застосовуємо Гауссів розмив
# kernel_size: розмір Гауссового ядра (ширина, висота), обидва мають
бути непарними та позитивними.
# 0: стандартне відхилення Гауссового ядра в напрямку X (0 означає,
що воно обчислюється з kernel_size).

print(f"Застосовано Гауссів розмив з розміром ядра {kernel_size}")

# Застосовуємо Медіанний фільтр (kernel_size має бути непарним цілим
числом, наприклад, 5)
# Цей фільтр добре усуває "сіль і перець" шум.

print(f"Застосовано Медіанний фільтр з розміром ядра {ksize}")

# Переконуємось, що розмір ядра непарний для медіанного фільтра
print(f"Застосовано Білатеральний фільтр з d={d},

print(f"Помилка: Невідомий тип фільтра '{filter_type}'. Підтримувані
типи: 'gaussian', 'median', 'bilateral'.")
```

Функція `cv2.cvtColor(img, cv2.COLOR_BGR2HSV)` — конвертація зображення з кольорового простору BGR у HSV. Кольоровий простір HSV (Hue — відтінок, Saturation — насиченість, Value — яскравість) часто є більш зручним для виділення об'єктів за кольором, ніж BGR. Hue (H) — представляє сам колір (червоний, зелений, синій тощо), це значення менш чутливе до змін освітлення порівняно з компонентами R, G, B. Saturation (S) показує "чистоту" або інтенсивність кольору. Висока насиченість означає яскравий, чистий колір, низька — відтінки сірого. Value (V) представляє яскравість кольору. Використання HSV дозволяє легше визначити діапазони для конкретних кольорів (наприклад, червоного для знака "STOP") незалежно від невеликих варіацій освітлення.

— конвертація зображення з кольорового простору BGR у відтінки сірого. Деякі операції, зокрема виявлення контурів або країв, краще працюють на одноканальних зображеннях у відтінках сірого. Це зображення містить лише інформацію про інтенсивність світла (яскравість) кожного пікселя, відкидаючи інформацію про колір.

Лістинг 3.2 — зчитування зображення

```
# Перевірка наявності файлу (важливо для уникнення помилок)
```

```
print(f"Помилка: Файл {image_path} не знайдено.")  
exit() # Завершуємо, якщо файл не існує
```

```
print(f"Помилка: Не вдалося завантажити зображення за шляхом
```

```
print("Перевірте шлях, цілісність файлу або формат.")
```

```
# Створення копії зображення, щоб зберегти оригінал незмінним
```

```
cv2.imshow("Оригінальне зображення", original_img)
```

Функція `cv2.GaussianBlur(gray, (5, 5), 0)` — розмиття зображення за Гауссом для зменшення шуму. Ця функція застосовує до зображення (у нашому випадку, до зображення у відтінках сірого `gray`) фільтр Гаусса. `(5, 5)` — це розмір ядра фільтра (матриці, яка "проходить" по зображенню). Більші значення ядра призводять до сильнішого розмиття. Розміри ядра повинні бути непарними. Стандартне відхилення Гаусса по осі `X` (та `Y`, якщо не вказано окремо). Якщо встановлено `0`, воно обчислюється автоматично на основі розміру ядра. Розмиття допомагає згладити дрібні деталі та шум на зображенні, що може покращити результати наступних кроків, таких як виявлення країв.

Після цих кроків попередньої обробки ми отримуємо кілька проміжних зображень: зображення з виділеними краями (`edges`) та колірну маску (`red_mask`), які потім використовуються для виявлення контурів самих дорожніх знаків.

Блок фільтрації зображень допомагає "очистити" зображення та підкреслити важливі ознаки, що є критично важливим для наступних етапів аналізу. Завантаження та базове дублювання зображення — перш ніж ми почнемо фільтрувати зображення, його потрібно завантажити та, можливо, створити копію, щоб не змінювати оригінальний файл. Функція `cv2.imread()` — читання зображення. Ця функція є точкою входу в світ обробки зображень. Вона завантажує зображення з вказаного шляху.

Перетворення колірного простору, тому що деякі операції краще виконувати в інших колірних просторах, ніж стандартний `BGR` (який використовує `OpenCV` за замовчуванням). `HSV` (`Hue`, `Saturation`, `Value`) є особливо корисним для виділення об'єктів за кольором, тоді як градації сірого є обов'язковими для багатьох алгоритмів виявлення країв.

Лістинг 3.3 — Нормалізація зображення

```
print(f"Помилка: Не вдалося завантажити зображення за шляхом\n\nprint("Можливо, файл пошкоджений, має непідтримуваний формат або відсутні\nправа доступу.")
```

```
original_img = img.copy() # Зберігаємо копію оригінального зображення для
малювання
# Змінюємо розмір для послідовної обробки
```

Функція `cv2.cvtColor()` — конвертація кольорового простору. Ця функція дозволяє перетворювати зображення між різними кольорними просторами.

Згладжування та зменшення шуму, коли зображення, отримані з камер, майже завжди містять шум, який може заважати точній ідентифікації об'єктів або створювати хибні краї. Фільтри згладжування допомагають приглушити цей шум.

Функція `cv2.GaussianBlur()` — розмиття за Гаусом. Гаусів фільтр — це поширений фільтр низьких частот, який використовується для розмиття зображення та зменшення шуму. Він обчислює зважене середнє значення пікселів у певній околиці, де ваги визначаються Гаусовим розподілом, надаючи більшу вагу пікселям ближче до центру ядра.

Колірна сегментація є надзвичайно важливим кроком для дорожніх знаків, де колір є визначальною ознакою.

Функція `cv2.inRange()` — порогова фільтрація за кольором. Ця функція створює бінарну маску, яка виділяє всі пікселі, чий колір знаходиться в заданому діапазоні. Це найкраще працює в кольоровому просторі HSV.

Лістинг 3.4 — Застосування фільтра Гаусса

