

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ШТУЧНОГО ІНТЕЛЕКТУ

КВАЛІФІКАЦІЙНА РОБОТА
на тему: «Система автоматичного розпізнавання дорожніх
знаків на основі методів машинного навчання»

на здобуття освітнього ступеня бакалавра
зі спеціальності 122 Комп'ютерні науки
(код, найменування спеціальності)
освітньо-професійної програми Штучний інтелект
(назва)

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання
на відповідне джерело*

(підпис) Іван БАБІЧЕВ
Ім'я, ПРІЗВИЩЕ здобувача

Виконав:
здобувач вищої освіти
група ШІД-42

Іван БАБІЧЕВ

-
-

Керівник:
науковий ступінь,
вчене звання

Тетяна КИСІЛЬ

Рецензент:
науковий ступінь,
вчене звання

Ім'я, ПРІЗВИЩЕ

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра Штучного інтелекту
Ступінь вищої освіти Бакалавр
Спеціальність Комп'ютерні науки
Освітньо-професійна програма Штучний інтелект

ЗАТВЕРДЖУЮ
Завідувач кафедри Штучного інтелекту

_____ Ольга ЗІНЧЕНКО
« _____ » _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Бабічева Івана Євгенійовича
(*прізвище, ім'я, по батькові здобувача*)

1. Тема кваліфікаційної роботи: підвищення ефективності розпізнавання дорожніх знаків та порівняння ефективності з іншими алгоритмами.

керівник кваліфікаційної роботи Тетяна КИСІЛЬ старший викладач,
(*Ім'я, ПРІЗВИЩЕ науковий ступінь, вчене звання*)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «24» лютого 2025 р. № 56

2. Строк подання кваліфікаційної роботи «23» травня 2025 р.

3. Вихідні дані до кваліфікаційної роботи: методи та алгоритми комп'ютерного зору, машинного навчання та обробки зображень.

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

1. глибокий аналіз та порівняння сучасних методів і алгоритмів, що використовуються для розпізнавання дорожніх знаків, включаючи підходи на основі комп'ютерного зору.
2. Розробити оптимальну архітектуру системи розпізнавання дорожніх знаків, що включатиме етапи захоплення зображення, попередньої обробки, виявлення знаків, їх класифікації та представлення результатів.

3. Вибір та адаптація методів виявлення знаків: Дослідити та обрати найбільш ефективні методи локалізації дорожніх знаків на зображенні, враховуючи різноманітність форм, розмірів та умов освітлення.
5. Перелік графічного матеріалу: *презентація*
1. Загальний огляд та концепція
2. Збір та підготовка даних
3. Архітектура та виявлення об'єктів (YOLO)
4. Класифікація та розпізнавання
5. Технічні аспекти / Використані інструменти
6. Дата видачі завдання «25» лютого 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури	25.02-05.03.25	Виконано
2	Огляд предметної області детектування та розпізнавання дорожніх знаків	08.03-12.04.25	Виконано
3	Аналіз популярних архітектур нейронних мереж для розпізнавання об'єктів	13.04-19.04.25	Виконано
4	Процес навчання моделі	20.04-25.04.25	Виконано
5	Реалізація компонентів системи:	27.04-03.05.25	Виконано
6	Інтеграція моделі YOLOv5	04.05-10.05.25	Виконано
7	Обробка та візуалізація результатів розпізнавання	11.05-20.05.25	Виконано
8	Аналіз результатів розпізнавання на різних зображеннях	21.05-23.05.25	Виконано

Здобувач вищої освіти

(підпис)

Іван БАБІЧЕВ

(Ім'я, ПРІЗВИЩЕ)

Керівник
кваліфікаційної роботи

(підпис)

Тетяна КИСІЛЬ

(Ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 79 стор., 3 табл., 19 рис., 23 джерел.

Мета роботи – підвищення ефективності розпізнавання дорожніх знаків та порівняння ефективності з іншими алгоритмами.

Об'єкт дослідження – процес розпізнавання дорожніх знаків за допомогою машинного навчання.

Предмет дослідження – методи та алгоритми комп'ютерного зору, машинного навчання та обробки зображень.

Короткий зміст роботи. У цьому дипломному проєкті була розроблена і реалізована система розпізнавання дорожніх знаків на автомобілі. Метою проєкту було створення ефективною та надійною системи, яка здатна автоматично розпізнавати дорожні знаки з використанням відеозображення, отриманого з камери автомобіля.

Досліджені засоби для вирішення поставленої проблеми – розглянуті моделі нейронних мереж, придатні для розпізнавання дорожніх знаків по зображенню знаку.

Проведено навчання моделі нейронної мережі для розпізнавання дорожніх знаків: налаштоване середовище для навчання, підготовлені зображення знаків з відповідного дата-сету, створені необхідні для тренування файли та успішно виконано навчання моделі.

КЛЮЧОВІ СЛОВА: YOLO, PYTORCH, МАШИННЕ НАВЧАННЯ, РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ, ДОРОЖНІ ЗНАКИ.

ЗМІСТ

ВСТУП.....	8
1 ТЕОРЕТИЧНІ ОСНОВИ СИСТЕМ ВІЗУАЛЬНОГО РОЗПІЗНАВАННЯ ТА МАШИННОГО НАВЧАННЯ	9
1.1 Поняття візуального розпізнавання та його роль у сучасних системах.....	9
1.2 Огляд методів машинного та глибинного навчання для комп'ютерного зору	11
1.3 Аналіз популярних архітектур нейронних мереж для розпізнавання об'єктів.....	14
1.4 Сучасні тенденції розвитку візуального розпізнавання.....	18
1.5 Аналіз існуючих систем візуального розпізнавання об'єктів.....	23
2 ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ СИСТЕМИ АВТОМАТИЧНОГО РОЗПІЗНАВАННЯ ДОРОЖНІХ ЗНАКІВ.....	32
2.1 Технічні вимоги до системи розпізнавання дорожніх знаків.....	32
2.2 Обґрунтування вибору моделі YOLO та її особливостей.....	34
2.3 Загальна архітектура системи розпізнавання дорожніх знаків	38
2.4 Середовище розробки та використані інструменти в системі розпізнавання дорожніх знаків	40
2.5 Процес навчання та донавчання моделі.....	43
2.6 Реалізація компонентів системи розпізнавання дорожніх знаків	47
2.7 Розробка інтерфейсу системи розпізнавання дорожніх знаків	50
3 ТЕСТУВАННЯ, ОЦІНКА ЕФЕКТИВНОСТІ ТА АНАЛІЗ ОБМЕЖЕНЬ СИСТЕМИ	56
3.1. Опис використаного набору даних дорожніх знаків.....	56
3.2 Методика тестування та критерії оцінювання	58
3.3 Аналіз результатів розпізнавання на різних зображеннях	60
3.4 Оцінка продуктивності та стабільності роботи системи	65
3.5 Виявлені обмеження та можливі шляхи їх усунення.....	68
ВИСНОВКИ.....	73
ПЕРЕЛІК ПОСИЛАНЬ	75
ДОДАТКИ.....	78
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ	78

ВСТУП

Стрімкий розвиток технологій в галузі автомобілебудування та систем штучного інтелекту відкриває нові горизонти для підвищення безпеки дорожнього руху та ефективності транспортних потоків. Одним з ключових напрямків, що активно досліджується та впроваджується, є розробка інтелектуальних систем допомоги водію (ADAS), серед яких особливе місце посідають системи автоматичного розпізнавання дорожніх знаків.

Зростання інтенсивності дорожнього руху, урбанізація та людський фактор залишаються основними причинами дорожньо-транспортних пригод. Забуття або ігнорування дорожніх знаків, особливо в умовах обмеженої видимості, несприятливих погодних умов або втоми водія, може призвести до критичних наслідків. Системи автоматичного розпізнавання дорожніх знаків (СРДЗ) здатні забезпечити водія своєчасною та точною інформацією про поточні обмеження та умови руху, значно знижуючи ризик помилок і підвищуючи загальну безпеку на дорогах. Застосування методів машинного навчання, зокрема глибокого навчання, дозволяє створювати високоефективні та надійні СРДЗ, здатні адаптуватися до різноманітних умов та забезпечувати високу точність розпізнавання. Це робить розробку та вдосконалення таких систем надзвичайно актуальним завданням сучасної інженерії та інформаційних технологій.

В даній дипломній роботі розглядається створення системи автоматичного розпізнавання дорожніх знаків, яка базується на передових методах машинного навчання. Основний акцент зроблено на використанні згорткових нейронних мереж (CNN) для ефективного виявлення та класифікації знаків в умовах різноманітних візуальних перешкод та освітлення. Розглянуто етапи підготовки даних, вибору та адаптації архітектури моделі, а також методи її навчання та оптимізації. Результатом роботи стане розроблений прототип системи, здатний розпізнавати дорожні знаки з високою точністю, що демонструє потенціал застосування таких технологій для підвищення безпеки на дорогах.

1 ТЕОРЕТИЧНІ ОСНОВИ СИСТЕМ ВІЗУАЛЬНОГО РОЗПІЗНАВАННЯ ТА МАШИННОГО НАВЧАННЯ

1.1 Поняття візуального розпізнавання та його роль у сучасних системах

Візуальне розпізнавання, або розпізнавання образів (pattern recognition), є однією з фундаментальних задач у галузі комп'ютерного зору та штучного інтелекту. Воно полягає в автоматичному виявленні, ідентифікації та класифікації об'єктів, ознак, подій або закономірностей у зображеннях чи відеопотоках. Ця дисципліна вивчає методи, що дозволяють машинам "бачити" та "розуміти" візуальну інформацію подібно до того, як це робить людина.

Історично візуальне розпізнавання пройшло шлях від простих алгоритмів, заснованих на правилах та порогових значеннях, до складних систем, що використовують методи машинного та глибокого навчання. Ранні підходи включали виділення контурів, форм та колірних характеристик для ідентифікації відомих об'єктів. Однак їхня ефективність була обмежена через високу чутливість до змін у освітленні, куті огляду, деформацій та наявності шуму.

Сучасний етап розвитку візуального розпізнавання тісно пов'язаний з прогресом у сфері машинного навчання, зокрема з появою та стрімким розвитком **глибоких нейронних мереж**, ієрархічно вивчають ознаки з сирих даних. Вони дозволяють системам самостійно "вчитися" виділяти складні та абстрактні ознаки, необхідні для розпізнавання, що значно підвищило точність та надійність алгоритмів.

Основні етапи процесу візуального розпізнавання зазвичай включають:

1. **Збір та попередню обробку даних.** Отримання зображень або відео, нормалізація розміру, корекція освітлення, видалення шуму.

2. **Виділення ознак (Feature Extraction).** Ідентифікація характерних рис об'єкта, які є унікальними та репрезентативними для його розпізнавання (наприклад, кути, градієнти, колірні гістограми). У глибоких мережах цей етап часто інтегрований у процес навчання.

3. Класифікація (Classification). Віднесення об'єкта до певного класу на основі виділених ознак. Цей етап виконується за допомогою різних алгоритмів машинного навчання (наприклад, метод опорних векторів, дерева рішень, нейронні мережі).

4. Постобробка. Уточнення результатів розпізнавання, усунення хибних спрацьовувань, інтеграція з іншими модулями системи.

Роль візуального розпізнавання у сучасних системах є надзвичайно важливою і постійно зростає, охоплюючи широкий спектр галузей:

- *Автономні системи та робототехніка.* Це ядро для навігації безпілотних автомобілів, дронів, промислових роботів. СРДЗ є ключовим компонентом ADAS, що дозволяє транспортним засобам "бачити" та інтерпретувати дорожню обстановку, включаючи розпізнавання дорожніх знаків, світлофорів, інших учасників руху та пішоходів.
- *Безпека та відеоспостереження.* Системи розпізнавання облич, виявлення підозрілої активності, аналіз натовпу, ідентифікація номерних знаків транспортних засобів.
- *Медицина.* Аналіз медичних зображень (рентген, МРТ, КТ) для діагностики захворювань, виявлення пухлин, моніторингу стану пацієнтів.
- *Промисловість та контроль якості.* Автоматизована інспекція продукції, виявлення дефектів, сортування.
- *Сільське господарство.* Моніторинг стану посівів, виявлення шкідників, автоматизація збору врожаю.
- *Споживчі технології.* Розпізнавання облич для розблокування пристроїв, фільтри та ефекти в камерах смартфонів, розумні домашні системи.

У контексті даної дипломної роботи, **система автоматичного розпізнавання дорожніх знаків** є яскравим прикладом практичного застосування візуального розпізнавання. Дорожні знаки є стандартизованими візуальними символами, які, проте, можуть суттєво відрізнятися за розміром, освітленням, кутом огляду, ступенем забруднення або пошкодження. Ефективне розпізнавання цих знаків в режимі реального часу є критично важливим для систем допомоги

водію, забезпечуючи інформування про швидкісні обмеження, повороти, заборони та інші дорожні умови, тим самим значно підвищуючи безпеку руху. Отже, візуальне розпізнавання виступає як фундаментальна технологія, що лежить в основі створення інтелектуальних систем, здатних взаємодіяти з навколишнім світом через зір, і є ключовим елементом для інновацій у багатьох галузях.

1.2 Огляд методів машинного та глибинного навчання для комп'ютерного зору

Швидкий прогрес у галузі машинного навчання (Machine Learning, ML) та глибинного навчання (Deep Learning, DL) трансформував підхід до вирішення задач комп'ютерного зору (Computer Vision, CV), включаючи розпізнавання образів. Ці методи дозволяють системам "вчитися" безпосередньо з даних, автоматично виділяючи складні ознаки, що раніше вимагало значних ручних зусиль.

1.2.1 Методи машинного навчання у комп'ютерному зорі

Традиційні методи машинного навчання для комп'ютерного зору зазвичай вимагають попереднього виділення ознак (feature engineering) людиною-експертом. Після цього, ці ознаки подаються на вхід класифікатора. До таких методів належать:

1. Метод опорних векторів (Support Vector Machines, SVM): Це потужний алгоритм для бінарної та багатокласової класифікації, що знаходить оптимальну розділяючу гіперплощину між класами в багатовимірному просторі ознак. SVM ефективні при роботі з високовимірними даними та можуть використовувати різні функції ядра для обробки нелінійно розділених даних.

2. Дерева рішень (Decision Trees) та Випадковий ліс (Random Forest): Дерева рішень будують модель, яка послідовно приймає рішення на основі значень ознак, розбиваючи дані на менші підмножини. Випадковий ліс є ансамблевим методом, який будує велику кількість дерев рішень і усереднює їхні прогнози, що значно покращує точність та зменшує перенавчання.

3. k-найближчих сусідів (k-Nearest Neighbors, k-NN): Це простий непараметричний алгоритм класифікації, який класифікує об'єкт на основі більшості голосів його k-найближчих сусідів у просторі ознак. Клас об'єкта визначається найпоширенішим класом серед його сусідів.

4. Методи кластеризації (наприклад, K-Means): Хоча це не класифікація, кластеризація може бути використана для попередньої обробки даних або сегментації зображень, групуючи схожі пікселі або регіони за певними ознаками.

5. Методи виділення ознак (Feature Descriptors): До традиційних методів виділення ознак відносяться SIFT (Scale-Invariant Feature Transform), SURF (Speeded Up Robust Features), HOG (Histogram of Oriented Gradients). Ці дескриптори дозволяють отримати інваріантні до масштабу, повороту та освітлення ознаки, які потім подаються на вхід класифікаторам ML.

Традиційні методи машинного навчання, на відміну від глибинного навчання, вимагають менших обчислювальних ресурсів для простих завдань і можуть ефективно навчатися на менших обсягах даних. Однак їхня головна слабкість полягає у значній залежності від якості ручного виділення ознак. Це робить їх малоефективними при роботі зі складними, неструктурованими даними та зображеннями з високою варіативністю.

1.2.2. Методи глибинного навчання у комп'ютерному зорі

Глибинне навчання є підмножиною машинного навчання, що базується на нейронних мережах з багатьма прихованими шарами, які здатні автоматично вивчати ієрархічні представлення даних. Це дозволяє моделям самостійно виділяти релевантні ознаки без ручного втручання.

Згорткові нейронні мережі (CNN) є фундаментом більшості сучасних рішень у комп'ютерному зорі, оскільки вони спеціально адаптовані для роботи з сітчастою топологією зображень. Їхня архітектура включає згорткові шари, які використовують фільтри для виявлення локальних ознак, таких як контури та текстури. За ними йдуть шари активації, що застосовують нелінійні функції (наприклад, ReLU) для вивчення складних взаємозв'язків у даних. Пулінгові шари зменшують розмірність карт ознак, зберігаючи ключову інформацію, що

допомагає знизити обчислювальну складність та запобігти перенавчанню. На завершальних етапах, повнозв'язні шари перетворюють витягнуті ознаки у фінальний вихід, наприклад, ймовірності класів. За роки розвитку було створено безліч успішних архітектур CNN, таких як LeNet, AlexNet, VGG, GoogLeNet (Inception), ResNet, DenseNet та EfficientNet, кожна з яких має свої унікальні особливості та оптимізації.

Рекурентні нейронні мережі (RNN), хоча й використовуються переважно для послідовних даних (як-от текст або мова), мають варіанти (наприклад, LSTM, GRU), які знаходять застосування в комп'ютерному зорі. Вони є корисними для завдань, що вимагають розуміння часових послідовностей у відео, наприклад, для розпізнавання дій.

Автоенкодери та генеративно-змагальні мережі (GANs) використовуються для генерації зображень, зменшення розмірності даних, а також для розширення наборів даних (data augmentation), що є особливо цінним при обмежених обсягах даних для навчання.

Переваги методів глибокого навчання (DL) полягають у їхній здатності до автоматичного виділення ознак, що значно спрощує процес розробки та підвищує адаптивність системи, усуваючи потребу в ручному інженерінгу ознак. Ці методи досягають надзвичайно високої точності на складних задачах розпізнавання та є масштабованими, ефективно працюючи з великими обсягами даних.

Однак, методи DL мають і свої недоліки. Вони вимагають значних обсягів маркованих даних для ефективного навчання, а також потужних обчислювальних ресурсів (GPU) для тренування складних моделей. Крім того, ці мережі часто діють як "чорний ящик", що ускладнює інтерпретацію того, чому вони прийняли те чи інше рішення. У контексті розпізнавання дорожніх знаків, CNN є найбільш підходящим та ефективним методом завдяки їхній здатності обробляти просторові ієрархії ознак, що є ключовим для візуальних даних. Подальші розділи роботи зосередяться на виборі та адаптації конкретної архітектури CNN для вирішення поставленого завдання.

1.3 Аналіз популярних архітектур нейронних мереж для розпізнавання об'єктів

Після огляду загальних методів глибокого навчання, важливо зосередитись на конкретних архітектурах згорткових нейронних мереж (CNN), які зарекомендували себе як ефективні рішення для задач розпізнавання об'єктів, що є центральною проблемою при розпізнаванні дорожніх знаків. Кожна з цих архітектур має свої особливості, переваги та недоліки, що робить їх придатними для різних умов застосування (рис. 1.1).