```
# Застосування Гаусового розмиття до зображення в градаціях сірого
# Параметри:
# - src: Вхідне зображення (тут: gray_img)
# - ksize: Розмір ядра Гаусового фільтра (наприклад, (5, 5)).
#         Ширина та висота ядра повинні бути непарними та позитивними.
# - sigmaX: Стандартне відхилення Гаусової функції по осі X. Якщо 0,
#         воно обчислюється автоматично на основі ksize.
```

```
# - sigmaY: Стандартне відхилення Гаусової функції по осі Y. Якщо 0,  
# воно береться рівним sigmaX.
```

```
cv2.imshow("Зображення після Гаусового розмиття", blurred_img)
```

Після того, як ми обробили зображення (виявили краї або створили колірну маску), наступним логічним кроком є пошук контурів — кривих, що з'єднують безперервні точки вздовж межі, які мають однакову інтенсивність або колір. Контури є ключовими для ідентифікації форм об'єктів.

Функція `cv2.findContours()` — пошук контурів. Ця функція виявляє контури на бінарному зображенні.

Лістинг 3.5 — Перетворення колірного простору та визначення країв дорожніх знаків

```
# Знаходимо контури  
  
recognized_sign_name = "Невідомий знак"  
found_bbox = None # Для збереження обмежувального прямокутника (x, y, w, h)  
розпізнаного знака  
  
  
if area > 500: # Відфільтровуємо дрібний шум (налаштуйте за потребою)  
  
  
# Перевіряємо наявність червоного кольору в обмежувальному прямокутнику  
контуру
```

Блок фільтрації зображень є фундаментальним кроком у багатьох задачах комп'ютерного зору. Ретельний вибір і послідовність застосування фільтрів може суттєво вплинути на якість подальших етапів обробки та кінцеву точність розпізнавання. Експерименти з різними параметрами та комбінаціями фільтрів є ключем до оптимізації цього блоку для конкретного завдання, такого як розпізнавання дорожніх знаків.

Класифікація дорожнього знаку за його геометричними ознаками

Після ретельної попередньої обробки зображення, яка згладила шум, виділила краї та колірні області, настає етап розпізнавання. У цьому блоці використовуємо геометричні ознаки знайдених об'єктів для ідентифікації дорожніх знаків. Це означає, що ми аналізуємо форму, розмір та співвідношення сторін потенційних знаків.

Попередній блок фільтрації зображень завершується отриманням бінарного зображення (наприклад, `edges_img` від Canny або `red_mask` від колірної сегментації), де об'єкти (або їхні межі) представлені білими пікселями на чорному фоні. Саме на цьому бінарному зображенні ми шукаємо контури.

Аналіз контурів та виділення ознак форми виконуємо після того, як отримали список потенційних контурів, потрібно проаналізувати кожен з них, щоб визначити, чи відповідає він формі дорожнього знаку, який ми шукаємо.

Ітерація та фільтрація контурів за площею — досить часто `findContours` знаходить багато дрібних контурів, які є просто шумом або незначними деталями. Фільтрація за площею допомагає відкинути такі контури.

Функція `cv2.arcLength(contour, closed)` — обчислення довжини периметра. Просто сумує довжини всіх сегментів, що складають контур. Параметри: `contour` — вхідний контур (масив точок), `closed` — булеве значення, `True` — контур замкнутий (як у випадку з об'єктами), `False` — відкрита крива, для дорожніх знаків завжди

Функція `cv2.approxPolyDP(contour, epsilon, closed)` — апроксимація багатокутника. Ця функція апроксимує контур до багатокутника з меншою

кількістю вершин, при цьому зберігаючи його основну форму. Вона використовує алгоритм Дугласа-Пекера. Функція "згладжує" контур, замінюючи криві сегменти прямими лініями, видаляючи зайві точки, які лежать на одній лінії з іншими. Параметри: `contour` — вхідний контур, `epsilon` — параметр точності. Це максимальна відстань між оригінальним контуром і його апроксимованим варіантом. Менші значення `epsilon` дають більш точну апроксимацію (більше вершин), більші — грубішу апроксимацію (менше вершин). Зазвичай `epsilon` встановлюється як певний відсоток від довжини периметра контуру (наприклад,

Класифікація знаків за формою та кількістю вершин. Визначення форми за кількістю вершин (`len(approx)`). Восьмикутник (знак "STOP") — восьмикутник має 8 вершин. Якщо `len(approx)` дорівнює 8, і співвідношення сторін близьке до 1.0, це сильний індикатор знака "STOP".

Трикутник (знак "Дай дорогу") має 3 вершини. Якщо `len(approx)` дорівнює 3, і співвідношення сторін близьке до 1.0, це може бути знаком "Дай дорогу". Для цього знака важливо також перевірити орієнтацію (чи спрямований він вершиною вниз).

Перевірка орієнтації трикутника, щоб визначити, чи спрямований трикутник вниз (як знак "Дай дорогу"), ми можемо порівняти Y-координату центроїда контуру (центру маси) з Y-координатою найвищої точки контуру.

Лістинг 3.6 — Код програми із визначення знаку за його ознаками

```
# Восьмикутна форма для знака "STOP"
recognized_sign_name = "Знак 'STOP'"

break # Припускаємо, що на зображенні є лише один основний знак для
спрощення

# Трикутна форма для знака "Дай дорогу" (спрямований вниз)
# Більш надійна перевірка передбачатиме орієнтацію вершин
```

Знаходимо найменшу координату Y серед вершин (верх трикутника)

Якщо центроїд нижче верхньої вершини, то, ймовірно, трикутник спрямований вниз

recognized_sign_name = "Знак 'Дай дорогу'"

Кругла форма (обмеження швидкості, в'їзд заборонено тощо)

Для справжнього круглого знака знадобиться більш точне виявлення кола

Для реальних обмежень швидкості потрібен OCR або зіставлення з шаблоном для чисел

Функція `cv2.moments(contour)` обчислює всі моменти зображення до третього порядку. Моменти використовуються для обчислення таких властивостей зображення, як площа, центроїд, орієнтація. З отриманих моментів можна обчислити координати центроїда (`cX`, `cY`). Якщо `cY` (Y-координата центроїда) більша, ніж `min_y_vertex` (Y-координата найвищої вершини трикутника), це вказує на те, що трикутник спрямований донизу.

Візуалізація результатів розпізнавання, після того, як знак розпізнано та його обмежувальний прямокутник ідентифіковано, повертаємося до оригінального зображення, щоб відобразити результати.

Функція `cv2.rectangle(img, pt1, pt2, color, thickness)` — малювання прямокутника. Малює прямокутник на зображенні. Використовує дві точки (верхній лівий та нижній правий кути) для визначення прямокутника. Параметри: — зображення, на якому потрібно малювати, `pt1`: — координати верхнього лівого кута (`x`, `y`), `pt2`: — координати нижнього правого кута (`x + w`, `y + h`), `color`: — колір прямокутника у форматі BGR (наприклад, `(0, 255, 0)` для зеленого), `thickness`: — товщина лінії прямокутника в пікселях.

Блок розпізнавання за геометричними ознаками є ефективним для простих сценаріїв, де дорожні знаки мають чітко визначені форми (восьмикутник, трикутник, коло). Однак, його надійність значно знижується в реальних умовах через зміни освітлення: які можуть спотворити кольори та утруднити виявлення країв; оклюзії: коли частина знака закрита (наприклад, листям або снігом); перспектива — знаки, сфотографовані під кутом, можуть виглядати спотвореними, і їхня "геометрична" форма може сильно відрізнятись від ідеальної. Різноманітність знаків — багато дорожніх знаків мають складніші форми або розрізняються лише внутрішніми символами (наприклад, різні обмеження швидкості), які цей метод не аналізує.

Саме через ці обмеження в сучасних системах розпізнавання дорожніх знаків широко застосовуються нейронні мережі, які можуть "вчитися" на великих обсягах даних і бути набагато стійкішими до варіацій у реальному світі. Однак, для освітніх цілей та розуміння базових принципів комп'ютерного зору, аналіз геометричних ознак є чудовою відправною точкою.

Результати роботи та тестування програми

Кінцеві результати роботи програми є найважливішим аспектом для користувача, оскільки вони безпосередньо відображають функціональність системи та її здатність надавати корисну інформацію.

Формат вихідних даних для кожного успішно розпізнаного дорожнього знака система генерує структурований набір даних.

Рівень впевненості є критично важливою інформацією для подальших систем або операторів. Низький рівень впевненості може ініціювати попередження, повторну оцінку або резервний механізм, що забезпечує прийняття рішень на основі даних. Це означає, що система надає не лише інформацію про те, що було виявлено, а й про те, наскільки вона впевнена у цьому, що дозволяє інтелектуально приймати рішення та оцінювати ризики у критично важливих для безпеки застосуваннях. Перелічені специфічні точки даних свідчать про стандартизований формат вихідних даних. Це має вирішальне значення, оскільки модуль

розпізнавання часто є частиною більшої системи ADAS або автономного водіння. Послідовний та чітко визначений вихід (див. табл. 3.2) дозволяє безперешкодно інтегруватися з іншими модулями (наприклад, навігацією, керуванням транспортним засобом), мінімізуючи проблеми сумісності та сприяючи масштабованості системи.

Таблиця 3.2 — Формат вихідних даних розпізнавання

Поле даних	Опис	Приклад формату
	Унікальний ідентифікатор розпізнаного знака в поточному кадрі.	