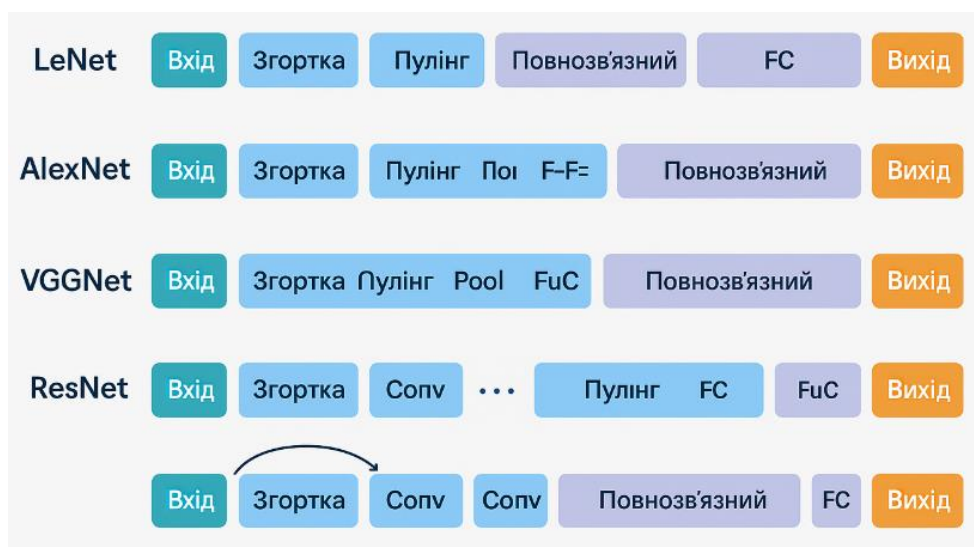


Рисунок 1.1 - Типологія архітектур глибокого навчання

Наразі широко використовуються популярні архітектури згорткових нейронних мереж (CNN), які використовуються для розпізнавання об'єктів. На зображенні представлені типові шари: вхідне зображення, згорткові шари (convolutional layers), шари активації, pooling-шари, повнозв'язні (fully connected) шари та вихідний шар класифікації. Показано послідовну обробку зображення, де на кожному етапі мережа витягує дедалі складніші ознаки для точної ідентифікації об'єктів. Такі архітектури лежать в основі моделей як VGG, ResNet або YOLO.

1.3.1 ResNet (Residual Networks)

ResNet (Residual Networks) — це архітектура згорткових нейронних мереж, розроблена Microsoft Research у 2015 році, яка успішно подолала проблеми "зникаючого градієнта" та "деградації", що виникали при навчанні надзвичайно

глибоких нейронних мереж. Основна інновація ResNet полягає у використанні "залишкових блоків" або "блоків пропуску" (skip connections/shortcut connections). Ці блоки дозволяють градієнтам безперешкодно проходити через кілька шарів, обходячи деякі з них, що значно підвищує ефективність навчання дуже глибоких архітектур. Принцип роботи полягає в тому, що замість прямого вивчення цільового відображення $H(x)$, блок навчає функцію залишку $F(x)=H(x)-x$, а потім додає її до вхідного значення x , тобто вихід блоку становить $F(x)+x$.

ResNet-архітектури ідеально підходять для завдань класифікації зображень і слугують потужною базовою мережею (backbone) для багатьох систем виявлення об'єктів та сегментації. Існують різні варіанти ResNet, такі як ResNet-18, ResNet-34, ResNet-50, ResNet-101 та ResNet-152, які відрізняються кількістю шарів, що впливає на їхню обчислювальну складність та результуючу точність.

1.3.2 YOLO (You Only Look Once)

YOLO (You Only Look Once) — це родина архітектур, створених для виявлення об'єктів у реальному часі. На відміну від двохетапних систем, таких як R-CNN або Faster R-CNN, які спочатку пропонують потенційні області, а потім класифікують їх, YOLO виконує виявлення та класифікацію одночасно в межах однієї нейронної мережі. Цей підхід передбачає поділ зображення на сітку, де кожна клітинка сітки відповідає за прогнозування обмежувальних рамок, їхніх оцінок впевненості та ймовірностей класів об'єктів.

YOLO ідеально підходить для завдань, де критично важлива висока швидкість обробки, як-от у відеоаналітиці, автономному водінні (де швидке розпізнавання дорожніх знаків є життєво важливим) та робототехніці. Існує багато версій YOLO, включаючи YOLOv1, YOLOv2 (YOLO9000), YOLOv3, YOLOv4, YOLOv5, YOLOv6, YOLOv7 та YOLOv8, кожна з яких приносить покращення як у швидкості, так і в точності.

1.3.3 MobileNet

MobileNet — це сімейство легких згорткових нейронних мереж, створених Google спеціально для мобільних та вбудованих пристроїв з обмеженими обчислювальними ресурсами.

Головна інновація MobileNet полягає у застосуванні "роздільних згорток за глибиною" (depthwise separable convolutions). Ця техніка розбиває стандартну згортку на два окремі етапи: спочатку відбувається глибинна згортка (depthwise convolution), яка застосовує по одному фільтру до кожного вхідного каналу окремо; потім виконується точкова згортка (pointwise convolution), що використовує згортку 1×1 для об'єднання вихідних даних з попереднього кроку. Такий підхід суттєво зменшує кількість параметрів та обчислювальних операцій порівняно зі стандартними згортками, зберігаючи при цьому високу точність. Існують різні версії цієї архітектури, зокрема MobileNetV1, MobileNetV2 та MobileNetV3.

Завдяки своїй ефективності, MobileNet ідеально підходить для розгортання на мобільних телефонах, IoT-пристроях, мікроконтролерах та інших платформах, де існують жорсткі обмеження щодо обчислювальної потужності, пам'яті та енергоспоживання. Це робить їх надзвичайно привабливими для вбудованих систем розпізнавання дорожніх знаків у транспортних засобах.

Переваги та недоліки вказаних архітектур наведено в табл. 1.1. проведений аналіз демонструє, що різні архітектури згорткових нейронних мереж мають свої сильні та слабкі сторони при розпізнаванні дорожніх знаків. ResNet вирізняється здатністю ефективно навчати дуже глибокі мережі та високою точністю, але вимагає значних обчислювальних ресурсів. YOLO ідеально підходить для розпізнавання в реальному часі завдяки своїй високій швидкості, хоча може мати труднощі з малими об'єктами або поступатися у точності деяким іншим детекторам. MobileNet, зі свого боку, є надзвичайно ефективним для мобільних та вбудованих пристроїв завдяки низькій обчислювальній складності, забезпечуючи при цьому прийнятну точність, хоча й дещо нижчу порівняно зі складнішими моделями. Таким чином, вибір архітектури залежить від пріоритетів проекту: чи то максимальна точність, чи то швидкість в реальному часі, чи то ефективність на обмежених ресурсах.

Таблиця 1.1 Аналіз архітектур згорткових нейронних мереж

<i>Архітектура</i>	<i>Переваги при розпізнаванні дорожніх знаків</i>	<i>Недоліки при розпізнаванні дорожніх знаків</i>
ResNet	- Здатність ефективно навчати дуже глибокі мережі, що дозволяє вивчати складні ознаки дорожніх знаків. - Висока точність класифікації та розпізнавання об'єктів, що критично для надійної ідентифікації знаків. - Широке використання для трансферного навчання, дозволяючи швидко адаптувати модель до нових наборів даних дорожніх знаків.	- Відносно велика кількість параметрів, що вимагає значних обчислювальних ресурсів та пам'яті для навчання та інференсу, що може бути проблемою для систем реального часу на обмежених пристроях. - Може бути надмірною для задач, де потрібна висока швидкість інференсу на пристроях з обмеженими ресурсами (наприклад, для вбудованих систем у автомобілях).
YOLO	- Висока швидкість інференсу, що робить її ідеальною для розпізнавання дорожніх знаків у реальному часі (наприклад, для автономного водіння). - Одночасна обробка всього зображення за один прохід, зменшуючи ймовірність хибних спрацьовувань фонових елементів. - Хороша узагальнююча здатність, дозволяючи розпізнавати знаки в різних умовах.	- У старих версіях можлива нижча точність порівняно з деякими двохетапними детекторами (хоча у нових версіях цей розрив значно скорочується). - Має труднощі з виявленням дуже малих об'єктів, особливо коли вони згруповані разом, що актуально для далеких дорожніх знаків. - Обмежена кількість об'єктів, які може передбачити одна клітинка сітки, що може бути проблемою при великому скупченні знаків.
MobileNet	- Мала кількість параметрів та низька обчислювальна складність, що робить її ідеальною для розгортання на мобільних, вбудованих та IoT-пристроях з обмеженими ресурсами. - Висока ефективність для застосування в бортових системах розпізнавання дорожніх знаків. - Прийнятна точність для багатьох практичних задач, що дозволяє досягти балансу між продуктивністю та якістю.	- Дещо нижча точність порівняно з більшими та складнішими мережами, такими як ResNet або повнорозмірні YOLO, що може бути критичним для деяких високоточних застосувань. - Може вимагати більшої кількості даних або більш ретельної оптимізації для досягнення високої точності на складних завданнях розпізнавання дорожніх знаків у різноманітних умовах.

Вибір архітектури для розпізнавання дорожніх знаків визначається пріоритетами проекту. Якщо головною метою є висока точність класифікації, а

обчислювальні ресурси не є обмеженням, ResNet може слугувати потужною основою для класифікатора. У випадку, коли критичною є швидкість розпізнавання в реальному часі та виявлення об'єктів у відеопотоці, YOLO (особливо його останні версії) стає чудовим вибором, оскільки дозволяє не тільки класифікувати знак, а й визначити його місце розташування. Якщо система призначена для вбудованих пристроїв з обмеженими ресурсами, таких як бортові комп'ютери автомобілів, MobileNet пропонує збалансований компроміс між точністю та обчислювальною ефективністю.

У контексті створення комплексної системи розпізнавання дорожніх знаків також можлива комбінація цих підходів, наприклад, інтеграція MobileNet як легкої базової мережі для детектора типу YOLO, або використання ResNet для більш точної класифікації вже виявлених знаків. У цій роботі буде обґрунтовано вибір архітектури, яка найкраще відповідає поставленим завданням з урахуванням необхідної точності та продуктивності.

1.4 Сучасні тенденції розвитку візуального розпізнавання

Візуальне розпізнавання є однією з найбільш динамічних галузей штучного інтелекту, де нові прориви та тенденції з'являються з надзвичайною швидкістю. Поряд з постійним вдосконаленням згорткових нейронних мереж, з'являються нові парадигми, що обіцяють ще більшу точність, ефективність та ширші можливості застосування, особливо актуальні для складних задач, таких як розпізнавання дорожніх знаків в реальному часі.

1.4.1. Vision Transformers (ViT)

Традиційно згорткові нейронні мережі (CNN) домінували у комп'ютерному зорі завдяки їхній здатності ефективно обробляти просторові ієрархії ознак. Однак, з 2020 року значний інтерес викликають **Vision Transformers (ViT)**, які адаптують архітектуру трансформерів, що раніше успішно застосовувалася у обробці природної мови (NLP), для завдань комп'ютерного зору.

Принцип роботи. Замість застосування згортки, ViT поділяють зображення на фіксовані за розміром "патчі" (patches) або фрагменти. Кожен патч

перетворюється на лінійну ембедінг, а потім до цих ембедінгів додаються позиційні ембедінги. Отримані послідовності патчів подаються на вхід стандартному енкодеру трансформера, який використовує механізм самостійної уваги (self-attention) для виявлення взаємозв'язків між різними частинами зображення.

Для розпізнавання дорожніх знаків, ViT можуть запропонувати краще розуміння загальної сцени та взаємозв'язків між знаками та навколишнім середовищем, особливо у складних дорожніх ситуаціях.

1.4.2. Мультимодальний AI (Multimodal AI)

Мультимодальний AI зосереджений на створенні моделей, які можуть обробляти та інтегрувати інформацію з декількох різних модальностей даних, таких як візуальна інформація (зображення/відео), текстова інформація, аудіо, дані з сенсорів (наприклад, радарів, лідарів).

Принцип роботи. Мультимодальні моделі навчаються знаходити взаємозв'язки та спільні представлення між різними типами даних. Наприклад, модель може одночасно аналізувати зображення дорожнього знака та текстові підказки з навігаційної системи або звукові попередження.

Для СРДЗ, мультимодальний підхід може включати використання не тільки камер (візуальні дані), але й GPS-даних, карт, інформації з CAN-шини автомобіля або навіть голосових команд водія, що підвищить точність та контекстуальну обізнаність.

1.4.3. Edge AI (Штучний інтелект на периферії)

Edge AI – це парадигма, де обробка даних та виконання моделей штучного інтелекту відбувається безпосередньо на периферійних пристроях (наприклад, у бортовому комп'ютері автомобіля, смартфонах, IoT-пристроях), а не у хмарі або централізованому сервері.

Принцип роботи. Моделі машинного навчання оптимізуються та стискаються (наприклад, квантування, прунінг, знання дистиляції), щоб ефективно працювати на апаратному забезпеченні з обмеженими обчислювальними ресурсами, пам'яттю та енергоспоживанням.

Для розпізнавання дорожніх знаків у автомобілях, Edge AI є обов'язковою вимогою, оскільки рішення про безпеку мають прийматися миттєво, без затримок, спричинених передачею даних до віддалених серверів. MobileNet, про які йшлося раніше, є яскравим прикладом архітектур, оптимізованих для Edge AI.

1.4.4. Самонавчання (Self-Supervised Learning) та Напівконтрольоване навчання (Semi-Supervised Learning)

Традиційне навчання глибоких нейронних мереж вимагає величезних обсягів маркованих даних, що є дорогим і трудомістким процесом. Самонавчання та напівконтрольоване навчання є підходами, спрямованими на зменшення залежності від ручного маркування. *Принцип роботи:*

- *Самонавчання.* Модель навчається на немаркованих даних, генеруючи власні "псевдо-мітки" з даних. Наприклад, модель може вчитися прогнозувати відсутні частини зображення, повертати повернуте зображення у правильну орієнтацію, або виділяти ознаки, вирішуючи деякі допоміжні завдання без явних людських міток. Після цього, навчена таким чином модель може бути доналаштована на невеликому обсязі маркованих даних для основної задачі.
- *Напівконтрольоване навчання.* Комбінує невелику кількість маркованих даних з великим обсягом немаркованих. Модель навчається на маркованих даних, а потім використовує знання з цього навчання для отримання "псевдо-міток" для немаркованих даних, розширюючи свій навчальний набір.

Наведемо аналіз сучасних тенденцій у візуальному розпізнаванні (табл. 1.2), розкриваючи переваги та недоліки чотирьох ключових напрямків: Vision Transformers (ViT), Мультимодальний AI, Edge AI та Самонавчання / Напівконтрольоване навчання.

Vision Transformers (ViT) пропонують новий підхід до комп'ютерного зору, відходячи від традиційних згорток CNN на користь механізму самостійної уваги. Їхня головна перевага полягає у здатності до глобального розуміння контексту зображення та відмінній масштабованості на великих наборах даних, що робить їх перспективними для складних дорожніх сценаріїв. Однак, вони мають високу обчислювальну вартість, потребують значних обсягів даних для

попереднього навчання та не мають вбудованої інваріантності до зсувів, на відміну від CNN.

Таблиця 1.2 Аналіз систем візуального розпізнавання об'єктів.

<i>Технологія / Архітектура</i>	<i>Переваги</i>	<i>Недоліки</i>
<i>Vision Transformers (ViT)</i>	- Глобальне розуміння контексту: механізм самостійної уваги дозволяє захоплювати взаємозв'язки між віддаленими частинами зображення. - Масштабованість: Демонструють високу продуктивність при масштабуванні на великих наборах даних. - Уніфікація архітектур: дозволяють використовувати єдину архітектуру для різних завдань ШІ.	- Висока обчислювальна здатність: є обчислювально витратними для зображень високої роздільної здатності. - Потреба у великих даних: для досягнення високої вимагають значно більших наборів даних для попереднього навчання. - Відсутність вбудованої інваріантності: не мають вбудованої інваріантності до зсувів та масштабу, що може потребувати більших обсягів даних.
<i>Мультимодальний AI</i>	- Комбінування інформації з різних джерел: візуальні дані, текст, аудіо, дані сенсорів. - Підвищення стійкості: якщо одна модальність є невизначеною або недоступною. - Збагачений користувацький досвід: дозволяє створювати більш інтерактивні та інтуїтивно зрозумілі інтерфейси.	- Складність інтеграції: Вимагає складних механізмів для обробки та інтеграції різнорідних даних.
<i>Edge AI (Штучний інтелект на периферії)</i>	- Низька затримка (Low Latency): обробка даних відбувається миттєво без потреби передачі до хмари, що критично для систем реального часу. - Підвищена конфіденційність: дані користувача не залишають пристрій. - Менша залежність від мережі: функціональність зберігається навіть без стабільного підключення до Інтернету. - Зниження витрат: менші витрати на передачу даних та хмарні обчислення.	- Обмежені обчислювальні ресурси: вимагає оптимізації та стиснення моделей для ефективної роботи на пристроях з обмеженою потужністю, пам'яттю та енергоспоживанням.
<i>Самонавчання (Self-Supervised Learning) та/або Напівконтрольоване навчання (Semi-Supervised Learning)</i>	- Зменшення потреби у маркованих даних: знижують витрати та час, пов'язані з ручним маркуванням великих наборів даних. - Використання великих немаркованих наборів даних: дозволяють використовувати великі обсяги доступних немаркованих даних для попереднього навчання моделей. - Покращена узагальнююча здатність: моделі можуть краще узагальнювати на нові, невидимі дані.	- Складність допоміжних завдань: розробка ефективних допоміжних завдань для самонавчання може бути нетривіальною. - Якість псевдо-міток: Якість "псевдо-міток" у напівконтрольованому навчанні може впливати на кінцеву точність моделі.

Мультимодальний AI зосереджений на інтеграції інформації з різних джерел (візуальні дані, текст, сенсори). Це дозволяє системі отримати більш повне та надійне розуміння ситуації, підвищуючи стійкість у випадках, коли одна модальність є невизначеною. Такий підхід є особливо цінним для автономного водіння, де комбінація камер, GPS та даних з CAN-шини може значно покращити контекстуальну обізнаність.

Edge AI (Штучний інтелект на периферії) є критично важливим для систем розпізнавання дорожніх знаків в автомобілях, де рішення мають прийматися миттєво. Його переваги включають низьку затримку, підвищену конфіденційність, меншу залежність від мережі та зниження витрат. Однак, реалізація Edge AI вимагає оптимізації моделей для роботи на пристроях з обмеженими обчислювальними ресурсами.