	Координати обмежувального прямокутника:	
	Класифікований тип дорожнього знака.	"Обмеження швидкості 50"
	Рівень впевненості розпізнавання (від 0.0 до	
	Часова мітка кадру, в якому було розпізнано знак.	

Візуалізація результатів на вихідних рисунках для інтуїтивно зрозумілого представлення результатів, особливо для операторів або систем моніторингу, розпізнані дані накладаються безпосередньо на вихідні зображення. Візуальні елементи включають обмежувальні прямокутники, які малюються навколо кожного виявленого знака, візуально підтверджуючи його локалізацію. Поруч із прямокутником відображається текстова мітка, що вказує на тип розпізнаного знака (наприклад, "СТОП", "50 км/год") та, за бажанням, рівень впевненості. Прямокутники можуть бути кодовані кольором для швидкого розрізнення типів знаків (наприклад, червоний для заборонних, жовтий для попереджувальних) або для відображення рівня впевненості (наприклад, зелений для високої, помаранчевий для середньої).

Ця візуалізація забезпечує миттєвий зворотний зв'язок, дозволяючи людині або іншій системі швидко оцінити результати розпізнавання та зрозуміти контекст.

Акцент на візуальному зворотньому зв'язку для операторів/водіїв вказує на покращення взаємодії людини та машини. Чіткі, інтуїтивно зрозумілі візуальні підказки зменшують когнітивне навантаження та дозволяють швидше реагувати або перевіряти систему людиною. Візуальне накладання виявлених знаків та їхніх рівнів впевненості також забезпечує прозорість системи, дозволяючи користувачам бачити, що "бачить" система і наскільки вона впевнена, що є вирішальним для формування довіри в критично важливих для безпеки застосуваннях.

Типи розпізнаваних знаків система здатна розпізнавати широкий спектр дорожніх знаків, що охоплює основні категорії (рис. 3.5). Це включає регулюючі знаки (наприклад, "Стоп", "Рух заборонено"), попереджувальні знаки (наприклад, "Небезпечний поворот", "Пішохідний перехід"), інформаційно-вказівні знаки (наприклад, "Місце відпочинку", "Напрямок руху") та знаки пріоритету.

Система може бути адаптована для розпізнавання знаків, що відповідають специфічним стандартам та варіаціям різних країн або регіонів, що забезпечується шляхом навчання на відповідних локалізованих наборах даних (рис. 3.6). Здатність системи адаптуватися до "варіацій за країною/регіоном" свідчить про її глобальну масштабованість та локалізацію. Це означає, що комерційно життєздатна або глобально розгорнута система не може бути монолітною; вона повинна бути або високоадаптивною (наприклад, через модульні бібліотеки знаків), або вимагати перенавчання/тонкого налаштування для конкретних географічних регіонів.



Рисунок 3.5 — Результат розпізнавання знаку STOP

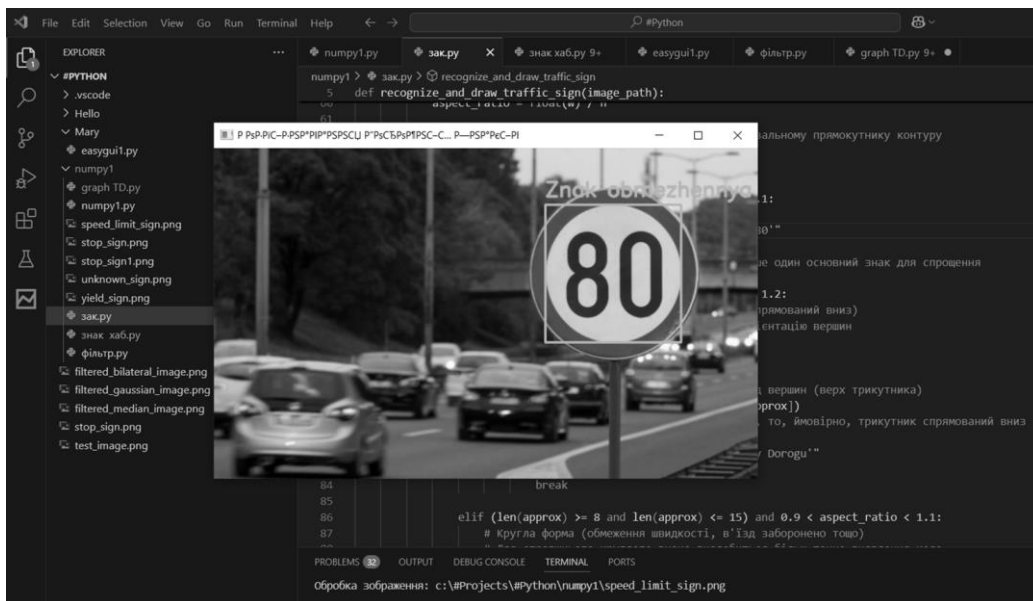


Рисунок 3.6 — Результат розпізнавання знаку обмеження швидкості

Модуль розпізнавання дорожніх знаків зазвичай є частиною більших, складніших систем. Він може бути інтегрований у системи допомоги водієві для попередження про знаки обмеження швидкості або знаки "Стоп", у системи автономного водіння для навігації та прийняття рішень, а також у системи моніторингу дорожнього руху для збору даних. Така інтеграція дозволяє підвищити ситуаційну обізнаність транспортного засобу, покращити безпеку та ефективність дорожнього руху, а також автоматизувати рутинні завдання водіння. Крім того, стандарти дорожніх знаків можуть змінюватися, і можуть з'являтися нові знаки (рис. 3.7).



Рисунок 3.7 — Результат розпізнавання знаку «Поступись дорогою»

Здатність розпізнавати різні типи знаків передбачає, що система має механізм для оновлення своєї бази знань, що вказує на постійну потребу в оновленні знань через поточне обслуговування, потенційно за допомогою бездротових оновлень або повторного розгортання оновлених моделей, щоб забезпечити її актуальність та ефективність.

Справжня цінність розпізнавання знака полягає не лише в самому розпізнаванні, а в тому, що ця інформація дозволяє зробити в більшій системі (наприклад, коригування швидкості транспортного засобу, оновлення навігації). Це означає, що справжній вплив системи реалізується, коли її вихідні дані споживаються та використовуються іншими інтелектуальними модулями, що призводить до синергетичного ефекту, який підвищує загальний інтелект та безпеку транспортного засобу.