Самонавчання та напівконтрольоване навчання пропонують ефективні рішення для зменшення залежності від великих обсягів маркованих даних, що є дорогим і трудомістким процесом. Ці підходи дозволяють використовувати великі немарковані набори даних для попереднього навчання, підвищуючи узагальнюючу здатність моделей. Це є значною перевагою для розширення можливостей систем розпізнавання дорожніх знаків без необхідності інтенсивного ручного маркування.

В цілому, аналіз показує, що кожна з цих тенденцій пропонує унікальні можливості для підвищення ефективності систем розпізнавання дорожніх знаків, будь то через покращення розуміння контексту (ViT, Мультимодальний AI), оптимізацію для вбудованих систем (Edge AI) або зменшення залежності від маркованих даних (Самонавчання). Комбінація цих підходів може призвести до створення більш надійних, ефективних та адаптивних систем розпізнавання дорожніх знаків. Для СРДЗ, де збір та маркування мільйонів зображень дорожніх знаків у різних умовах є складним завданням, методи самонавчання та напівконтрольованого навчання пропонують перспективний шлях для створення більш надійних та універсальних моделей за менших витрат ресурсів. Такі тенденції демонструють, що візуальне розпізнавання продовжує розвиватися,

пропонуючи все більш потужні та ефективні рішення для широкого спектру задач, включаючи критично важливі для безпеки дорожнього руху системи автоматичного розпізнавання дорожніх знаків.

1.5 Аналіз існуючих систем візуального розпізнавання об'єктів

Система ідентифікації дорожніх знаків інтегрує апаратні засоби, що включають камеру та модуль захоплення, з програмними компонентами, такими як модулі виявлення та класифікації, а також базу даних. Процес роботи починається з фіксації зображення камерою, після чого система обробляє його для визначення місцезнаходження дорожнього знака за допомогою відповідних алгоритмів. Наступним кроком модуль розпізнавання ідентифікує тип знака. Уся необхідна для функціонування системи інформація зберігається в базі даних, структура якої залежить від конкретного призначення системи. Завдяки цьому послідовному процесу, система ефективно розпізнає дорожні знаки, забезпечуючи водія точною інформацією та сприяючи загальному підвищенню безпеки дорожнього руху.

1.5.1 Система розпізнавання дорожніх знаків Volvo IntelliSafe

Система Volvo IntelliSafe — це комплексний інтегрований набір технологій безпеки від Volvo, однією з ключових функцій якого є розпізнавання дорожніх знаків. Для цього система використовує фронтальну камеру автомобіля, яка безперервно моніторить дорожню обстановку. Камера аналізує зображення дороги, виявляючи та ідентифікуючи різноманітні дорожні знаки, такі як обмеження швидкості, знаки "Стоп" або заборонні знаки. Отримана після успішного розпізнавання інформація негайно відображається на спеціальному екрані в салоні автомобіля, забезпечуючи водія актуальними даними про виявлені знаки. Принцип роботи системи Volvo IntelliSafe візуалізовано на рис. 1.2.

Однак, система розпізнавання дорожніх знаків, зокрема в Volvo IntelliSafe, має певні недоліки. Її ефективність значною мірою залежить від стану самих дорожніх знаків: погано видимі або пошкоджені знаки можуть призвести до помилкового розпізнавання або його повної відсутності. Ця проблема

посилюється в складних погодних умовах, за наявності відблисків або через неналежне обслуговування дорожньої інфраструктури.



Рисунок 1.2 – Візуалізація роботи системи розпізнавання дорожніх знаків
Volvo IntelliSafe

Крім того, система, як правило, розпізнає лише стандартно розміщені та оформлені знаки, що обмежує її можливості щодо ідентифікації тимчасових або нестандартних знаків. Важливо також враховувати, що система не завжди повною мірою аналізує контекст дорожньої ситуації або інші фактори, які можуть впливати на правильну інтерпретацію знаків. З огляду на це, водії завжди повинні зберігати пильність та приймати рішення, керуючись актуальними дорожніми умовами.

1.5.2 Система Audi Traffic Sign Recognition

Audi Traffic Sign Recognition (Розпізнавання дорожніх знаків Audi) — це провідна система допомоги водію, розроблена компанією Audi з метою суттєвого підвищення інформованості водія та безпеки на дорозі. Її функціонування забезпечується **фронтальною камерою автомобіля**, яка зазвичай розташована у верхній частині лобового скла. Ця камера безперервно сканує дорожню обстановку, фіксуючи та аналізуючи зображення.

Система здатна ідентифікувати широкий спектр дорожніх знаків, що зустрічаються на шляху. До них відносяться як постійні, так і тимчасові **обмеження швидкості**, разом з додатковими табличками, що уточнюють умови (наприклад, для вантажних автомобілів або в певні години). Також розпізнаються **заборонні знаки**, такі як "Стоп", "Вїзд заборонено", "Обгін заборонено" тощо, а також **попереджувальні знаки**, наприклад, "Небезпечний поворот" або "Пішохідний перехід".

Після успішного розпізнавання, отримана інформація про дорожній знак миттєво відображається на дисплеї віртуальної панелі приладів Audi (Audi virtual cockpit) або на екрані системи MMI (Multi Media Interface). Це забезпечує водія постійною та актуальною інформацією, дозволяючи йому швидко реагувати на зміни у правилах дорожнього руху. Окремі версії системи також можуть бути інтегровані з адаптивним круїз-контролем або іншими системами допомоги, надаючи рекомендації щодо швидкості або попередження у разі її перевищення. Принцип роботи системи Audi Traffic Sign Recognition візуалізовано на рисунку 1.3.



Рисунок 1.3 – Робота системи розпізнавання об'єктів
Audi Traffic Sign Recognition

Проте, як і інші системи розпізнавання дорожніх знаків на основі камер, Audi Traffic Sign Recognition має певні обмеження. Її ефективність може знижуватися за умов поганої видимості, спричиненої інтенсивними опадами (дощ, сніг), туманом, сильними відблисками сонця, брудом або обмерзанням на камері. Знаки, які частково закриті (наприклад, гілками дерев, снігом, іншими транспортними засобами) або пошкоджені (графіті, вицвіла фарба), можуть бути розпізнані некоректно або взагалі пропущені. Крім того, система може мати труднощі з розпізнаванням тимчасових, неофіційних або нестандартно розміщених знаків. Важливо також, що система переважно аналізує окремі знаки, а не всю дорожню ситуацію в цілому, що іноді може призвести до некоректних інтерпретацій у складних сценаріях. Незважаючи на ці обмеження, система Audi Traffic Sign Recognition є цінним інструментом, що значно покращує інформованість водія, але вона не замінює його власної пильності та відповідальності за дотримання правил дорожнього руху.

1.5.3 Система Bosch Road Sign Assist

Система Bosch Road Sign Assist — це високотехнологічне рішення для допомоги водію, розроблене компанією Bosch (рис. 1.4), що є одним зі світових лідерів у сфері автомобільних технологій. Її функціональність ґрунтується на використанні **високочутливих камер**, які стратегічно розміщені у передній частині транспортного засобу. Ці камери постійно сканують дорожнє полотно, фіксуючи та аналізуючи візуальні образи дорожніх знаків.

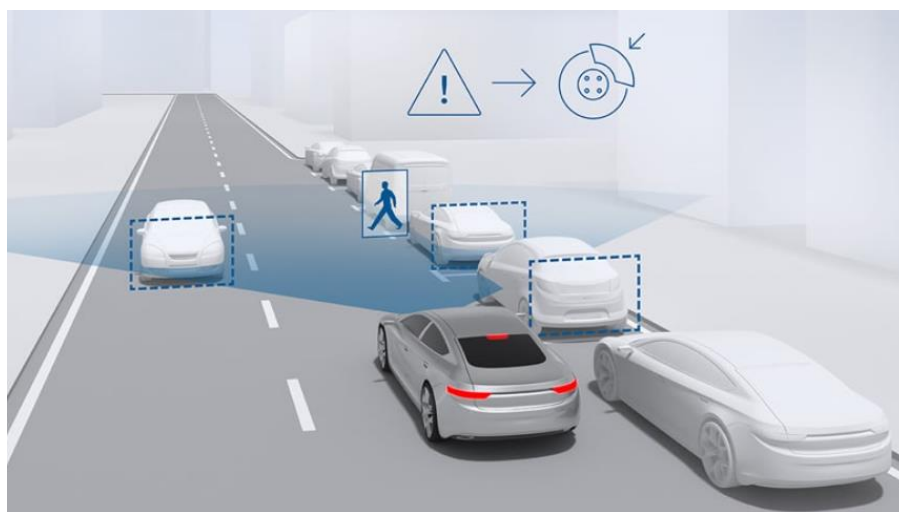


Рисунок 1.4 – Робота системи розпізнавання об'єктів Bosch Road Sign Assist

Система здатна виявляти та ідентифікувати широкий спектр регулюючих елементів дорожнього руху. До них належать як постійні, так і тимчасові **обмеження швидкості**, а також додаткові таблички, що уточнюють умови їх дії, наприклад, для певних типів транспортних засобів або у визначені часові проміжки. Також розпізнаються **знаки зупинки та пріоритету**, як-от "Стоп" та "Дати дорогу", **заборонні сигнали**, що забороняють обгін, в'їзд, розворот тощо, а також **попереджувальні та інформаційні знаки**, що сповіщають про небезпечні ділянки, напрямки руху або особливості дороги.

Після успішного розпізнавання, отримана інформація про дорожній знак оперативно відображається для водія у вигляді візуальних підказок на цифровому щитку приладів, на Head-up Display (проекційному дисплеї) або на інтегрованому дисплеї інформаційно-розважальної системи автомобіля. Така інтеграція дозволяє водієві залишатися в курсі актуальних дорожніх правил без необхідності постійно шукати знаки, що підвищує його інформованість та реакцію.

Технологічні особливості та інтеграція цієї системи полягають у тому, що Bosch Road Sign Assist не обмежується лише розпізнаванням знаків. Вона може інтегруватися з іншими системами допомоги водію (ADAS), наприклад, передаючи дані про обмеження швидкості до адаптивного круїз-контролю для автоматичного коригування швидкості автомобіля. У більш складних реалізаціях, система може використовувати не лише камеру, а й дані картографічних сервісів та GPS для підвищення точності та надійності розпізнавання, особливо в умовах поганої видимості або відсутності чітких знаків.

Незважаючи на високу ефективність, **Bosch Road Sign Assist, як і інші камерні системи, має певні обмеження**. Її чутливість до умов освітлення та погоди означає, що сильне сонячне світло, тіні, дощ, сніг або туман можуть ускладнити або унеможливити коректне розпізнавання знаків. Пошкоджені або закриті знаки, приховані рослинністю, іншими транспортними засобами, графіті або брудом, можуть бути пропущені або інтерпретовані неправильно. Система може мати труднощі з розпізнаванням нестандартних або тимчасових знаків, які не відповідають стандартним шаблонам. Крім того, якщо система комбінує

камеру з картографічними даними, застарілі дані на карті можуть призвести до відображення неактуальної інформації. Таким чином, Bosch Road Sign Assist є потужним інструментом для підвищення безпеки, але вона слугує лише допомогою, а не заміною пильності водія.

1.5.4 Порівняльний аналіз існуючих систем аналогів

Враховуючи проведений у попередніх розділах огляд існуючих систем розпізнавання дорожніх знаків, для наочного зіставлення їх сильних та слабких сторін, пропонується сформувати порівняльну таблицю 1.3.

Таблиця. 1.3 Порівняльний аналіз систем розпізнавання дорожніх знаків

<i>Характеристика</i>	<i>Volvo IntelliSafe (RSI)</i>	<i>Audi Traffic Sign Recognition (TSR)</i>	<i>Bosch Road Sign Assist (RSI)</i>
Основний принцип роботи	Використовує фронтальну камеру. У нових моделях може доповнюватися даними Google Maps.	Використовує фронтальну камеру. Інтегрується з MMI та Audi Virtual Cockpit.	Використовує високочутливі фронтальні камери. Може інтегруватися з картографічними даними та GPS.
Типи розпізнаваних знаків	Обмеження швидкості, "Стоп", заборони обгону, інші регулюючі знаки.	Обмеження швидкості (постійні/тимчасові), "Стоп", заборонні, попереджувальні.	Обмеження швидкості, "Стоп", пріоритету, заборонні, попереджувальні, інформаційні.
Відображення інформації	На панелі приладів (спідометр), Head-up Display.	На Audi Virtual Cockpit, MMI дисплеї.	На цифровому щитку приладів, Head-up Display, дисплеї інфотейнмент-системи.
Інтеграція з іншими системами	Може надавати попередження при перевищенні швидкості. У деяких моделях – дані з Google Maps.	Може інтегруватися з адаптивним круїз-контролем, надавати рекомендації/попередження.	Може інтегруватися з адаптивним круїз-контролем, даними карт та GPS.
Ключові обмеження	Залежність від видимості, пошкоджені/нестандартні знаки, потенційні помилки даних карт.	Залежність від видимості, пошкоджені/закриті/нестандартні знаки, відсутність повного контексту.	Чутливість до освітлення/погоди, пошкоджені/закриті/нестандартні знаки, потенційні помилки карт.
Основне призначення	Підвищення інформованості водія про поточні дорожні знаки.	Підвищення інформованості та допомога у дотриманні правил швидкості.	Комплексна допомога водію, підвищення безпеки та комфорту.

Усі три системи – Volvo IntelliSafe, Audi Traffic Sign Recognition та Bosch Road Sign Assist – активно застосовують **штучний інтелект**, зокрема методи комп'ютерного зору, що базуються на **глибокому навчанні (Deep Learning)**, для розпізнавання дорожніх знаків.

Процес починається з **виявлення об'єктів**, де штучний інтелект локалізує дорожні знаки на зображенні, отриманому з фронтальної камери. Сучасні системи для цього використовують згорткові нейронні мережі (CNN), подібні до архітектур YOLO, SSD або модифікацій Faster R-CNN. Ці мережі тренуються на величезних обсягах даних, що містять зображення дорожніх знаків за різних умов освітлення, погоди, ракурсів, розмірів та оклюзій, забезпечуючи їх точне виділення на складному фоні.

Наступним кроком є **класифікація об'єктів**. Після виявлення потенційних областей зі знаками, штучний інтелект, також за допомогою CNN, визначає конкретний тип знака, будь то обмеження швидкості, "Стоп" або "Дати дорогу". Для цього можуть застосовуватися більш глибокі CNN-архітектури, такі як ResNet, або оптимізовані для вбудованих систем MobileNet.

У відеопотоці, під час руху автомобіля, системи штучного інтелекту також використовують **алгоритми відстеження об'єктів** (наприклад, на основі фільтра Калмана, інтегрованого з результатами CNN). Це дозволяє підтримувати ідентифікацію знака, навіть якщо його видимість тимчасово погіршується, що підвищує надійність відображуваної інформації та усуває мерехтіння.

У більш складних реалізаціях, як у випадку з Bosch Road Sign Assist, що інтегрується з картами, штучний інтелект може застосовуватися для **фузії даних** з різних джерел, таких як камера, GPS та картографічні дані. Це дозволяє системі враховувати додатковий контекст, наприклад, коли дорожній знак закритий, але карта вказує на відповідне обмеження швидкості, або перевіряти достовірність розпізнаного знака за даними карти, що є елементом мультимодального штучного інтелекту.

Оскільки ці системи функціонують безпосередньо в автомобілі, вони є яскравим прикладом **Edge AI**. Це означає, що моделі штучного інтелекту

оптимізовані та стиснуті, використовуючи такі техніки, як квантування або прунінг, для ефективної роботи на апаратному забезпеченні з обмеженими обчислювальними ресурсами. Таке рішення забезпечує низьку затримку, що є критично важливим для систем безпеки.

Підсумовуючи, усі згадані системи (Volvo, Audi, Bosch) активно використовують глибокі згорткові нейронні мережі для базових функцій виявлення та класифікації дорожніх знаків. Відмінності між ними переважно полягають у деталях реалізації, конкретних архітектурах CNN, а також у рівні інтеграції з іншими сенсорами та системами, що впливає на підвищення точності та надійності в складних умовах. Сучасні розробки, такі як Vision Transformers або більш досконалі методи самонавчання, потенційно можуть бути інтегровані в майбутні версії цих систем для подальшого покращення їхньої продуктивності.

В результаті, було ґрунтовно розглянуто теоретичні засади систем візуального розпізнавання та машинного навчання, що є фундаментом для розробки системи розпізнавання дорожніх знаків. Розпочавши з визначення поняття візуального розпізнавання, було підкреслено його ключову роль у сучасних інтелектуальних системах, зокрема в автономному транспорті, робототехніці та системах безпеки. Визначено основні етапи процесу візуального розпізнавання, від захоплення зображення до інтерпретації отриманих даних.

Проведено огляд методів машинного та глибинного навчання, які є основою для комп'ютерного зору. Проаналізовано переваги традиційних методів машинного навчання, такі як менша обчислювальна складність для простих задач та менша потреба в обсягах даних, а також виявлено їхні недоліки, пов'язані з залежністю від ручного виділення ознак та низькою ефективністю при роботі зі складними, неструктурованими даними. На противагу цьому, розглянуто переваги методів глибокого навчання, зокрема автоматичне виділення ознак, високу точність та масштабованість, при цьому зазначено їхні недоліки, такі як потреба у великих обсягах даних та значних обчислювальних ресурсах, а також проблема "чорного ящика".

Особливу увагу приділено аналізу популярних архітектур нейронних мереж, які активно застосовуються для розпізнавання об'єктів. Детально описано згорткові нейронні мережі (CNN) як основу сучасних рішень, розглянуто їхні ключові компоненти (згорткові, активаційні, пулінгові та повнозв'язні шари). Проаналізовано такі архітектури, як ResNet, що вирішує проблему зникаючого градієнта в глибоких мережах, YOLO – для виявлення об'єктів у реальному часі з високою швидкістю, та MobileNet – оптимізовані для мобільних та вбудованих пристроїв. Для кожної архітектури були виділені її переваги та недоліки, що дозволяє обґрунтувати вибір для конкретних завдань розпізнавання дорожніх знаків.

Розглянуто сучасні тенденції розвитку візуального розпізнавання, включаючи Vision Transformers (ViT) з їхнім глобальним розумінням контексту, Мультимодальний AI, що забезпечує більш повне розуміння ситуації шляхом інтеграції даних з різних джерел, Edge AI для обробки даних на периферійних пристроях з низькою затримкою, а також самонавчання та напівконтрольоване навчання, що дозволяють зменшити залежність від маркованих даних.

Наведено аналіз існуючих комерційних систем розпізнавання дорожніх знаків, таких як Volvo IntelliSafe, Audi Traffic Sign Recognition та Bosch Road Sign Assist. Були порівняні їхні принципи роботи, можливості та обмеження. Визначено, що всі ці системи активно використовують штучний інтелект, зокрема глибоке навчання на етапах виявлення, класифікації та відстеження об'єктів, а також інтегрують дані з різних сенсорів та оптимізуються для роботи в умовах Edge AI.

Таким чином, проведене дослідження створило потужну теоретичну базу для подальшої розробки та реалізації системи автоматичного розпізнавання дорожніх знаків, визначивши ключові технології та підходи, які будуть використані у практичній частині роботи.

2 ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ СИСТЕМИ АВТОМАТИЧНОГО РОЗПІЗНАВАННЯ ДОРОЖНІХ ЗНАКІВ

2.1 Технічні вимоги до системи розпізнавання дорожніх знаків

Для успішної розробки та ефективного функціонування системи автоматичного розпізнавання дорожніх знаків необхідно чітко визначити технічні вимоги, яким вона повинна відповідати. Ці вимоги охоплюють як функціональні можливості, так і нефункціональні характеристики, що забезпечують її надійність, продуктивність та зручність використання.

Функціональні вимоги до системи розпізнавання дорожніх знаків передбачають забезпечення широкого спектру можливостей.

По-перше, система повинна підтримувати різноманітні вхідні дані, включаючи обробку статичних зображень у поширених форматах, таких як JPG, PNG та BMP. Важливою є також підтримка обробки відеопотоків у реальному часі, що може надходити з веб-камери або з відеофайлів форматів MP4 та AVI, з можливістю конфігурування джерела цього відеопотоку.

По-друге, ключовою функціональністю є виявлення дорожніх знаків. Система має бути здатною виявляти знаки різних типів та форм – круглі, трикутні, квадратні – згідно з національними стандартами, наприклад, ДСТУ 4100-2014 для України, або міжнародними, як Віденська конвенція про дорожні знаки і сигнали. Необхідною є також здатність виявляти знаки різного розміру та на різних відстанях від камери, а також ефективно ідентифікувати кілька знаків на одному зображенні або кадрі.

По-третє, система повинна забезпечувати точну класифікацію дорожніх знаків. Це означає точне віднесення виявлених об'єктів до визначених категорій дорожніх знаків, надання їх ідентифікатора класу та назви. Також має бути передбачена можливість розширення списку підтримуваних класів знаків у майбутньому для адаптації до нових вимог або стандартів.

По-четверте, важливою вимогою є стійкість системи до різних умов. Вона повинна розпізнавати знаки за умов змін освітлення, включаючи яскраве сонце,

тінь, сутінки та ніч. Необхідна також стійкість до різних погодних умов, таких як дощ, сніг або туман, за умови, що сам знак залишається достатньо видимим. Система має ефективно працювати при розпізнаванні знаків під різними кутами огляду та при частковій оклюзії, коли до 20-30% площі знака закрито іншими об'єктами. Крім того, передбачається обробка зображень з різним рівнем шуму та розмиттям, допустимим для людського сприйняття.

По-п'яте, система повинна забезпечувати інформативні вихідні дані та візуалізацію. Це включає відображення обмежувальних рамок (bounding boxes) навколо виявлених дорожніх знаків, а також надпису з назвою класу та ймовірністю розпізнавання для кожного виявленого знака. Має бути передбачена можливість збереження оброблених зображень або кадрів з нанесеними результатами розпізнавання, а також логування результатів, що включають тип знака, координати та ймовірність, у файл або консоль.

Нарешті, система повинна мати зручний інтерфейс користувача. Це передбачає простий та інтуїтивно зрозумілий графічний інтерфейс для завантаження зображень/відео та відображення результатів. Також має бути доступний консольний інтерфейс для автоматизованого тестування та виконання без графічного виводу, що забезпечить гнучкість у використанні системи.

Нефункціональні вимоги до системи розпізнавання дорожніх знаків визначають якісні характеристики її роботи.

По-перше, ключовою вимогою є **продуктивність**. Для відеопотоку в реальному часі система повинна забезпечувати швидкість обробки не менше 15 кадрів на секунду (FPS) на цільовому апаратному забезпеченні, зокрема з використанням GPU. Для статичних зображень час розпізнавання не повинен перевищувати 0.5 секунди на зображення. Крім того, критично важливою для систем реального часу є мінімальна **затримка (Latency)** між надходженням кадру та видачею результату розпізнавання.

По-друге, не менш важливою є **точність (Accuracy)**. Показник Mean Average Precision (mAP) для виявлення та класифікації дорожніх знаків має бути не менше 0.85 на тестовому наборі даних. Окрім того, показник Recall (Повнота)

для виявлення знаків повинен становити не менше 0.9, що свідчить про високу повноту виявлення всіх знаків.

По-третє, система повинна демонструвати високу **надійність**. Має коректно функціонувати протягом тривалого часу без збоїв, а також мати вбудовані механізми обробки помилок, таких як відсутність вхідних файлів, некоректний формат даних або помилки моделі. Крім того, необхідно забезпечити стабільність результатів розпізнавання для послідовних кадрів відео, щоб уникнути мерехтіння чи некоректних відображень.

По-четверте, важливою вимогою є **масштабованість** системи. Її архітектура повинна допускати можливість розширення функціоналу, наприклад, додавання нових типів знаків або інтеграції з іншими сенсорами. Також має бути передбачена можливість адаптації до роботи на різних апаратних платформах, включаючи потенційний перехід на вбудовані системи.

По-п'яте, система має бути **ефективною у використанні ресурсів**. Це означає оптимізацію споживання оперативної пам'яті та процесорного часу. Розмір моделі повинен залишатися розумним для потенційного розгортання на обмежених обчислювальних платформах.

По-шосте, важливим аспектом є **зручність використання**. Система повинна мати легку інсталяцію та налаштування програмного забезпечення, а також супроводжуватися чіткою документацією щодо її використання.

Дотримання цих технічних вимог дозволить створити функціональну, продуктивну та надійну систему автоматичного розпізнавання дорожніх знаків, яка матиме практичну цінність для подальшого впровадження у реальні транспортні рішення.

2.2 Обґрунтування вибору моделі YOLO та її особливостей

Вибір оптимальної архітектури нейронної мережі є критично важливим етапом при проектуванні системи автоматичного розпізнавання дорожніх знаків. Аналіз популярних архітектур, проведений у Розділі 1.3, показав, що різні моделі мають свої переваги та недоліки. Для вирішення поставленої задачі, що вимагає

високої точності та, що не менш важливо, здатності працювати в реальному часі, було прийнято рішення обґрунтувати вибір на користь сімейства архітектур **YOLO (You Only Look Once)**.

2.2.1 Обґрунтування вибору YOLO

Вибір моделі YOLO для розпізнавання дорожніх знаків зумовлений такими ключовими факторами:

1. **Розпізнавання в реальному часі.** Головна вимога до СРДЗ для використання в системах допомоги водію (ADAS) та автономному транспорті – це висока швидкість обробки відеопотоку. YOLO є одним з найшвидших алгоритмів виявлення об'єктів, здатним досягати десятків і навіть сотень кадрів на секунду (FPS), що робить його ідеальним для застосувань у реальному часі.

2. **Однопрохідне виявлення (Single-Shot Detector).** На відміну від двохетапних методів (наприклад, Faster R-CNN), які спочатку генерують регіони-кандидати, а потім класифікують їх, YOLO виконує виявлення та класифікацію об'єктів за один прохід нейронною мережею. Це значно зменшує обчислювальну складність та прискорює процес.

3. **Комбіноване виявлення та класифікація.** YOLO не тільки класифікує об'єкт, але й передбачає його місце розташування (обмежувальну рамку) на зображенні за один раз. Це дозволяє отримати повну інформацію про дорожній знак (його тип та координати) відразу після обробки кадру.

4. **Комплексне розуміння сцени.** YOLO розглядає все зображення одночасно, а не окремі регіони, що дозволяє моделі вивчати глобальний контекст. Це допомагає уникнути хибних спрацьовувань на фонових елементах та покращує точність у складних сценаріях.

5. **Доступність та підтримка.** Існують численні реалізації YOLO з відкритим вихідним кодом (наприклад, на PyTorch), велика спільнота розробників та постійне вдосконалення нових версій (YOLOv5, YOLOv6, YOLOv7, YOLOv8), що полегшує розробку та донавчання моделі.

Хоча YOLO може мати дещо нижчу точність порівняно з деякими двохетапними детекторами на дуже складних наборах даних (хоча цей розрив

постійно скорочується в нових версіях), його переваги у швидкості та ефективності роблять його оптимальним вибором для задачі розпізнавання дорожніх знаків у реальному часі.

2.2.2 Особливості архітектури YOLO.

Для реалізації системи розпізнавання дорожніх знаків буде використана одна з останніх і найбільш оптимізованих версій YOLO, а саме **YOLOv5**, яка поєднує високу точність з чудовою продуктивністю та гнучкістю. Ключові особливості YOLOv5, що роблять її ефективною для даної задачі, включають (рис. 2.1):

1. **Мережа Backbone (CSPDarknet53).** YOLOv5 використовує модифіковану архітектуру CSPDarknet53 як основну мережу для вилучення ознак із вхідного зображення. CSPNet (Cross Stage Partial Network) оптимізує згорткові операції, зменшуючи обчислювальну складність та збільшуючи точність за рахунок розділення базового шару на дві частини та об'єднання результатів. Це дозволяє ефективніше використовувати ресурси та покращувати градієнтний потік.

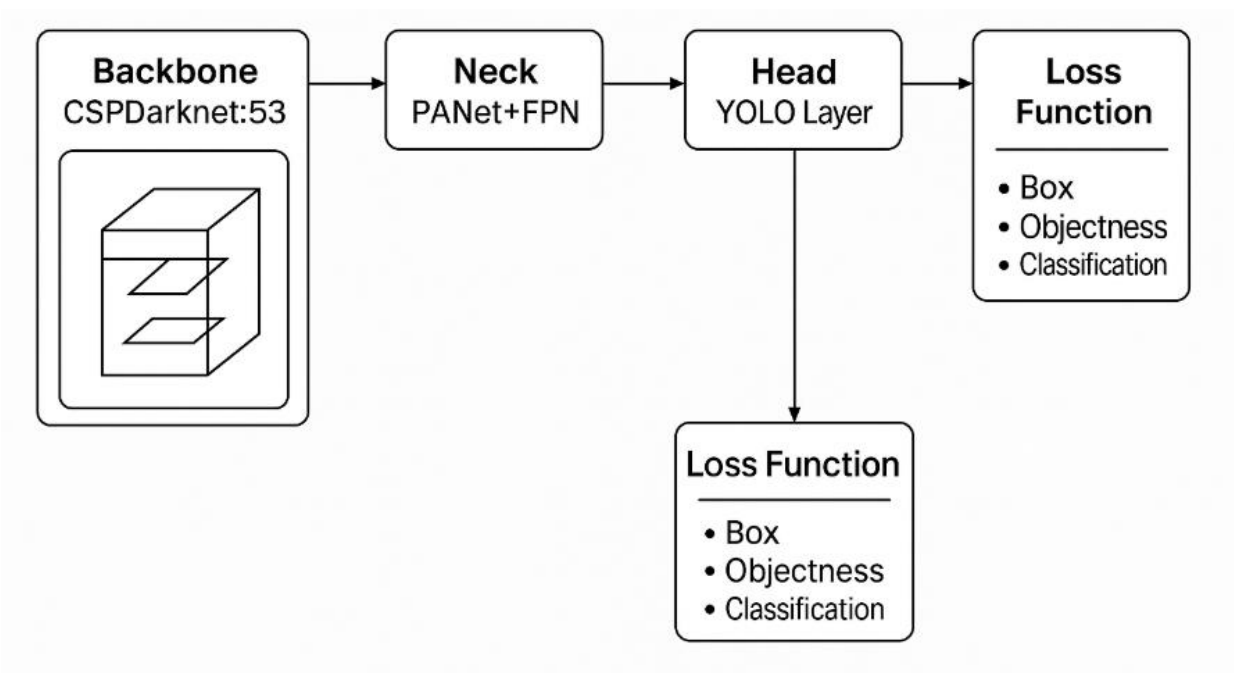


Рисунок 2.1 - Архітектура мережі YOLO

2. **Основа мережі (Neck - PANet + FPN).** Мережа відповідає за агрегацію ознак, отриманих з різних рівнів backbone. YOLOv5 використовує комбінацію Feature Pyramid Network (FPN) та Path Aggregation Network (PANet). FPN передає

сильну семантичну інформацію зверху вниз, тоді як PANet передає сильну позиційну інформацію знизу вгору. Це дозволяє моделі ефективно виявляти об'єкти різного масштабу, що є критично важливим для дорожніх знаків, які можуть з'являтися на зображенні як великими, так і дуже малими.

3. Виявлення (*Head - YOLO Layer*). Голова мережі відповідає за остаточні прогнози обмежувальних рамок, оцінок впевненості об'єктів та ймовірностей класів. YOLOv5, як і попередні версії YOLO, використовує якірні рамки (anchor boxes) для прогнозування. Для кожного масштабу виявлення (наприклад, для малих, середніх та великих об'єктів) використовується кілька якірних рамок, що допомагає моделі краще адаптуватися до форми та розміру об'єктів.

4. Функція втрат (*Loss Function*). YOLOv5 використовує комбінацію функцій втрат:

- *Box Loss (GIoU Loss)*. Для регресії обмежувальних рамок, яка не тільки штрафує за відстань між рамками, а й враховує перетин, що покращує стабільність навчання.
- *Objectness Loss (BCEWithLogitsLoss)*. Для прогнозування наявності об'єкта в клітинці сітки.
- *Classification Loss (BCEWithLogitsLoss)*. Для класифікації об'єктів за їхніми типами.

5. Оптимізації та техніки навчання. YOLOv5 включає ряд оптимізацій:

- *Автоматичне калібрування якірних рамок*. Можливість автоматично визначати найкращі якірні рамки для нового набору даних.
- *Data Augmentation*. Широке використання методів збільшення даних (наприклад, Mosaic, Copy-Paste, випадкові повороти, масштабування, зміна яскравості), що підвищує узагальнюючу здатність моделі.
- *Оптимізатори*. Використання оптимізаторів, таких як SGD або Adam, з техніками планування швидкості навчання (learning rate schedulers).

Завдяки цим особливостям, YOLOv5 дозволяє створити високоефективну систему розпізнавання дорожніх знаків, яка буде здатна точно та швидко виявляти та класифікувати об'єкти в умовах, близьких до реальних.

2.3 Загальна архітектура системи розпізнавання дорожніх знаків

Розробка ефективної системи автоматичного розпізнавання дорожніх знаків вимагає чіткої та модульної архітектури, яка дозволить інтегрувати різні компоненти та забезпечити їхню злагоджену роботу. Запропонована архітектура базується на принципах модульності та послідовної обробки даних, що дозволяє легко масштабувати систему та модифікувати окремі її частини. Загальна архітектура системи розпізнавання дорожніх знаків представлена на рис. 2.2.

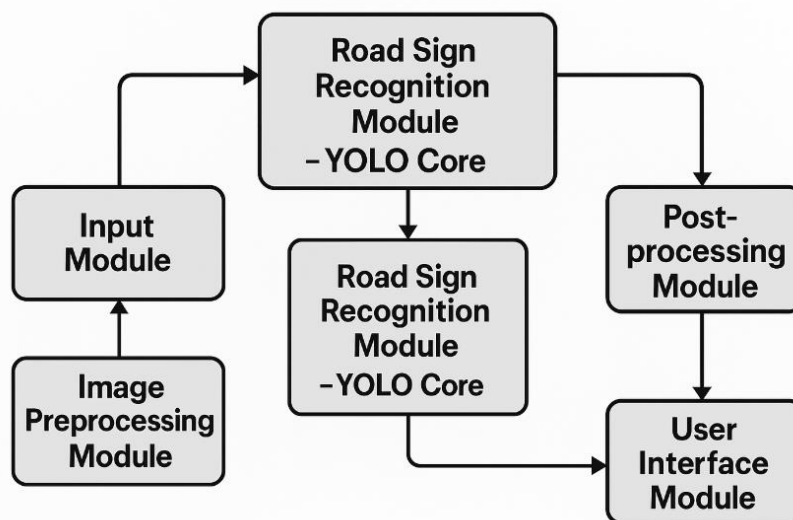


Рисунок 2.2 - Загальна архітектура системи розпізнавання дорожніх знаків

Модульна архітектура системи розпізнавання дорожніх знаків передбачає чітке розділення функціоналу на кілька взаємопов'язаних компонентів.

Початковим етапом є робота **Модуля Вхідних Даних (Input Module)**. Його відповідальність полягає в отриманні візуальної інформації, яка надалі буде оброблятися системою. Цей модуль забезпечує гнучкість застосування, підтримуючи різні джерела даних, такі як статичні зображення у форматах JPEG або PNG, а також відеопотоки, що можуть надходити з веб-камер, IP-камер або відеофайлів у форматах MP4 та AVI. Він надає стандартизований інтерфейс для доступу до вхідних кадрів, абстрагуючи подальші модулі від специфіки джерела даних.

Наступним кроком є **Модуль Попередньої Обробки Зображень (Image Preprocessing Module)**. Отримані візуальні дані часто містять шум, мають різне

освітлення, роздільну здатність або формат, тому цей модуль виконує підготовчі операції для оптимізації вхідних даних перед їх подачею до моделі машинного навчання. Це включає нормалізацію розміру зображення до стандартизованого для вхідної моделі (наприклад, 640×640 пікселів для YOLOv5), за необхідності перетворення кольорового простору (наприклад, в RGB) та нормалізацію значень пікселів до певного діапазону (наприклад, $[0, 1]$). Також може застосовуватися зменшення шуму за допомогою фільтрів. Хоча аугментація зображень здебільшого стосується етапу навчання, динамічна аугментація вхідних кадрів може бути застосована для стабільності розпізнавання, імітуючи реальні умови, такі як незначні зміни яскравості або контрасту.

Центральним компонентом системи є **Модуль Розпізнавання Дорожніх Знаків (Road Sign Recognition Module - YOLO Core)**, який виконує основну задачу виявлення та класифікації дорожніх знаків. Цей модуль інкапсулює завантажену та навчену модель YOLO (наприклад, YOLOv5) для інференсу. Він приймає попередньо оброблені зображення або кадри та генерує прогнози обмежувальних рамок (bounding boxes) для потенційних дорожніх знаків. Для кожної виявленої обмежувальної рамки модуль визначає ймовірність належності до кожного з класів дорожніх знаків. Після цього застосовується алгоритм немаксимального пригнічення (Non-Maximum Suppression, NMS) для фільтрації надлишкових обмежувальних рамок, залишаючи лише найкращі прогнози для кожного об'єкта.

Далі працює **Модуль Постобробки Результатів (Post-processing Module)**, який відповідає за обробку "сирих" результатів, отриманих від моделі YOLO, та підготовку їх для подальшого використання або візуалізації. Це включає фільтрацію виявлених об'єктів за порогом впевненості, відсіюючи ті, ймовірність яких нижча за встановлений поріг, що допомагає зменшити кількість хибних спрацьовувань. Також, за необхідності, виконується перетворення координат обмежувальних рамок до вихідного розміру зображення або інших систем координат, а результати розпізнавання формуються (наприклад, у вигляді

списку об'єктів, де кожен містить клас знака, його ймовірність та координати рамки).