ВИСНОВКИ

Система розпізнавання дорожніх знаків є високотехнологічним рішенням, яке інтегрує передові методи обробки зображень та машинного навчання для забезпечення безпеки та ефективності дорожнього руху. Її архітектура, що включає попередню обробку, виявлення, класифікацію та постобробку, розроблена для роботи в реальному часі та в складних умовах. Кінцеві результати надаються у структурованому форматі, що містить точні координати, тип знака, а також візуалізуються безпосередньо на вихідних зображеннях для покращення взаємодії з користувачем. Здатність системи адаптуватися до різноманітних умов та типів знаків, а також її інтеграція в ширші системи автономного водіння, підкреслюють її критичну роль у розвитку інтелектуального транспорту. Постійне вдосконалення та адаптація до нових даних та сценаріїв є ключовими для підтримки її високої продуктивності та надійності в динамічному світі.

Розглянуто проблему визначення дорожніх знаків та підходи до її вирішення. Виявлено, що ефективність розпізнавання суттєво знижується за наявності низької роздільної здатності зображень, складної фонові структури, різних спотворень внаслідок невдалих ракурсів, поганих погодних умов, різних кутів та умов освітлення, шумових перешкод, афінних та проекційних спотворень.

Досліджено популярні фреймворки для реалізації алгоритмів навчання, такі як Python, а також їх особливості, переваги та недоліки. Python відзначений як повноцінна мова програмування з високорівневими типами даних та великою бібліотекою стандартних модулів.

Розроблена комп'ютерна система розпізнавання дорожніх знаків успішно вирішує поставлену задачу з допустимою точністю.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ciresan U. Meier J. Masci L. M. Gambardella J. Schmidhuber Flexible. High Performance Convolutional Neural Networks for Image Classification. Proceedings of the Twenty-Second international joint conference on Artificial Intelligence, 2011. P. 1237–1242.
2. Open CV Library. URL: <http://opencv.org/downloads.html> (дата звернення: 31.03.2025).
3. Метод розпізнавання дорожніх знаків на основі Хеммінгової віддалі. URL: <https://www.researchgate.net/328144774> (дата звернення: 29.04.2025). 3
4. Стан функції відображення дорожніх знаків. URL: <https://www.volvocars.com/uk-ua/support/3a091da> (дата звернення: 30.04.2025).
5. Jianmin Duan and M. Viktor. Real time road edges detection and road signs recognition. International Conference on Automation Sciences. 2015. pp. 107-112. URL: <https://ieeexplore.ieee.org/document/7338642> (дата звернення: 29.05.2023).
6. Y. -L. Kuo and S. -H. Lin. Applications of Deep Learning to Road Sign Detection in DVR Images. International Symposium on Measurement and Control in Robotics. 2019, pp. 1-6, doi: 10.1109/ISMCR47492.2019.8955719.
7. A. Z. Syaharuddin, Z. Zainuddin and Andani. Multi-Pole Road Sign Detection Based on Faster Region-based Convolutional Neural Network. International Conference on Artificial Intelligence (AIMS). 2021. pp. 1-5.
8. W. H. D. Fernando and S. Sotheeswaran. Automatic road traffic signs detection and recognition using 'You Only Look Once' version 4 (YOLOv4). International Research Conference on Systems Engineering 2021, pp. 38-43. URL: <https://ieeexplore.ieee.org/document/9568285> (дата звернення: 29.05.2023).
9. R. Qian, Q. Liu, Y. Yue, F. Coenen and B. Zhang. Road surface traffic sign detection with hybrid region proposal and fast R-CNN. International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery. 2016, pp. 555-559. URL: <https://ieeexplore.ieee.org/7603233> (дата звернення: 24.05.2025).

10. Jongmin Y., Bongnam K., Kim D., Car License Plate Detection under Large Variations Using Covariance and HOG Descriptors. 8th International Symposium on Visual Computing (ISVC). 2012. P. 636–647.

8. Method and apparatus for identifying scale invariant features in an image and use of same for locating an object in an image: patent No. US6711293B1 Appl. No. 09/519,893; Filed 06.03.2000; publ. 23.03.2004, 20 p.

9. Dalal N., Triggs B., Histograms of oriented gradients for human detection. IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Montbonnot, France 2005. P. 886-893

10. Bay H., Ess A., Tuytelaars T., Luc Van Gool. Speeded Up Robust Features. Computer Vision and Image Understanding, 2008, doi: 10.1016/J.CVIU.2007.09.014

11. Schapire R. E., Singer Y., Improved Boosting Algorithms Using Confidence-rated Predictions. Machine Learning. Netherlands: Kluwer Academic Publishers, 1999, P. 297–336.

12. Viola P. Robust real-time face detection. International Journal of Computer Vision, 2004. Vol. 57, No. 2. P.137–145.

13. Houben S. Detection of Traffic Signs in Real-World Images: The German Traffic Sign Detection Benchmark. International Joint Conference on Neural Networks. 2013. C. 58–60.

ДОДАТОК А

Технічне завдання

Міністерство освіти і науки України

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра обчислювальної техніки

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ

д.т.н., проф. Азаров О.Д.

«21» березня 2025 року

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання бакалаврської кваліфікаційної роботи

««Комп'ютерна система розпізнавання дорожніх знаків»»

08-54.БКР.014.00.000 ТЗ

Науковий керівник phd, ст.вик. каф. ОТ

_____ Обертюх М.Р.

Студент групи ІСП-216

_____ Стадник О.В.

1 Підстави для виконання бакалаврської кваліфікаційної роботи (БКР)

1.1 Розробка системи розпізнавання дорожніх знаків є актуальною через зростання попиту на автоматизовані системи керування транспортними засобами та підвищення безпеки дорожнього руху в умовах інтенсивного трафіку. Сучасні виклики, такі як складні погодні умови, часткове перекриття знаків і різноманітність їхніх типів, вимагають створення стійких і точних алгоритмів комп'ютерного зору. Інтеграція таких систем у технології ADAS і автономні автомобілі сприяє зниженню кількості аварій, спричинених людським фактором. Крім того, використання глибокого навчання та спеціалізованих бібліотек, як у цій роботі, відповідає глобальним тенденціям розвитку інтелектуальних транспортних систем

1.2 Наказ про затвердження теми БКР ректором університету №97 від 20 березня 2025 року.

2 Мета БКР і призначення розробки

2.1 Метою роботи є розширення функціональних можливостей комп'ютерної системи для ефективного розпізнавання дорожніх знаків, яка забезпечує надійне виявлення та ідентифікацію знаків у складних умовах видимості для підвищення безпеки дорожнього руху та підтримки автоматизованих систем керування транспортними засобами.

2.2 Призначення розробки полягає у створенні програмної системи, здатної класифікувати зображення. Для цього застосовуються сучасні підходи, такі як аналіз контурів та обробка просторових зв'язків між фрагментами зображення. Таким чином, розроблена система забезпечує надійне розпізнавання дорожніх знаків, сприяючи підвищенню безпеки дорожнього руху та ефективності роботи автоматизованих систем керування транспортними засобами.

3 Вихідні дані для виконання БКР

3.1 Проведення аналізу сучасних методів обробки та класифікації зображень

коду, а також огляд інструментів і бібліотек, що використовуються у сфері комп'ютерного зору у середовищі Windows.

3.2 Проектування структури системи та розробка загальної структури системи, визначення основних модулів та принципів їх взаємодії.

3.3 Реалізація функціональних компонентів та розробка комп'ютерної моделі роботи системи: модулів попередньої обробки, визначення країв, сегментації та класифікації зображень.

3.4 Тестування та перевірка працездатності — визначення вхідних тестових зображень для перевірки правильності розпізнавання дорожніх знаків та визначення коректності їх розпізнавання.

3.5 Відповідність вимогам безпеки та забезпечення сумісності з архітектурою Windows та дотримання принципів безпечного доступу до пам'яті.

4 Вимоги до виконання БКР

Головна вимога — реалізувати систему для ефективного розпізнавання дорожніх знаків, яка забезпечує надійне виявлення та ідентифікацію знаків у складних умовах видимості для підвищення безпеки дорожнього руху та підтримки автоматизованих систем керування транспортними засобами. Особливу увагу слід приділити реалізації механізму попередньої обробки та сегментації зображень.

5 Етапи БКР та очікувані результати