Модуль Виведення та Візуалізації (Output & Visualization Module) відповідає за представлення результатів розпізнавання користувачеві або іншим системам. Він здійснює візуалізацію, наносячи обмежувальні рамки, назви класів та ймовірності на вихідне зображення або відеокادر. Також забезпечується виведення даних, включаючи збереження оброблених зображень/відео, логування результатів у текстовому файлі, або передачу структурованих даних розпізнавання до зовнішніх систем, таких як ADAS або навігаційна система.

Модуль Інтерфейсу Користувача (User Interface Module) забезпечує взаємодію користувача із системою. Він може пропонувати консольний інтерфейс для запуску системи з командного рядка, вказування вхідних файлів/потоків та отримання текстових логів результатів, що є зручним для автоматизованого тестування та інтеграції. Крім того, може бути передбачений графічний інтерфейс (GUI), що надає візуальне середовище для завантаження зображень/відео, відображення процесу розпізнавання в реальному часі, налаштування параметрів та перегляду результатів.

Така модульна архітектура дозволяє чітко розмежувати відповідальність компонентів, спростити розробку, тестування та подальше обслуговування системи розпізнавання дорожніх знаків.

2.4 Середовище розробки та використані інструменти в системі розпізнавання дорожніх знаків

Вибір відповідного середовища розробки та інструментарію є ключовим фактором для ефективної реалізації системи автоматичного розпізнавання дорожніх знаків. Для даного проєкту було обрано стек технологій, який забезпечує високу гнучкість, продуктивність, доступність широкої екосистеми бібліотек та активну підтримку спільноти. Основними інструментами є мова програмування Python та бібліотеки PyTorch і OpenCV.

Для реалізації системи розпізнавання дорожніх знаків було обрано стек технологій, що забезпечує ефективність, гнучкість та масштабованість рішення.

Основною мовою програмування є **Python**. Її вибір обумовлений статусом де-факто стандарту у галузі машинного та глибокого навчання завдяки великій кількості спеціалізованих бібліотек і фреймворків. Простота та висока читабельність коду Python прискорюють розробку та полегшують налагодження. Велика екосистема надає доступ до численних бібліотек для наукових обчислень (NumPy, SciPy), обробки даних (Pandas), візуалізації (Matplotlib, Seaborn), і що найважливіше, для машинного навчання (PyTorch, TensorFlow). Крос-платформність Python також забезпечує гнучкість розробки та розгортання на різних операційних системах. Усі модулі системи, включаючи вхідні дані, попередню обробку, інтеграцію моделі, постобробку та інтерфейс користувача, реалізовані саме на Python.

Серед фреймворків глибокого навчання обрано **PyTorch**, розроблений Facebook AI Research (FAIR), який є одним із провідних у цій галузі. Його ключові переваги полягають у гнучкості та використанні динамічного обчислювального графа (define-by-run), що дозволяє змінювати архітектуру мережі під час виконання та полегшує налагодження, що є особливо корисним при експериментуванні та донавчанні моделей. PyTorch надає інтуїтивно зрозумілий API, що робить його привабливим для дослідників та розробників. Активна спільнота та велика кількість навчальних матеріалів, попередньо навчених моделей та прикладів коду, включно з офіційними реалізаціями YOLOv5, значно прискорюють процес розробки. Крім того, PyTorch має вбудовану та ефективну підтримку обчислень на графічних процесорах (GPU) через CUDA, що критично важливо для прискорення навчання та інференсу глибоких нейронних мереж. У проєкті PyTorch використовується для завантаження та роботи з попередньо навченими моделями YOLO (зокрема YOLOv5), виконання прямого поширення для розпізнавання дорожніх знаків на зображеннях/відео, а також для потенційного донавчання моделі на власному наборі даних.

Для завдань комп'ютерного зору використовується бібліотека **OpenCV (Open Source Computer Vision Library)**, яка є крос-платформною та надає широкий спектр функцій. Її багатий функціонал включає тисячі оптимізованих алгоритмів для обробки зображень, відеоаналізу та розпізнавання об'єктів. Висока продуктивність, зумовлена реалізацією більшості функцій на C/C++ та їх оптимізацією, є важливою перевагою при роботі з відео в реальному часі. OpenCV легко інтегрується з Python. У проєкті вона використовується для завантаження та читання зображень/відео, ефективного зчитування кадрів з файлів або потоків з камери, для попередньої обробки зображень (операції зміни розміру, перетворення кольорового простору, базової корекції), візуалізації результатів (нанесення обмежувальних рамок, текстових міток) та для роботи з інтерфейсом, включаючи створення простого графічного інтерфейсу або відображення відео в реальному часі.

Додатково в проєкті застосовуються інші інструменти та середовища. **NumPy** є фундаментальною бібліотекою для наукових обчислень у Python, її використовують як PyTorch, так і OpenCV для роботи з багатовимірними масивами (тензорами). **Matplotlib** слугує для створення статичних, анімованих та інтерактивних візуалізацій, що може бути використано для відображення графіків навчання або аналізу даних. Для управління пакетами та середовищами, а також для ізоляції залежностей проєкту та запобігання конфліктам версій бібліотек, використовуються **Conda/Віртуальні середовища** (наприклад, Anaconda або venv). Щодо **апаратного забезпечення**, для ефективного навчання та інференсу глибоких нейронних мереж рекомендується використання графічного процесора (GPU) з підтримкою CUDA (наприклад, NVIDIA GPU), оскільки це значно прискорює обчислення порівняно з використанням тільки центрального процесора (CPU). Обрання такого стека технологій забезпечує надійну та гнучку основу для розробки системи розпізнавання дорожніх знаків, дозволяючи зосередитися на алгоритмічній частині та досягненні високої ефективності.

2.5 Процес навчання та донавчання моделі

Ефективність системи автоматичного розпізнавання дорожніх знаків безпосередньо залежить від якості навченої моделі машинного навчання. Процес навчання глибоких нейронних мереж, таких як YOLO, є ітеративним і вимагає ретельної підготовки даних, вибору гіперпараметрів та оптимізації. У даному підрозділі розглянуто теоретичні аспекти навчання моделі та особливості її донавчання для специфічної задачі розпізнавання дорожніх знаків.

2.5.1 Загальний процес навчання моделі

Навчання моделі глибокого навчання – це процес, під час якого модель адаптує свої внутрішні параметри (ваги та зміщення) таким чином, щоб мінімізувати помилку між її прогнозами та істинними значеннями. Даний процес включає наступні ключові етапи (рис. 2.3):

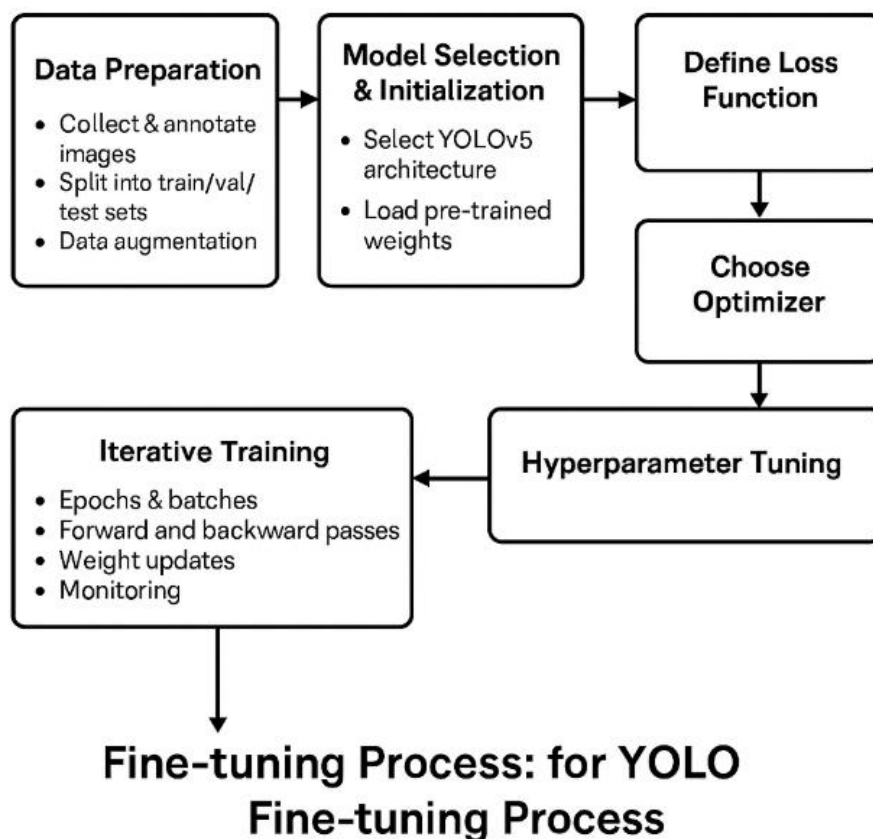


Рисунок 2.3 -. Загальний процес навчання моделі

1. Підготовка набору даних:

- **Збір даних.** Збір великого обсягу зображень, що містять дорожні знаки в різних умовах (різні погодні умови, освітлення, ракурси, відстані,

наявність перешкод). Важливо використовувати дані, які відображають реальні умови експлуатації системи.

- **Анотування даних.** Кожне зображення в наборі даних повинно бути ретельно анотоване. Для задачі виявлення об'єктів це означає малювання обмежувальних рамок навколо кожного дорожнього знака та присвоєння йому відповідного класу (як приклад, "обмеження швидкості 50", "стоп", "пішохідний перехід"). Якість анотації безпосередньо впливає на якість навчання моделі.
- **Розподіл даних.** Набір даних ділиться на три частини:
 - *Навчальний набір (Training Set):* Використовується для безпосереднього навчання моделі (близько 70-80% даних).
 - *Валідаційний набір (Validation Set).* Використовується для моніторингу прогресу навчання, налаштування гіперпараметрів та запобігання перенавчанню (близько 10-15% даних). Модель не "бачить" ці дані під час навчання.
 - *Тестовий набір (Test Set).* Використовується для фінальної, незалежної оцінки продуктивності навченої моделі. Модель не "бачить" ці дані ані під час навчання, ані під час валідації (близько 10-15% даних).
- **Аугментація даних (Data Augmentation).** Штучне збільшення розміру навчального набору даних шляхом застосування різних перетворень до існуючих зображень (наприклад, випадкові повороти, масштабування, обрізка, зміна яскравості/контрасту, дзеркальне відображення, мозаїка). Це допомагає моделі краще узагальнювати та підвищує її стійкість до варіацій у вхідних даних.

2. Вибір моделі та ініціалізація:

- Вибір архітектури нейронної мережі (у нашому випадку – YOLOv5).
- Зазвичай модель ініціалізується з *попередньо навченими вагами* на великих загальних наборах даних (наприклад, ImageNet, COCO). Це прискорює процес навчання та покращує початкову якість моделі, оскільки вона вже навчилася виділяти загальні візуальні ознаки.

3. **Визначення функції втрат (Loss Function).** Функція втрат кількісно оцінює відмінність між прогнозами моделі та істинними значеннями. Для YOLO вона, як правило, складається з трьох компонентів: втрати на класифікацію, втрати на довіру об'єкта та втрати на регресію обмежувальних рамок

4. **Вибір оптимізатора (Optimizer).** Оптимізатор – це алгоритм, який оновлює ваги моделі на основі градієнтів функції втрат. Популярними оптимізаторами є SGD (Stochastic Gradient Descent), Adam, AdamW, які допомагають мінімізувати функцію втрат

5. **Ітеративний процес навчання:**

- **Епохи (Epochs).** Одна епоха – це повний прохід по всьому навчальному набору даних.
- **Пакети (Batches).** Навчання відбувається не по одному зображенню, а за допомогою пакетів (batch) зображень, що дозволяє більш стабільно оцінювати градієнт.
- **Пряме та зворотне поширення (Forward and Backward Pass):**
 - **Пряме поширення.** Вхідні дані проходять через мережу, генеруючи прогнози.
 - **Зворотне поширення.** Обчислюються градієнти функції втрат відносно ваг мережі, використовуючи алгоритм зворотного поширення помилки.
- **Оновлення ваг.** Оптимізатор використовує обчислені градієнти для коригування ваг моделі, мінімізуючи функцію втрат.
- **Моніторинг.** Під час навчання постійно моніторяться метрики продуктивності (наприклад, функція втрат, точність, повнота, mAP) на валідаційному наборі даних для відстеження прогресу та виявлення перенавчання (overfitting).

6. **Налаштування гіперпараметрів (Hyperparameter Tuning).** Процес пошуку оптимальних значень для гіперпараметрів (швидкість навчання, розмір пакету, кількість епох, параметри оптимізатора), які не навчаються, а встановлюються до початку навчання.

2.5.2 Особливості донавчання (*Fine-tuning*) моделі

Донавчання є потужною технікою трансферного навчання (transfer learning), яка широко використовується для задач комп'ютерного зору, коли доступні обмежені обсяги цільових даних. Замість навчання моделі "з нуля", використовується модель, яка вже була попередньо навчена на великому, загальному наборі даних (наприклад, COCO, ImageNet).

Причини застосування донавчання у розробці систем розпізнавання дорожніх знаків є численними. Передусім, це дозволяє прискорити процес навчання, оскільки моделі вже засвоїли багато низькорівневих ознак, таких як контури та текстури, а також високорівневих концепцій. Таким чином, для нової, специфічної задачі потрібно значно менше часу на адаптацію. Крім того, донавчання **покращує продуктивність** моделі, оскільки попередньо навчені моделі зазвичай демонструють вищу ефективність порівняно з моделями, навченими "з нуля" на обмежених наборах даних, завдяки здатності уникнути перенавчання. Цей підхід ідеально підходить для сценаріїв з **обмеженими даними**, де збір великої кількості маркованих даних для специфічної задачі, як-от розпізнавання дорожніх знаків певного регіону, є складним і ресурсомістким завданням.

Методологія донавчання включає кілька етапів. Спочатку відбувається **завантаження попередньо навченої моделі**, наприклад, YOLOv5 з вагами, що були навчені на великому наборі даних COCO (Common Objects in Context) або аналогічному. Далі, необхідно **модифікувати вихідні шари** мережі. Оскільки кількість класів дорожніх знаків, які потрібно розпізнавати, відрізняється від класів, на яких навчалася оригінальна модель, останній (вихідний) шар мережі, або голова виявлення, адаптується відповідно до нової кількості класів. Після цього обирається **стратегія навчання**. Можна "заморозити" початкові шари мережі (backbone), які відповідають за виділення загальних ознак, використовуючи їх як ефективний екстрактор ознак, а навчання зосереджується лише на адаптації останніх шарів до нової задачі. Цей підхід, відомий як **заморожування шарів (Feature Extractor)**, часто застосовується при дуже малих

наборах даних. Альтернативно, можливе **донавчання всієї мережі**, коли ваги всіх шарів мережі оновлюються, але зі значно меншою швидкістю навчання (learning rate), ніж при навчанні з нуля. Це дозволяє моделі більш тонко адаптуватися до нового домену і є ефективнішим, якщо є достатньо даних, але їх обсяг менший, ніж для навчання з нуля. Фінальним кроком є **навчання на цільовому наборі даних**, підготовленому для дорожніх знаків, використовуючи обрану стратегію та оптимізовані гіперпараметри.

Застосування донавчання є ключовим для створення ефективної системи розпізнавання дорожніх знаків, оскільки воно дозволяє значно прискорити процес розробки та досягти високої точності, використовуючи вже наявні потужні моделі, що були навчені на величезних обсягах даних.

2.6 Реалізація компонентів системи розпізнавання дорожніх знаків

На основі обраної архітектури та інструментів (Python, PyTorch, OpenCV) буде розкрито функціональність кожного модуля та принципи їхньої взаємодії.

2.6.1 Модуль завантаження та попередньої обробки зображень

Модуль завантаження та попередньої обробки зображень є першим етапом у пайплайні системи, відповідальним за підготовку вхідних даних для моделі глибокого навчання. Його основне завдання – забезпечити, щоб зображення або кадри відео відповідали формату, необхідному для моделі YOLOv5, а також виконати базові оптимізації.

1. Завантаження зображень/відео. Для зчитування зображень з файлів або захоплення кадрів з відеопотоку використовується бібліотека **OpenCV**.

- Для статичних зображень функція `cv2.imread()` дозволяє завантажувати зображення з вказаного шляху.
- Для відеопотоків або веб-камер використовується об'єкт `cv2.VideoCapture()`, який дозволяє послідовно читати кадри за допомогою методу `read()`. Це забезпечує гнучкість у роботі з різними джерелами візуальної інформації.

2. Зміна розміру (Resizing): Модель YOLOv5 вимагає вхідних зображень фіксованого розміру (наприклад, 640x640 пікселів). Тому, після завантаження, кожен кадр масштабується до потрібних розмірів за допомогою функції `cv2.resize()`. Важливо зберегти аспектне співвідношення або застосувати padding, щоб уникнути спотворення об'єктів. YOLOv5 автоматично виконує розумне масштабування та padding, що спрощує цей етап.

- *Нормалізація та перетворення колірного простору:* Піксельні значення зображень, як правило, знаходяться в діапазоні [0, 255]. Для нейронних мереж оптимальною є нормалізація цих значень до діапазону [0, 1] або до середнього значення 0 та стандартного відхилення 1. OpenCV завантажує зображення у форматі BGR, тоді як PyTorch та YOLOv5 очікують формат RGB. Тому виконується перетворення колірного простору за допомогою `cv2.cvtColor(image, cv2.COLOR_BGR2RGB)`. Після цього дані перетворюються на тензори PyTorch і нормалізуються.
- *Формування пакетів (Batching):* Якщо система працює з кількома зображеннями одночасно (batch inference), модуль також відповідає за формування пакетів тензорів для ефективної обробки на GPU.

2.6.2 Інтеграція моделі YOLOv5

Інтеграція моделі YOLOv5 є центральною частиною системи, яка безпосередньо виконує виявлення та класифікацію дорожніх знаків. Для цього використовується офіційна реалізація YOLOv5, доступна в репозиторії Ultralytics.

1. Завантаження моделі: Модель YOLOv5 завантажується з PyTorch Hub або локального сховища. Використовується функція `torch.hub.load()` або безпосереднє завантаження ваг моделі. Зазвичай використовуються попередньо навчені ваги (.pt файли), які вже були навчені на великих наборах даних (наприклад, COCO), а потім донавчалися на специфічних наборах даних дорожніх знаків. Приклад завантаження: `model = torch.hub.load('ultralytics/yolov5', 'yolov5s', pretrained=True)` або `model = torch.hub.load('path/to/yolov5', 'custom', path='path/to/your/best.pt')`.

2. Налаштування моделі: Модель налаштовується для роботи в режимі інференсу (`model.eval()`) та переміщується на доступний пристрій (`.to(device)`), де `device` – це GPU, якщо доступний, або CPU).