5.1 Підготовчий етап — аналіз літературних джерел, наукових статей та технічної документації з тематики роботи, вивчення архітектури засобів програмної розробки комп'ютерних систем обробки зображень. Очікувані результати: формування теоретичної бази для реалізації системи, виявлення недоліків існуючих рішень та постановка вимог до розробки.

5.2 Етап розробки концепції та формування загальної структури системи, визначення основних модулів, їх функцій та взаємозв'язків. Очікувані результати: структура системи та логічна схема взаємодії її компонентів.

5.3 Етап проектування та визначення сучасних програмних інструментів для розробки системи обробки зображень. Аналіз можливостей мов програмування та спеціалізовані бібліотеки та фреймворки для обробки зображень. Вибір інтегрованого середовища розробки (IDE) та інструментів розробки.

5.4 Етап тестування та відлагодження — проведення тестів на різних прикладах зображень дорожніх знаків, перевірка точності роботи механізмів розпізнавання. Очікувані результати: виявлені помилки, підвищення точності системи.

5.5 Етап оформлення та аналізу результатів: оформлення пояснювальної записки, графічного матеріалу, формулювання висновків та узагальнення отриманих результатів. Очікувані результати: завершений звіт з описом реалізації, підтвердженою працездатністю та аналізом досягнутих цілей.

6 Матеріали, що подаються до захисту БКР

До захисту подаються: пояснювальна записка БКР, графічні та ілюстративні матеріали, протокол попереднього захисту на кафедрі, відгуки наукового керівника та опонента, протоколи складання державних екзаменів, анотації до БКР українською та іноземною мовами, довідка про відповідність оформлення БКР діючим вимогам.

7 Порядок контролю виконання та захисту БКР

Виконання етапів документації БКР контролюється науковим керівником згідно зі встановленими термінами. Захист БКР відбувається на засіданні Екзаменаційної комісії, затвердженої наказом ректора.

8 Вимоги до оформлювання та порядок виконання БКР

8.1 При оформлюванні БКР використовуються:

— ДСТУ 3008: 2015 «Звіти в сфері науки і техніки. Структура та правила оформлювання»;

— ДСТУ 8302: 2015 «Бібліографічні посилання. Загальні положення та правила складання»;

— методичні вказівки до виконання бакалаврських кваліфікаційних робіт зі спеціальності 123 – «Комп'ютерна інженерія» (освітня програма «Системне програмування»). Кафедра обчислювальної техніки ВНТУ 2023.

8.2 Порядок виконання БКР викладено в «Положення про кваліфікаційні роботи на першому (бакалаврському) рівні вищої освіти СУЯ ВНТУ-03.02.02-П.001.01:21».

ДОДАТОК Б
ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: Комп'ютерна система розпізнавання дорожніх знаків

Тип роботи: бакалаврська кваліфікаційна робота
бакалаврська кваліфікаційна робота / магістерська кваліфікаційна робота)

Підрозділ кафедра обчислювальної техніки
кафедра, факультет, навчальна група)

Коефіцієнт подібності текстових запозичень, виявлених у роботі
системою StrikePlagiarism 1,5 %

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

- ✓ Запозичення, виявлені у роботі, є законними і не містять ознак плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту
- ☐ У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.
- ☐ У роботі виявлено ознаки плагіату та/або текстових маніпуляцій як спроб укриття плагіату, фабрикації, фальсифікації, що суперечить вимогам законодавства та нормам академічної доброчесності. Робота до захисту не приймається.

Експертна комісія:

Азаров О.Д., зав. каф. ОТ
(прізвище, ініціали, посада) (підпис)

Т а
(прізвище, ініціали, посада) (підпис)

р
Особа, відповідальна за перевірку Захарченко С.М.
(прізвище, ініціали) (підпис)

с
Висновком експертної комісії ознайомлений(-на)

К
и Керівник Обертюх М.Р.
підпис) (прізвище, ініціали, посада)

М
Здобувач Стадник О.В.
підпис) (прізвище, ініціали)

Г

:

2

д

ДОДАТОК В
Графічна частина

КОМП'ЮТЕРНА СИСТЕМА РОЗПІЗНАВАННЯ ДОРОЖНІХ ЗНАКІВ

Рисунок В.1 — Структурна схема комп'ютерної системи

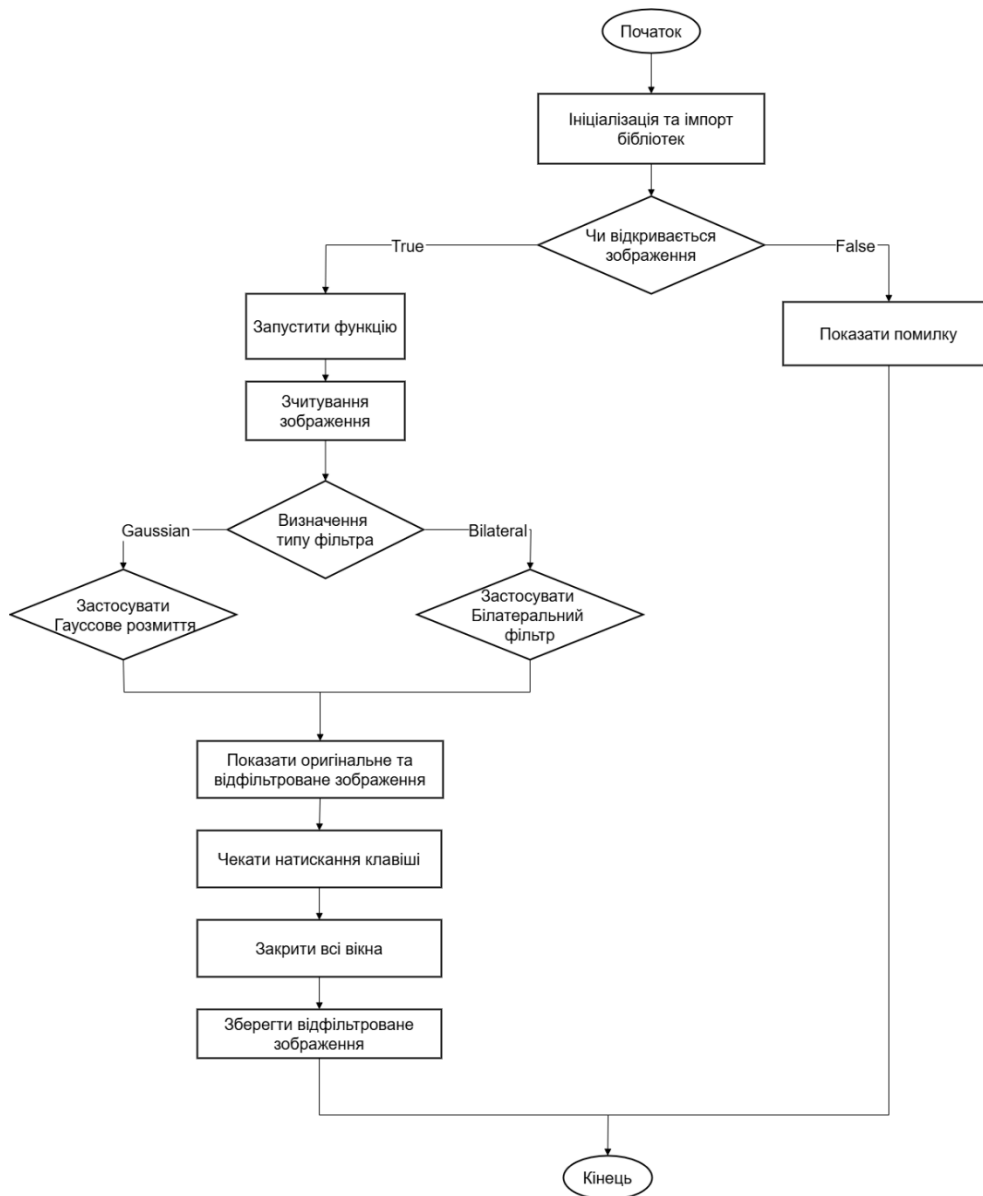


Рисунок В.2 — Блок-схема модуля попередньої обробки інформації

ДОДАТОК Г

Ілюстративна частина

К
О
М
П
,
Ю
Т
Е
Р
Н
А

С
И
С
Т
Е
М
А

Р
О
З
П



Рисунок Г.1 — Результат виконання роботи програми

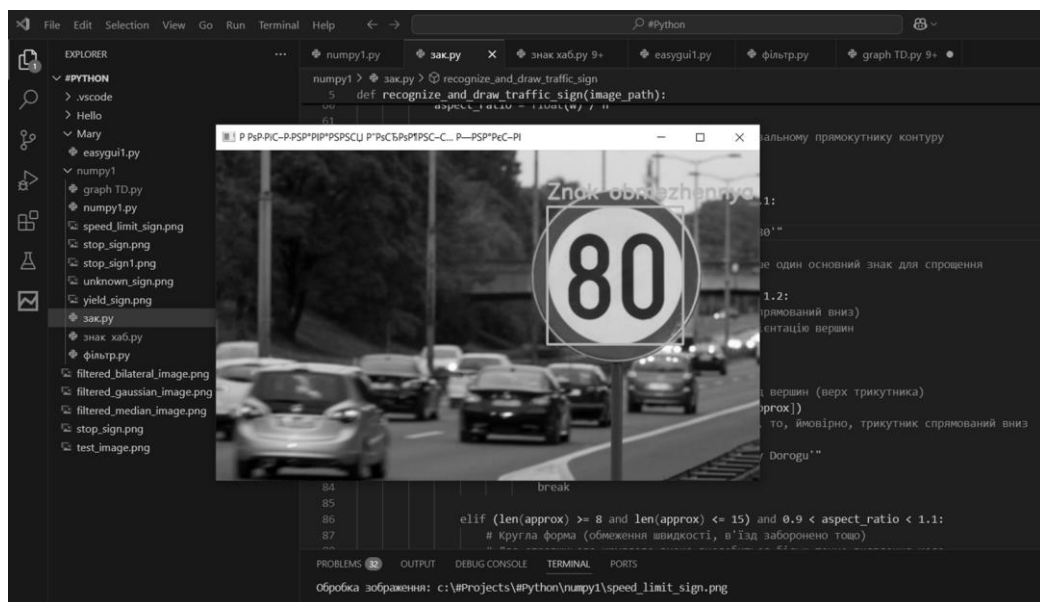


Рисунок Г.2 — Результат розпізнавання знаку обмеження швидкості

ДОДАТОК Д

Лістинг програмного забезпечення

```
import os # Імпортуємо модуль os для роботи зі шляхами до файлів

# --- Перевірка наявності файлу перед завантаженням ---

print(f"ПОПЕРЕДЖЕННЯ: Файл не знайдено за шляхом: {image_path}.
Пропускаємо.")

print(f"Помилка: Не вдалося завантажити зображення за шляхом

print("Можливо, файл пошкоджений, має непідтримуваний формат або
відсутні права доступу.")

original_img = img.copy() # Зберігаємо копію оригінального зображення для
малювання

# Змінюємо розмір для послідовної обробки

# Визначаємо діапазони кольорів для типових кольорів знаків (наприклад,
червоний для STOP/YIELD)
# Ці діапазони можуть потребувати налаштування
```

Знаходимо контури

recognized_sign_name = "Невідомий знак"
found_bbox = None # Для збереження обмежувального прямокутника (x, y, w, h)
розпізнаного знака

if area > 500: # Відфільтровуємо дрібний шум (налаштуйте за потребою)

Перевіряємо наявність червоного кольору в обмежувальному
прямокутнику контуру

--- Розпізнавання на основі форми ---

Восьмикутна форма для знака "STOP"
recognized_sign_name = "Знак 'STOP'"

break # Припускаємо, що на зображенні є лише один основний знак
для спрощення

Трикутна форма для знака "Дай дорогу" (спрямований вниз)

Більш надійна перевірка передбачатиме орієнтацію вершин
Знаходимо найменшу координату Y серед вершин (верх
трикутника)

Якщо центроїд нижче верхньої вершини, то, ймовірно, трикутник
спрямований вниз

```
recognized_sign_name = "Знак 'Дай дорогу'"
```

```
# Кругла форма (обмеження швидкості, в'їзд заборонено тощо)  
# Для справжнього круглого знака знадобиться більш точне виявлення  
кола
```

```
# Для реальних обмежень швидкості потрібен OCR або зіставлення з  
шаблоном для чисел
```

```
# Масштабуємо координати обмежувального прямокутника назад до розміру  
оригінального зображення
```

```
# Малюємо прямокутник  
cv2.rectangle(original_img, (x_orig, y_orig), (x_orig + w_orig, y_orig + h_orig),  
(0, 255, 0), 2) # Зелений прямокутник
```

```
# Додаємо текст  
# Регулюємо розмір тексту залежно від ширини зображення
```

```
# Переконаємося, що текст не виходить за межі зображення зверху
```

```
# --- Відображаємо зображення ---  
cv2.imshow("Розпізнавання Дорожніх Знаків", original_img)  
cv2.waitKey(0) # Чекаємо, доки буде натиснута будь-яка клавіша  
cv2.destroyAllWindows() # Закриваємо всі вікна OpenCV
```

```
# --- Основна частина програми ---
```

```
print("Будь ласка, переконайтеся, що у вас є тестові зображення дорожніх знаків  
(наприклад, 'stop_sign.png', 'yield_sign.png', 'speed_limit_sign.png')")  
print("у тому ж каталозі, що й цей скрипт, або вкажіть повний шлях.")  
print("Натисніть будь-яку клавішу після перегляду кожного зображення, щоб  
продовжити.")
```

```
# Отримуємо шлях до каталогу, де знаходиться цей скрипт
```

```
# Це робить шляхи до зображень відносними до розташування скрипта
```

```
os.path.join(script_dir, 'yield_sign.png'),    # Приклад: червоний трикутник,  
спрямований вниз  
os.path.join(script_dir, 'speed_limit_sign.png'), # Приклад: круглий знак,  
напрямок швидкості  
os.path.join(script_dir, 'stop_sign.png'),      # Приклад: червоний восьмикутник  
os.path.join(script_dir, 'non_existent_sign.png'), # Приклад: файл, якого не існує  
(для тестування перевірки)  
os.path.join(script_dir, 'unknown_sign.png')    # Приклад: зображення без  
розпізнаного знака ]
```

```
print(f"\nОбробка зображення: {img_path}")
```

```
print("\nОбробка завершена.")
```