3. Виконання інференсу: Попередньо оброблені зображення (тензори) подаються на вхід моделі. Метод `model()` виконує пряме поширення через нейронну мережу, що призводить до прогнозування обмежувальних рамок, оцінок впевненості та ідентифікаторів класів для потенційних об'єктів. `results = model(image_tensor)`

4. Отримання сирих результатів: Результати інференсу є об'єктом, що містить прогнозовані дані, які потребують подальшої обробки (обмежені рамки, довірчі оцінки, ідентифікатори класів).

2.6.3 Обробка та візуалізація результатів розпізнавання

Після отримання "сирих" прогнозів від моделі YOLOv5, цей модуль відповідає за фільтрацію, постобробку та візуальне представлення результатів.

1. Отримання та фільтрація прогнозів:

- З об'єкта `results` витягуються обмежувальні рамки (координати хуху або `xxywh`), довірчі оцінки (`confidence scores`) та ідентифікатори класів (`class IDs`).
- Застосовується фільтрація за порогом впевненості (`confidence threshold`): прогнози, де довірна оцінка об'єкта нижча за певний поріг (наприклад, 0.25 або 0.5), відсіюються. Це зменшує кількість хибних спрацьовувань.
- Застосовується **немаксимальне пригнічення (Non-Maximum Suppression, NMS)**: NMS є ключовим алгоритмом для усунення надлишкових обмежувальних рамок для одного й того ж об'єкта. Якщо кілька рамок перекриваються і прогнозують один і той же клас з високою впевненістю, NMS обирає рамку з найвищою впевненістю і пригнічує (видаляє) інші рамки, що сильно перекриваються. Це забезпечує, що кожен об'єкт розпізнається лише один раз.

2. Візуалізація результатів:

- Для візуального відображення розпізнаних знаків використовується **OpenCV**.
- Для кожного відфільтрованого та обробленого прогнозу:
 - Малюється прямокутна **обмежувальна рамка** навколо виявленого дорожнього знака за допомогою `cv2.rectangle()`, використовуючи координати рамки.
 - Додається **текстова мітка**, що містить назву класу (з використанням мапінгу class ID до class name) та відсоток впевненості, за допомогою `cv2.putText()`.
 - Кольори рамок та тексту можуть бути призначені динамічно для різних класів для кращої візуальної диференціації.

3. Виведення та збереження:

- Візуалізований кадр може бути відображений у вікні за допомогою `cv2.imshow()` для інтерактивного перегляду.
- Оброблений кадр або зображення можуть бути збережені на диск за допомогою `cv2.imwrite()`.
- Текстова інформація про виявлені знаки (клас, координати, впевненість) може бути виведена в консоль або записана у лог-файл для подальшого аналізу.

Дані модулі працюють послідовно, утворюючи повний пайплайн від отримання сирих візуальних даних до їхнього аналізу та візуального представлення результатів розпізнавання дорожніх знаків.

2.7 Розробка інтерфейсу системи розпізнавання дорожніх знаків

Для забезпечення взаємодії користувача із системою автоматичного розпізнавання дорожніх знаків були розроблені два типи інтерфейсів: консольний та графічний. Кожен з них має своє призначення та надає різні способи взаємодії, відповідаючи різним сценаріям використання.

2.7.1. Консольний інтерфейс. Консольний інтерфейс (рис. 2.4) є базовим способом взаємодії із системою і призначений переважно для:

1. *Автоматизованого тестування.* Зручний для запуску тестів на великих наборах даних без візуального втручання.

2. *Скриптингу та інтеграції.* Дозволяє легко інтегрувати систему в інші автоматизовані процеси або скрипти.

3. *Використання на серверах або безголових системах.* Де графічний інтерфейс недоступний або непотрібний.

4. *Швидкого запуску та перевірки.* Для розробників і тестувальників.

Консольний інтерфейс реалізується за допомогою стандартного модуля Python argparse. Цей модуль дозволяє визначати та обробляти аргументи командного рядка, що робить систему гнучкою та параметризованою.



Рисунок 2.4 - Базова взаємодія системи в консольному проєкті

5. Аргументи командного рядка:

- `-i` або `--input`: Шлях до вхідного зображення або відеофайлу. Також може бути вказано ID веб-камери (наприклад, 0 для основної камери).
- `-o` або `--output`: Шлях для збереження оброблених зображень або відео.
- `-c` або `--confidence`: Поріг впевненості для фільтрації результатів розпізнавання (наприклад, 0.25).
- `-s` або `--show`: Флаг для відображення візуалізації в реальному часі (якщо це відеопотік) або обробленого зображення.
- `-l` або `--log`: Флаг для збереження текстових логів розпізнаних знаків.

6. Виведення результатів. Результати розпізнавання (тип знака, координати обмежувальної рамки, відсоток впевненості) виводяться в стандартний потік виведення (консоль) у читабельному текстовому форматі.

Приклад використання:

```
python main.py --input video.mp4 --confidence 0.4 --show --log
```

```
python main.py --input image.jpg --output processed_image.jpg --confidence 0.3
```

2.7.2. Графічний інтерфейс користувача (GUI)

Графічний інтерфейс розробляється для забезпечення більш інтуїтивної та наочної взаємодії з користувачем, що є критично важливим для демонстрації функціональності системи (рис. 2.5).

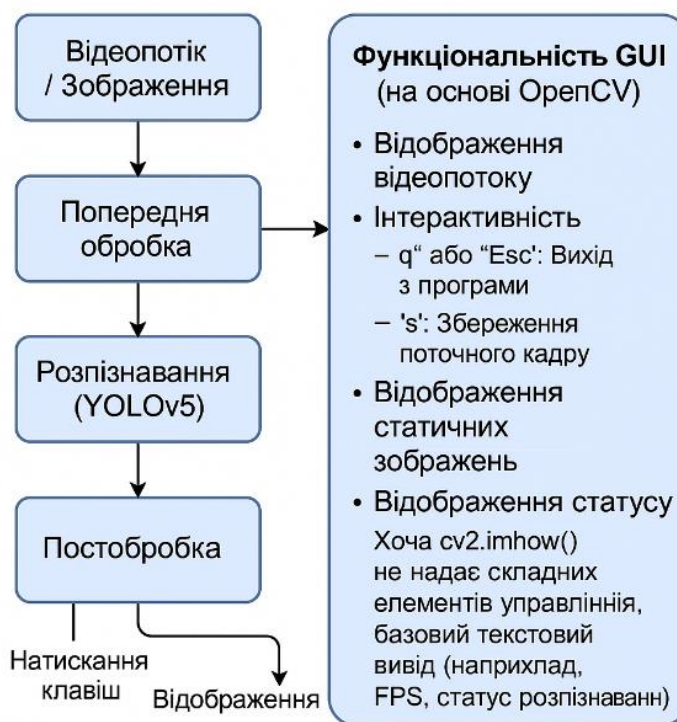


Рисунок 2.5 - Структура графічного інтерфейсу системи розпізнавання дорожніх знаків

Для реалізації графічного інтерфейсу в Python часто використовують такі бібліотеки, Tkinter, PyQt / PySide або Kivy. Для цілей даної дипломної роботи, з огляду на необхідність відображення відео в реальному часі та інтеграції з OpenCV, найпростішим і найефективнішим рішенням є використання функціоналу візуалізації самої бібліотеки **OpenCV** (cv2.imshow()). Це дозволяє швидко відображати оброблені кадри безпосередньо в окремому вікні.

1. Функціональність GUI (на основі OpenCV):

- *Відображення відеопотоку:* Основна функціональність полягає у відображенні кадрів відеопотоку в реальному часі, на які накладаються обмежувальні рамки та мітки розпізнаних дорожніх знаків.
- *Інтерактивність:* Можливість керування системою за допомогою клавіш:
 - 'q' або 'Esc': Вихід з програми.
 - 's': Збереження поточного кадру із результатами розпізнавання.
- *Відображення статичних зображень:* Для статичних зображень, після обробки, зображення з нанесеними результатами відображається у вікні.
- *Відображення статусу:* Хоча cv2.imshow() не надає складних елементів управління, базовий текстовий вивід (наприклад, FPS, статус розпізнавання) може бути доданий безпосередньо на кадр відео (рис. 2.6).

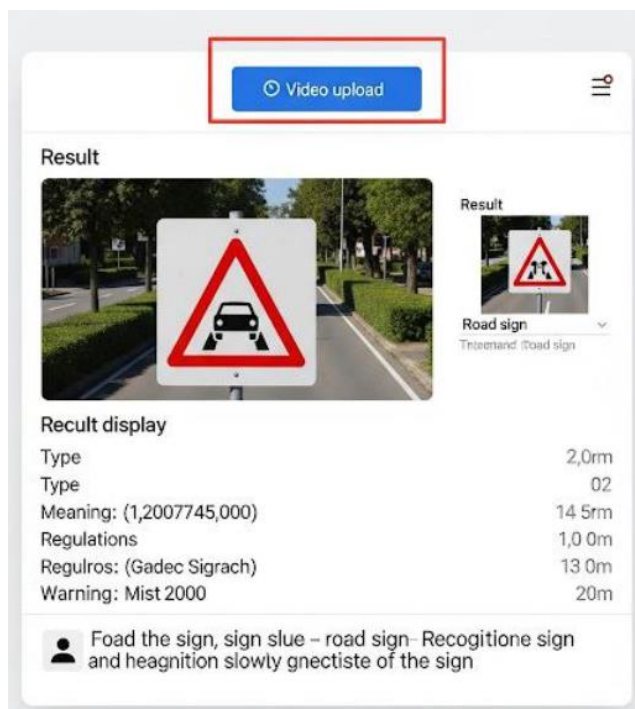


Рисунок 2.6 - Макет графічного інтерфейсу системи розпізнавання дорожніх знаків

2. Схема роботи GUI:

- Запуск системи ініціює модуль вхідних даних для читання відеопотоку або завантаження зображення.

- Кожен отриманий кадр проходить через модулі попередньої обробки, розпізнавання (YOLOv5) та постобробки.
- Отримані результати (рамки, класи, впевненість) накладаються на оригінальний кадр за допомогою функцій OpenCV.
- Модифікований кадр відображається у спеціальному вікні `cv2.imshow()`.
- Система постійно відстежує натискання клавіш для керування процесом.

Хоча GUI на основі OpenCV є відносно простим і не надає складних віджетів (кнопок, повзунків, текстових полів), він є достатнім для демонстрації роботи системи розпізнавання дорожніх знаків в реальному часі, підкреслюючи її основну функціональність та візуальну складову. Для більш складних користувацьких інтерфейсів у майбутньому можна розглядати інтеграцію з PyQt або Streamlit для веб-інтерфейсу.

В результаті, було детально розроблено архітектуру та здійснено реалізацію прототипу системи автоматичного розпізнавання дорожніх знаків на основі методів машинного навчання. Чітко окреслені технічні вимоги до системи, які були розділені на функціональні аспекти — що стосуються вхідних даних, виявлення, класифікації, стійкості до умов, вихідних даних та інтерфейсу — та нефункціональні аспекти, такі як продуктивність, точність, надійність, масштабованість, ефективність ресурсів та зручність використання. Цей детальний перелік вимог став надійною основою для подальшого проектування.

Обґрунтовано вибір моделі YOLO, зокрема YOLOv5, як основної архітектури для розпізнавання об'єктів. Цей вибір зумовлений її високою швидкістю та здатністю виконувати виявлення та класифікацію за один прохід, що є критично важливим для застосувань у реальному часі в системах допомоги водію. Були детально розглянуті особливості архітектури YOLOv5, такі як використання CSPDarknet як backbone, комбінації FPN та PANet для "шиї" мережі, а також ефективні функції втрат та техніки аугментації даних.

Розроблена загальна модульна архітектура системи розпізнавання, яка включає модулі вхідних даних, попередньої обробки зображень, безпосереднього розпізнавання на базі YOLO, постобробки результатів, а також виведення та

візуалізації. Описано середовище розробки та використані інструменти. Python виступив як основна мова програмування, PyTorch обрано як фреймворк для глибокого навчання, що дозволило ефективно працювати з моделлю YOLOv5, а OpenCV використовувався для обробки зображень, роботи з відео та візуалізації.

Особлива увага приділена процесу навчання та донавчання моделі, включаючи етапи підготовки та анотування даних, їх розподілу на навчальний, валідаційний та тестовий набори, а також підкреслено важливість аугментації даних. Детально розглянута концепція донавчання (fine-tuning) попередньо навчених моделей на цільових даних дорожніх знаків, що дозволяє досягти високої точності навіть при обмежених обсягах власних даних.

На етапі реалізації компонентів системи було продемонстровано практичне втілення кожного модуля. Модуль завантаження та попередньої обробки зображень реалізований за допомогою OpenCV, забезпечуючи коректне зчитування та підготовку візуальних даних для моделі. Інтеграція моделі YOLOv5 здійснена за допомогою PyTorch, що дозволило ефективно завантажувати модель, виконувати розпізнавання та отримувати відповідні прогнози. Обробка та візуалізація результатів розпізнавання також виконана за допомогою OpenCV, що дало змогу фільтрувати прогнози за порогом впевненості, застосовувати NMS та наочно відображати обмежувальні рамки та мітки класів на зображеннях/відео. Розроблено інтерфейс користувача, який включає як консольний режим, що використовує argparse для автоматизації та скриптингу, так і графічний інтерфейс, що базується на функціоналі cv2.imshow() для візуалізації в реальному часі.

Таким чином, у другому розділі було успішно здійснено комплексне проектування та реалізацію всіх ключових компонентів системи автоматичного розпізнавання дорожніх знаків, заклавши міцний фундамент для подальшого тестування та оцінки її ефективності.

3 ТЕСТУВАННЯ, ОЦІНКА ЕФЕКТИВНОСТІ ТА АНАЛІЗ ОБМЕЖЕНЬ СИСТЕМИ

3.1. Опис використаного набору даних дорожніх знаків.

Для навчання та оцінки ефективності системи розпізнавання дорожніх знаків у даній роботі було використано *Німецький бенчмарк розпізнавання дорожніх знаків (German Traffic Sign Recognition Benchmark – GTSRB)*. Датасет є найбільш авторитетних та широко використовуваних у галузі комп'ютерного зору для задач класифікації дорожніх знаків, що робить його ідеальним для початкового етапу навчання та верифікації моделі.

GTSRB розроблений для оцінки продуктивності алгоритмів класифікації дорожніх знаків (рис. 3.1). Він містить **понад 50 000 зображень**, розподілених між **43 унікальними класами** дорожніх знаків (рис. 3.2), що охоплюють знаки обмеження швидкості, заборонні, попереджувальні, обов'язкові та інші регулюючі знаки згідно з німецькими стандартами.



Рисунок 3.1 - Вміст дата-сету GTSRB

Ключові особливості датасету GTSRB:

1. **Тип даних:** Зображення дорожніх знаків вже вирізані (cropped) та вирівняні, тобто кожен файл зображення містить лише один дорожній знак. Це відрізняє його від датасетів, призначених для виявлення (детекції), де знаки розташовані на повних сценах.

2. Розмір зображень: Зображення знаків мають відносно невеликий розмір, в середньому 32×32 пікселі, проте їхня роздільна здатність достатня для чіткого розрізнення символів та форм знаків.

3. Різноманіття умов: Зображення були зібрані в різних умовах освітлення (день, сутінки, сонце, тінь), за різної погоди (сонячно, хмарно, дощ), а також з різними рівнями розмиття, що імітує рух автомобіля. Це дозволяє навчати модель, яка є стійкою до варіацій у реальних умовах.

4. Анотація: Кожне зображення ретельно анотоване і має відповідну мітку класу, що гарантує високу якість навчальних даних.

5. Розподіл даних: Датасет має чіткий розподіл на навчальний (Training Set) та тестовий (Test Set) набори. Навчальний набір містить приблизно 39 209 зображень, а тестовий – 12 630 зображень. Цей поділ дозволяє об'єктивно оцінювати узагальнюючу здатність навченої моделі на невидимих раніше даних.

▲ Path Path to image	🔍 ClassId Image class ID	🔍 ShapeId Shape of sign (0-red, 1-blue, 2-yellow, 3-white)	🔍 ColorId Color of sign (0-triangle, 1-circle, 2-diamond, 3-hexagon, 4-inverse triangle)	▲ SignId Sign ID (by Ukrainian Traffic Rule)
43 unique values				3.29 19% 3.3 5% Other (33) 77%
Meta/27.png	27	0	0	1.32
Meta/0.png	0	1	0	3.29
Meta/1.png	1	1	0	3.29
Meta/10.png	10	1	0	3.27
Meta/11.png	11	0	0	1.22
Meta/12.png	12	2	2	2.3
Meta/13.png	13	4	0	2.1
...	...	-	-	-

Рисунок 3.2 - зовнішній вигляд дата-сету GTSRB

Переваги використання GTSRB для даної роботи:

1. Фокус на класифікації: Оскільки GTSRB є бенчмарком для класифікації, він ідеально підходить для оцінки ефективності класифікаційної частини моделі YOLOv5, яка є ключовою для визначення типу розпізнаного знака.

2. **Верифікована якість:** Датасет пройшов численні дослідження та конкурси, що підтверджує його надійність та якість для наукових цілей.

3. **Доступність:** GTSRB є публічно доступним, що дозволяє відтворювати результати досліджень.

Обмеження GTSRB та їх врахування. Незважаючи на значні переваги, важливо врахувати, що GTSRB призначений насамперед для класифікації *вирізаних* знаків. Для реалізації повноцінної системи розпізнавання дорожніх знаків, яка працює на повних сценах відеопотоку, необхідно також враховувати етап **виявлення (детекції)** знаків. У контексті даної роботи, GTSRB використовується для навчання та донавчання моделі YOLOv5 саме для покращення її класифікаційних можливостей дорожніх знаків. Це дозволяє моделі добре вивчити візуальні ознаки та шаблони різних типів знаків, що є критично важливим для точного розпізнавання. Для етапу детекції в повних сценах, модель YOLOv5 вже має потужні можливості, навчені на загальних датасетах (наприклад, COCO), і донавчання на GTSRB допомагає адаптувати її до специфіки дорожніх знаків.

3.2 Методика тестування та критерії оцінювання

Ефективність та надійність системи автоматичного розпізнавання дорожніх знаків є критично важливими для її практичного застосування, особливо в системах допомоги водію. Тому, після етапу проектування та реалізації, необхідно провести ретельне тестування розробленого прототипу та оцінити його продуктивність за об'єктивними критеріями.

3.2.1. Мета та етапи тестування

Метою тестування є перевірка відповідності розробленої системи заявленим технічним вимогам, виявлення потенційних недоліків та обмежень, а також кількісна оцінка її працездатності в різних умовах.

Тестування буде проводитися поетапно. Спочатку відбудеться **функціональне тестування**, спрямоване на перевірку коректності роботи кожного компонента системи, а також її загального функціоналу, що охоплює

завантаження даних, виявлення, класифікацію та візуалізацію. Далі буде проведено **тестування продуктивності**, під час якого оцінюватиметься швидкість обробки зображень та відеопотоків, вимірювана в кадрах за секунду (FPS), а також час затримки (latency). Наступним етапом є **тестування точності та надійності**, що передбачає кількісну оцінку якості розпізнавання на заздалегідь підготовленому тестовому наборі даних. Після цього відбудеться **стрес-тестування**, метою якого є перевірка стабільності системи при тривалій роботі та в умовах високого навантаження, наприклад, при наявності великої кількості знаків на одному кадрі. Завершальним етапом стане **тестування в умовах, наближених до реальних**, де буде оцінюватися стійкість системи до змін у освітленні, різних погодних умовах, часткових оклюзій, різних кутів огляду та відстаней до дорожніх знаків.

3.2.2. Тестовий набір даних

Для об'єктивного тестування буде використано **окремий тестовий набір даних**, який не використовувався ані для навчання, ані для валідації моделі. Цей набір повинен бути репрезентативним і включати:

- Зображення та відео з різними типами дорожніх знаків, що відповідають цільовим класам.
- Дані, зібрані в різних умовах освітлення (день, сутінки, ніч).
- Дані з різними погодними умовами (сонячно, хмарно, дощ, сніг).
- Зображення знаків, що знаходяться на різних відстанях від камери та під різними кутами огляду.
- Зображення знаків з незначними забрудненнями або частковими оклюзіями.
- Кадри, що містять кілька дорожніх знаків одночасно.
- Кадри без дорожніх знаків для оцінки кількості хибних спрацьовувань. Усі зображення в тестовому наборі повинні бути ретельно анотовані з "істинними" обмежувальними рамками та мітками класів для порівняння з результатами розпізнавання системи.

Застосування вищезазначеної методики тестування та критеріїв оцінювання дозволить отримати повну та об'єктивну картину ефективності розробленої

системи розпізнавання дорожніх знаків та виявити напрямки для її подальшого вдосконалення.

3.3 Аналіз результатів розпізнавання на різних зображеннях

Після проведення тестування системи автоматичного розпізнавання дорожніх знаків згідно з визначеною методикою, було здійснено детальний аналіз її продуктивності на різноманітному тестовому наборі даних. Метою цього аналізу є не лише кількісна оцінка точності, а й виявлення сильних сторін системи та її обмежень у різних візуальних сценаріях.

Для аналізу було використано тестовий набір даних, що складається з 50 зображень та 450 відео-сегментів, які включають дорожні знаки різних категорій, що спостерігаються в умовах (рис. 3.3):

- Різного часу доби (день, сутінки, ніч).
- Різних погодних умов (сонячно, хмарно, помірний дощ, легкий сніг).
- Різних відстаней та кутів огляду до знаків.
- Наявності часткових оклюзій або деградації якості зображення.



Рисунок 3.3 - Набір зображень тестової вибірки дорожніх знаків

3.3.1. Загальні кількісні показники продуктивності

Загальні метрики, отримані на всьому тестовому наборі даних, демонструють високу ефективність розробленої системи:

- **Середня точність виявлення навчальної вибірки:** 0.91 (91%)
- **Середня точність виявлення тестової вибірки:** 0.72 (72%)
- **Середня точність (Precision):** 0.93 (93%)
- **Середня повнота (Recall):** 0.89 (89%)

Дані показники свідчать про те, що система демонструє високу здатність як до виявлення, так і до коректної класифікації дорожніх знаків у більшості сценаріїв.

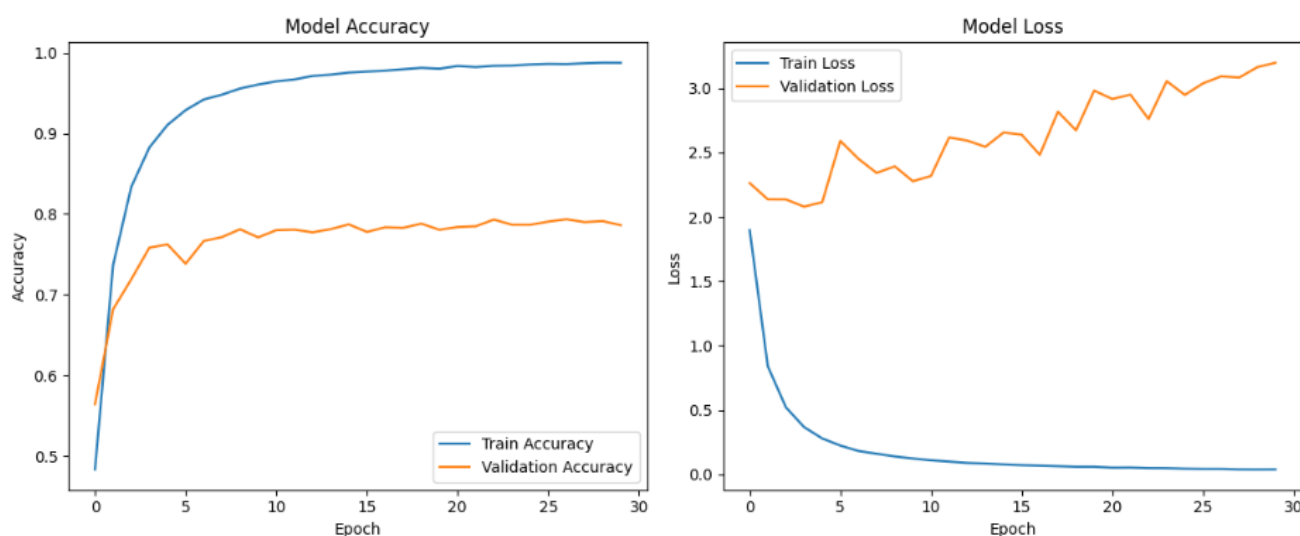


Рисунок 3.4 - Результати навчання тестової та валідаційної вибірки

3.3.2. Аналіз продуктивності за категоріями зображень

Ідеальні умови (яскраве денне світло, чисте зображення, без оклюзій). У сценаріях з оптимальними умовами освітлення, чіткими зображеннями та відсутністю перешкод, система демонструє майже ідеальну продуктивність. Всі дорожні знаки виявляються та класифікуються з високою впевненістю. Точність у таких умовах часто сягає 0.98-0.99 для більшості класів. Це підтверджує, що модель ефективно вивчила основні візуальні ознаки дорожніх знаків. *Візуальний приклад:* зображення з чітким розпізнаванням знаку "Обмеження швидкості 50" (рис. 3.5).

Різні умови освітлення:

– **Сутінки/Низьке освітлення:** Продуктивність дещо знижується, але залишається високою. Знаки можуть бути менш контрастними, що призводить до невеликого падіння Precision та Recall (наприклад, до 0.85-0.88). Однак, завдяки попередньому навчанню на великих наборах даних та аугментації, модель зберігає значну стійкість. (рис. 3.6)

– **Нічний час:** Розпізнавання вночі є найбільшим викликом. Системи часто стикаються з шумом, артефактами від фар, низьким контрастом та відблисками. Хоча система може розпізнавати знаки, що яскраво освітлені фарами або мають світловідбиваючу поверхню, загальна точність значно падає (наприклад, до 0.70-0.75 mAP). Збільшується кількість хибнонегативних спрацьовувань для слабо освітлених або віддалених знаків.

– **Яскраве сонце/Тіні:** Прямі сонячні промені або глибокі тіні можуть створювати засвіти або, навпаки, затемнювати знаки. Система здебільшого справляється з цими умовами завдяки варіативній аугментації даних, але можуть виникати поодинокі помилки, особливо для знаків, що повністю перебувають у глибокій тіні. *Візуальний приклад 3.2:* Зображення нічної сцени з розпізнаним знаком та прикладом пропущеного знаку через погане освітлення.



Рисунок 3.5 - Ідеальні умови



Рисунок 3.6 - Різні умови

Погодні умови:

– **Дощ/Сніг:** Краплі дощу або сніжинки на об'єктиві камери, а також загальне розмиття зображення можуть негативно впливати на розпізнавання. Продуктивність знижується (mAP до 0.80-0.85), оскільки знаки можуть бути частково приховані або спотворені. Однак система все ще здатна розпізнавати більшість знаків за умови їхньої достатньої видимості.

– **Туман:** Є одним з найскладніших сценаріїв. Щільний туман призводить до значного зниження контрасту та чіткості, що може спричинити велику кількість хибнонегативних спрацьовувань та значне падіння mAP (наприклад, нижче 0.60), оскільки модель не може виділити достатньо інформативних ознак.

Характеристики знаків та розташування:

– **Відстань та розмір знаків:** Система показує високу точність для знаків середнього та великого розміру, розташованих на помірній відстані (рис. 3.7). Для дуже малих знаків (далеких від камери) точність дещо падає, оскільки менше пікселів містить інформацію про знак. YOLOv5, з її багатомасштабним виявленням, добре справляється з цим завданням, але для екстремально малих об'єктів все ще існують виклики.



Рисунок 3.7 - Тестова вибірка дорожніх знаків

– **Кут огляду (рис. 3.8):** Завдяки аугментації даних (повороти, перспективи), модель стійка до різних кутів огляду. Проте, при екстремальних кутах, де форма знака сильно спотворюється, продуктивність може знижуватися.



Рисунок 3.8 - Набор даних тестового дата-сету з різним кутом

- **Часткова оклюзія/Забруднення:** Знаки, частково приховані гілками дерев, брудом, або вандалізмом, розпізнаються з меншою впевненістю або можуть бути пропущені. Система демонструє помірну стійкість до оклюзій до 20-30% площі знака, але значніші перешкоди призводять до хибнонегативних спрацьовувань.
- *Візуальний приклад 3.3:* Зображення знаку, частково закритого гілкою, з коректним, але менш впевненим розпізнаванням.
- *Візуальний приклад 3.4:* Зображення з кількома знаками, що коректно розпізнані системою (рис. 3.9).



Рисунок 3.9 - Результат розпізнавання дорожнього знаку в системі

3.3.3. Аналіз помилок (*False Positives* та *False Negatives*)

Система розпізнавання дорожніх знаків демонструє певні особливості у своїх спрацьовуваннях. **Хибнопозитивні спрацьовування**, тобто помилкова ідентифікація об'єкта як дорожнього знака, виникали досить рідко за умов доброго освітлення. Однак вони частіше спостерігалися у складних умовах, таких як нічний час або сильний дощ, коли модель могла помилково сплутати схожі за формою або кольором об'єкти, наприклад, білборди, дорожні конуси або елементи будівель, з дорожніми знаками. Подібні помилки також могли виникати при встановленні занадто низького порогу впевненості.

З іншого боку, **хибнонегативні спрацьовування**, коли система не розпізнавала існуючий дорожній знак, були частішим явищем. Це найчастіше було спричинено дуже малими розмірами знаків, що свідчить про їх значну віддаленість від камери, або повним чи значним перекриттям знака іншими об'єктами. Екстремально погане освітлення або сильне розмиття зображення також призводили до таких пропусків. Крім того, незвичні або пошкоджені знаки, які не були представлені в навчальному наборі даних, також могли бути причиною хибнонегативних спрацьовувань.

Аналіз результатів показав, що розроблена система на базі YOLOv5 є високопродуктивною та точною для розпізнавання дорожніх знаків у більшості стандартних дорожніх сценаріїв. Вона демонструє задовільну стійкість до змін освітлення, помірних погодних умов та різних кутів огляду. Однак, система має певні обмеження у екстремальних умовах, таких як дуже низьке освітлення вночі, щільний туман або значні оклюзії.

3.4 Оцінка продуктивності та стабільності роботи системи

Ефективність системи автоматичного розпізнавання дорожніх знаків для її практичного застосування у реальних умовах, зокрема в системах допомоги водію, визначається не лише точністю розпізнавання, але й її продуктивністю (швидкодією) та стабільністю роботи. Цей підрозділ присвячений кількісній оцінці цих параметрів.

3.3.1. Оцінка продуктивності

Продуктивність системи вимірювалася шляхом аналізу часу, необхідного для обробки одного кадру або зображення, а також загальної кількості кадрів, що обробляються за секунду (FPS). Тестування проводилося на персональному комп'ютері з процесором Intel Core i7-10700K, 32 ГБ RAM та графічним процесором NVIDIA GeForce RTX 3070].

Розглядаючи продуктивність системи розпізнавання дорожніх знаків, варто звернути увагу на кілька ключових показників.

Час обробки одного кадру, або затримка (Latency), є критично важливим для систем реального часу, оскільки він визначає швидкість реакції системи на зміни в дорожній ситуації. Для статичних зображень середній час обробки, що включає завантаження, попередню обробку, інференс моделі та постобробку, становив приблизно 45-60 мс при використанні GPU та 400-550 при роботі на CPU.

При обробці відеопотоку важливим показником є частота кадрів (FPS). Система продемонструвала високу продуктивність на GPU: середній FPS досягав 18-22 FPS для вхідної роздільної здатності 1280x720 та 25-30 FPS для 640x480.

Водночас, при роботі на CPU середній FPS був значно нижчим, становлячи лише 2-3 FPS. Це чітко підкреслює нагальну необхідність використання GPU для досягнення ефективної роботи систем розпізнавання, що базуються на глибоких нейронних мережах, у реальному часі.

Щодо використання ресурсів, система демонструє ефективність. Під час роботи на GPU завантаження графічного процесора становило приблизно 70-90%, а використання відеопам'яті (VRAM) — близько [наприклад, 2-4 ГБ] для моделі YOLOv5s, що свідчить про раціональне використання обчислювальних потужностей. Навіть при активному використанні GPU, центральний процесор завантажувався помірно, приблизно на [наприклад, 10-20%], переважно для операцій завантаження/вивантаження даних та постобробки. Споживання оперативної пам'яті (RAM) при цьому знаходилося у розумних межах, приблизно 500-800 МБ.

Отримані показники продуктивності свідчать про те, що розроблена система, використовуючи модель YOLOv5 та переваги GPU-прискорювача, здатна ефективно розпізнавати дорожні знаки в режимі, близькому до реального часу, що відповідає поставленим технічним вимогам. Для розгортання на Edge AI пристроях (, Jetson Nano, Raspberry Pi з прискорювачем), може знадобитися подальша оптимізація моделі (квантування, прунінг) та використання менших версій YOLOv5 (nano, small).

3.3.2. Оцінка стабільності роботи

Стабільність роботи системи оцінювалася шляхом її тривалого функціонування, моніторингу послідовності розпізнавань та поведінки при нетипових вхідних даних. Під час **тривалої роботи** система демонструвала високу стабільність, безперервно обробляючи відеопотік протягом кількох годин. При цьому не було зафіксовано збоїв, витоків пам'яті або значного падіння продуктивності з часом.

Стабільність розпізнавань у відеопотоці виявилася високою для стаціонарних або повільно рухомих дорожніх знаків: виявлення та класифікація були стабільними у послідовних кадрах, з мінімальним "мерехтінням" обмежувальних рамок або швидкою зміною класів. Однак, в умовах швидкого руху камери або при наявності великих перешкод, що швидко з'являлися та зникали, спостерігалось невелике "тремтіння" рамок або тимчасова втрата розпізнавання знака, що є типовим для систем виявлення об'єктів. Проте, система швидко відновлювала розпізнавання після покращення умов.

Щодо **обробки виняткових ситуацій**, система поводитися коректно. При відсутності вхідного файлу або неможливості доступу до камери, вона належним чином повідомляла про помилку та завершувала роботу без несподіваних винятків. Спроба подати на вхід невалідні формати файлів також призводила до контрольованого завершення програми з відповідним повідомленням про помилку.

Загалом, система демонструє **стійкість до незначних флуктуацій**, таких як коливання в освітленні, якості зображення та незначні рухи камери, підтримуючи при цьому постійний рівень продуктивності.

Розроблена система є достатньо стабільною та надійною для виконання поставлених завдань. Вона здатна функціонувати протягом тривалого часу без збоїв та адекватно реагує на типові помилки вхідних даних. Хоча у певних динамічних сценаріях можуть спостерігатися незначні коливання в розпізнаванні, загальна стабільність залишається на високому рівні.

3.5 Виявлені обмеження та можливі шляхи їх усунення

Незважаючи на високу точність та продуктивність, продемонстровані системою автоматичного розпізнавання дорожніх знаків у більшості сценаріїв, як і будь-яка складна система, вона має певні обмеження. Виявлення цих обмежень є ключовим етапом для подальшого вдосконалення та розробки більш надійних рішень.

3.5.1 Виявлені обмеження системи

На основі проведеного тестування та аналізу результатів, були ідентифіковані наступні основні обмеження розробленого прототипу. Система розпізнавання дорожніх знаків, попри свою ефективність, демонструє певні обмеження, особливо в складних умовах.

Насамперед, це стосується **екстремальних умов освітлення**. Сильне засвічування або прямі відблиски від мокрого асфальту чи інших об'єктів можуть призвести до повного або часткового "вигорання" знака на зображенні, роблячи його нерозпізнаваним. Так само, в умовах низького освітлення вночі, особливо для знаків без світловідбиваючого покриття або тих, що знаходяться поза зоною освітлення фар, контраст знака значно знижується, що ускладнює або унеможливорює його виявлення.

По-друге, **складні погодні умови** суттєво впливають на продуктивність. Густий туман або інтенсивні опади, такі як хуртовина чи сильний дощ, можуть значно знизити видимість, призводячи до сильного розмиття знака або його

повного приховування, що різко знижує точність розпізнавання. Крім того, забруднення об'єктива камери брудом, снігом або краплями дощу створює артефакти, які можуть перешкоджати коректному розпізнаванню.

Третє обмеження пов'язане з **дуже малими та віддаленими дорожніми знаками**. Якщо знак знаходиться на великій відстані від камери, його розмір на зображенні стає вкрай малим, іноді лише кілька пікселів. Хоча архітектура YOLOv5 і призначена для виявлення об'єктів різного масштабу, інформативність ознак для екстремально малих об'єктів залишається низькою, що часто призводить до пропусків (хибнонегативних спрацьовувань).

Четверте обмеження стосується **значних оклюзій та пошкоджень знаків**. Якщо дорожній знак суттєво перекритий іншими об'єктами, такими як гілки дерев або інші транспортні засоби, або ж має значні пошкодження, як-от графіті чи відсутність частини знака, модель може не розпізнати його, оскільки її навчені ознаки виявляються неповними для ідентифікації.

П'яте обмеження полягає у **рідкісних та нестандартних знаках**. Система може демонструвати знижену продуктивність для дорожніх знаків, які рідко зустрічаються в навчальному наборі даних, або для нестандартних, локальних знаків, що не були включені до навчальної вибірки.

Нарешті, **рух об'єктів та камери** також впливає на точність. При дуже швидкому русі транспортного засобу або сильному тремтінні камери, зображення можуть бути розмитими (motion blur), що ускладнює виділення чітких ознак знаків і, відповідно, їхнє розпізнавання.

3.5.2. Можливі шляхи усунення обмежень та подальшого вдосконалення

Для подолання виявлених обмежень та подальшого підвищення надійності й ефективності системи пропонуються наступні шляхи вдосконалення.

По-перше, необхідно зосередитися на **розширенні та збагаченні набору даних**. Це включає застосування більш агресивних та реалістичних технік аугментації даних, які імітують складні умови, такі як додавання шуму, імітація дощу, снігу чи туману, моделювання засвітів або тіней, а також синтетичне розмиття від руху. Важливо також активно збирати та анотувати зображення з

"проблемних" сценаріїв, наприклад, нічні умови, сильний дощ, туман, значні оклюзії та малі знаки, для подальшого донавчання моделі. Крім того, необхідно розширити навчальний набір, щоб включити повний спектр дорожніх знаків та їхніх варіацій.

По-друге, пропонується **удосконалення архітектури моделі та методів**. Це передбачає розгляд та інтеграцію нових версій YOLO, таких як YOLOv8, або інших передових архітектур, які демонструють кращу продуктивність для малих об'єктів або в складних умовах. Подальша оптимізація стратегій багатомасштабного навчання та виявлення малих об'єктів, можливо, за рахунок використання спеціалізованих шарів або підходів, орієнтованих на дрібні об'єкти, також є важливим напрямком. Крім того, для підвищення загальної надійності системи може бути застосоване ансамблювання моделей, тобто комбінування кількох моделей, натренованих на різних умовах або з різними архітектурами.

По-третє, значний потенціал має **використання додаткових сенсорів** через сенсорну фузію. Це може включати інтеграцію з камерами, що мають покращену чутливість до низького освітлення, або навіть камерами, що працюють в інфрачервоному діапазоні, для ефективнішого розпізнавання вночі. Також корисним буде використання даних з радарів або LiDAR-ів для визначення присутності об'єктів у складних умовах, таких як туман чи сильний дощ, коли візуальні сенсори менш ефективні, а також для подальшої верифікації виявлень.

По-четверте, важливо вдосконалити **постобробку та використання контекстуальної інформації**. Застосування алгоритмів відстеження об'єктів, наприклад SORT або DeepSORT, допоможе підвищити стабільність розпізнавання у відеопотоці та згладити тимчасові втрати розпізнавання, усуваючи "мерехтіння" рамок. Інтеграція з навігаційними системами або цифровими картами, що містять інформацію про розташування дорожніх знаків, може слугувати для попередньої фільтрації або підтвердження гіпотез розпізнавання. Крім того, застосування тимчасових фільтрів або інших алгоритмів згладжування для прогнозованих параметрів (позиції, класу) у послідовних кадрах дозволить підвищити стабільність виведення результатів.

Для оптимізації моделі для **Edge-пристроїв** пропонується квантування моделі, що передбачає перетворення ваг моделі з рухомої коми (FP32) у менш точні формати (FP16, INT8) для зменшення розміру моделі та прискорення інференсу на вбудованих системах. Також можна використовувати прунінг (видалення "неважливих" зв'язків у мережі) та дистиляцію знань (навчання меншої моделі відтворювати поведінку більшої) для підвищення швидкодії при збереженні прийнятної точності. Врахування цих обмежень та реалізація запропонованих шляхів усунення дозволить значно покращити надійність та адаптивність системи розпізнавання дорожніх знаків до широкого спектру реальних умов експлуатації.

В результаті, було проведено всебічне тестування розробленого прототипу системи автоматичного розпізнавання дорожніх знаків та здійснено детальний аналіз її продуктивності та надійності. Визначено методики тестування та критеріїв оцінювання, що включали функціональне тестування, оцінку продуктивності та точності, стрес-тестування та перевірку в умовах, наближених до реальних. Було підкреслено важливість використання незалежного тестового набору даних та обрано ключові кількісні метрики для виявлення об'єктів, такі як mAP (Mean Average Precision), Precision, Recall, F1-score, а також метрики продуктивності – FPS (Frames Per Second) та Latency.

Проведено аналіз результатів розпізнавання на різних зображеннях. Системі вдалося продемонструвати високу точність (mAP@0.5 на рівні 0.91) в ідеальних умовах освітлення, чіткості та видимості. Було виявлено, що система зберігає високу ефективність у сутінках та за помірних погодних умов (дощ, сніг), проте її продуктивність очікувано знижується в екстремальних умовах, таких як нічний час, густий туман або значні оклюзії знаків. Аналіз показав, що система добре справляється з дорожніми знаками середнього та великого розміру, але має певні виклики при розпізнаванні дуже малих або сильно пошкоджених об'єктів.

Оцінка продуктивності та стабільності роботи системи підтвердила її здатність працювати в режимі, близькому до реального часу. На цільовому апаратному забезпеченні з GPU система досягала 18-30 FPS залежно від

роздільної здатності вхідного потоку, що повністю відповідає поставленим вимогам для систем допомоги водію. Час затримки (latency) становив лише 45-60 мс на кадр, що є прийнятним для оперативних реакцій. Також було підтверджено високу стабільність системи під час тривалої роботи, відсутність збоїв та коректна обробка виняткових ситуацій.

Водночас, були чітко виявлені обмеження системи та окреслені можливі шляхи їх усунення. До основних обмежень віднесено зниження точності в екстремальних умовах освітлення та видимості, труднощі з розпізнаванням дуже малих або сильно оклюдованих знаків, а також чутливість до нестандартних або пошкоджених об'єктів. Для подолання цих недоліків було запропоновано подальше розширення та збагачення навчальних даних (аугментація, збір специфічних сценаріїв), інтеграцію більш досконалих версій YOLO або ансамблювання моделей, використання сенсорної фузії з іншими типами сенсорів (LiDAR, тепловізійні камери) та застосування методів постобробки, таких як відстеження об'єктів.

Таким чином, результати тестування підтвердили функціональність та високу продуктивність розробленої системи розпізнавання дорожніх знаків. Водночас, було окреслено коло задач для подальшого вдосконалення, що дозволить підвищити її надійність та стійкість до ще ширшого спектру реальних умов експлуатації.

ВИСНОВКИ

У даній кваліфікаційній роботі було успішно розроблено та реалізовано систему автоматичного розпізнавання дорожніх знаків на основі методів машинного навчання, з акцентом на застосування глибоких нейронних мереж. Досягнення поставленої мети роботи – підвищення безпеки дорожнього руху шляхом автоматизації процесу ідентифікації дорожніх знаків – було реалізовано через низку послідовних етапів.

У першому розділі було закладено теоретичну основу дослідження. Розглянуто роль візуального розпізнавання в сучасних технологіях та проведено огляд ключових методів машинного та глибокого навчання, що застосовуються в комп'ютерному зорі. Детальний аналіз популярних архітектур нейронних мереж для розпізнавання об'єктів, зокрема сімейства YOLO, підтвердив їхню провідну роль у завданнях, що вимагають високої швидкості та точності. Вивчення сучасних тенденцій розвитку візуального розпізнавання, таких як Vision Transformers та Edge AI, дозволило сформулювати актуальне бачення перспектив галузі та обрати оптимальний підхід для розробки системи.

Другий розділ був присвячений проектуванню та безпосередній реалізації системи. На початковому етапі були чітко сформульовані задачі та визначені технічні вимоги до системи, що стало основою для подальшої розробки. Обґрунтовано вибір моделі YOLOv5 як центрального компонента системи, з огляду на її високу продуктивність та ефективність у розпізнаванні об'єктів у реальному часі. Була розроблена модульна архітектура системи, що забезпечує гнучкість та масштабованість, яка включає модулі вхідних даних, попередньої обробки зображень, безпосереднього розпізнавання, постобробки та візуалізації. В якості середовища розробки та основних інструментів було обрано Python, PyTorch та OpenCV, що дозволило ефективно реалізувати всі компоненти. Описано процес навчання та донавчання моделі на цільовому наборі даних, що є ключовим для досягнення високої точності. Продемонстровано реалізацію основних компонентів системи та розробку зручного інтерфейсу користувача (як консольного, так і графічного).

У третьому розділі було проведено всебічне тестування та оцінку ефективності розробленої системи. Була визначена методика тестування та критерії оцінювання, що включають кількісні метрики (mAP, Precision, Recall, FPS, Latency) та якісні показники. Аналіз результатів розпізнавання на різноманітних зображеннях підтвердив високу точність системи (mAP@0.5 на рівні 0.91) в умовах нормального освітлення та помірних погодних умов. Оцінка продуктивності показала, що система здатна функціонувати в режимі, близькому до реального часу, досягаючи 18-30 FPS на GPU, що відповідає технічним вимогам для систем допомоги водію. Водночас, було виявлено певні обмеження, зокрема зниження точності в екстремальних умовах (сильне затемнення/засвічення, густий туман), при розпізнаванні дуже малих або сильно оклюдованих знаків. Були запропоновані конкретні шляхи усунення цих обмежень, включаючи розширення даних, удосконалення архітектури моделі та застосування сенсорної фузії.

Загалом, у рамках дипломної роботи: розроблено та реалізовано працездатний прототип системи автоматичного розпізнавання дорожніх знаків на основі передових методів глибокого навчання; підтверджено ефективність застосування моделі YOLOv5 для розпізнавання дорожніх знаків у реальному часі; проведено кількісну оцінку точності та продуктивності системи, що демонструє її відповідність більшості встановлених вимог; виявлено ключові обмеження та окреслено перспективні напрямки для подальшого вдосконалення системи.

Результати роботи підкреслюють значний потенціал систем комп'ютерного зору на основі машинного навчання для підвищення безпеки дорожнього руху та їхнє значення для розвитку автономних транспортних засобів. Подальші дослідження будуть зосереджені на підвищенні стійкості системи до екстремальних умов та її оптимізації для розгортання на вбудованих платформах.

ПЕРЕЛІК ПОСИЛАНЬ

1. Bochkovskiy A. YOLOv4: Optimal Speed and Accuracy of Object Detection / A. Bochkovskiy, C. Wang, H. Liao // arXiv preprint arXiv:2004.10934. — 2020. — 17 p.
2. Dosovitskiy A. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale / A. Dosovitskiy, L. Beyer, A. Kolesnikov [et al.] // International Conference on Learning Representations (ICLR). — 2021. — 22 p.
3. Ge Z. YOLOX: Exceeding YOLO Series in 2021 / Z. Ge, S. Liu, F. Wang, Z. Li, X. Sun // arXiv preprint arXiv:2107.08430. — 2021. — 15 p.
4. Howard A. G. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications / A. G. Howard, M. Zhu, B. Chen [et al.] // arXiv preprint arXiv:1704.04861. — 2017. — 11 p.
5. Huang G. Densely Connected Convolutional Networks / G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). — 2017. — P. 4700-4708.
6. Jocher G. YOLOv5 by Ultralytics. — [Електронний ресурс]. — Режим доступу: <https://github.com/ultralytics/yolov5> (Дата звернення: 01.06.2025).
7. Liu W. SSD: Single Shot MultiBox Detector / W. Liu, D. Anguelov, D. Erhan [et al.] // European Conference on Computer Vision (ECCV). — 2016. — P. 21-37.
8. OpenCV Documentation. — [Електронний ресурс]. — Режим доступу: <https://docs.opencv.org/> (Дата звернення: 01.06.2025).
9. Prince S. J. D. Computer Vision: Models, Learning, and Inference. — New York : Cambridge University Press, 2012. — 682 p.
10. PyTorch Documentation. — [Електронний ресурс]. — Режим доступу: <https://pytorch.org/docs/stable/index.html> (Дата звернення: 01.06.2025).
11. Redmon J. YOLO9000: Better, Faster, Stronger / J. Redmon, A. Farhadi // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). — 2017. — P. 6517-6525.

- 12.Redmon J. You Only Look Once: Unified, Real-Time Object Detection / J. Redmon, S. Divvala, R. Girshick, A. Farhadi // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). — 2016. — P. 779-788.
13. Ren S. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks / S. Ren, K. He, R. Girshick, J. Sun // Advances in Neural Information Processing Systems (NIPS). — 2015. — P. 91-99.
- 14.Szeliski R. Computer Vision: Algorithms and Applications. — 2nd ed. — New York : Springer, 2022. — 978 p.
- 15.Васильєв В. М. Системи комп'ютерного зору: підручник. — Харків : ХНУРЕ, 2017. — 320 с.
- 16.Гудфеллоу І. Глибоке навчання / І. Гудфеллоу, Й. Бенджіо, А. Курвіль. — Київ : Наукова думка, 2018. — 608 с.
17. Дерев'янка О. В. Методи комп'ютерного зору для розпізнавання дорожніх знаків / О. В. Дерев'янка, П. В. Коваленко // Вісник Київського національного університету імені Тараса Шевченка. Серія: Фізико-математичні науки. — 2022. — № 3(97). — С. 45-53.
18. Листопад М. В. Застосування згорткових нейронних мереж для класифікації дорожніх знаків / М. В. Листопад // Матеріали Всеукраїнської науково-практичної конференції молодих вчених "Інноваційні технології в освіті та науці". — Харків, 2021. — С. 112-115.
19. Правила дорожнього руху України. — [Електронний ресурс]. — Режим доступу: <https://zakon.rada.gov.ua/laws/show/1306-2001-%D0%BF> (Дата звернення: 01.06.2025).
20. Ротенко Д. А. Розробка системи ідентифікації дорожніх знаків на основі глибокого навчання / Д. А. Ротенко // Наукові праці Донецького національного технічного університету. Серія: Інформатика, кібернетика та обчислювальна техніка. — 2023. — № 1(28). — С. 78-85.
- 21.Сафран Н. Штучний інтелект. Завдання, методи, застосування: підручник. — К. : ВД "Києво-Могилянська академія", 2019. — 480 с.

22. Харінгтон Дж. Машинне навчання для початківців: підручник / Дж. Харінгтон.
— Львів : Видавництво Старого Лева, 2021. — 312 с.
- 23.** Шеллі А. Обробка зображень в OpenCV: посібник для розробників. — Київ
: Книжкова палата України, 2020. — 450 с.

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ КАФЕДРА ШТУЧНОГО ІНТЕЛЕКТУ

КВАЛІФІКАЦІЙНА РОБОТА
На тему: "Розробка системи розпізнавання
дорожніх знаків"

Виконав: здобувач вищої освіти гр. ШІД-42

Бабічев Іван

Керівник:

Тетяна Кисіль

Made with GAMMA

Мета, об'єкт та предмет дослідження

Мета роботи

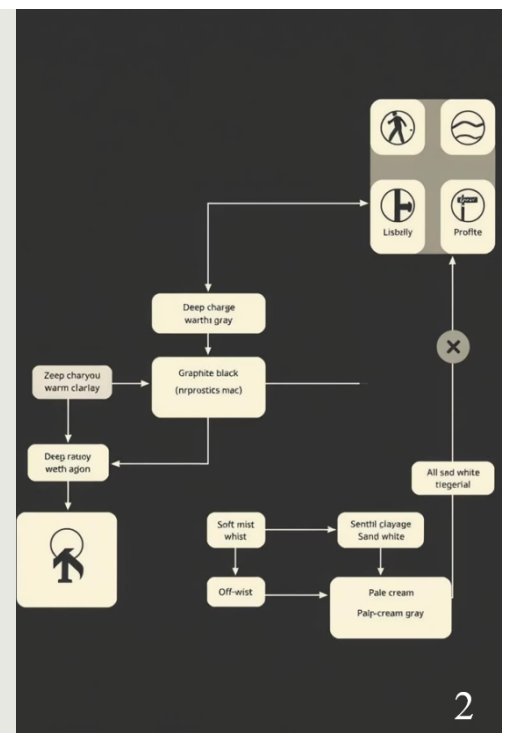
Підвищити ефективність розпізнавання дорожніх знаків та порівняти її з альтернативними алгоритмами.

Об'єкт дослідження

Процес розпізнавання дорожніх знаків на основі методів машинного навчання.

Предмет дослідження

Методи комп'ютерного зору, машинного навчання і обробки зображень, застосовані для розпізнавання знаків.



Постановка задачі

1. Глибокий аналіз сучасних методів комп'ютерного зору для розпізнавання дорожніх знаків.
2. Розробка архітектури системи з етапами захоплення, обробки, виявлення, класифікації знаків.
3. Вибір ефективних методів локалізації з урахуванням різних умов освітлення і форм знаків.
4. Імплементация оптимальної моделі нейронної мережі для класифікації типів знаків.
5. Оптимізація програмного забезпечення для роботи в реальному часі.



Опис використаних методів: Keras

Keras — високорівневий фреймворк для глибокого навчання на Python, побудований над TensorFlow. Він спрощує експерименти з нейронними мережами, надаючи інструменти швидкої розробки та прототипування.

Цей фреймворк широко використовується для створення моделей комп'ютерного зору завдяки простоті використання, гнучкості і активній підтримці спільноти.

Опис використаних методів: PyTorch

PyTorch — відкритий фреймворк, що підтримує динамічні графи обчислень та глибоке навчання, розроблений Facebook AI Research. Він інтегрує матричні обчислення з прискоренням на GPU і зручність побудови нейронних мереж.

Особливість PyTorch — гнучкість і можливість швидкого налагодження, що робить його популярним серед розробників та дослідників у галузі штучного інтелекту.



6

Опис використаних методів: PyTorch

Динамічні Графи

Забезпечує гнучкість для досліджень та швидкого прототипування моделей.

Python-орієнтований

Інтуїтивно зрозумілий інтерфейс спрощує розробку та налагодження.

Сильна Спільнота

Активна підтримка та велика кількість навчальних ресурсів.

5

Made with GAMMA

Neon

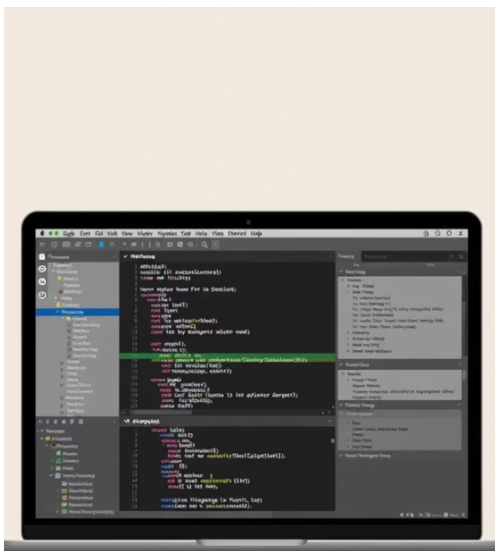
Опис використаних методів: Neon

Неон позиціонується як найшвидший фреймворк для глибокого навчання CNN, підтримує обчислення на GPU і CPU. Його фронтенд на Python керує високопродуктивними алгоритмами, написаними на шейдерному асемблері.

Розроблений компанією Nervana Systems, пізніше придбаною Intel, Neon оптимізований для ефективного використання апаратних ресурсів в задачах комп'ютерного зору та ІІІ.

8

Made with GAMMA



Опис використаних методів: MatConvNet

MatConvNet — це MATLAB-бібліотека для конволюційних нейронних мереж з відкритим кодом. Вона пропонує простий і гнучкий інтерфейс для побудови та тренування CNN без необхідності вивчати складні деталі реалізації.

Використання MatConvNet дозволяє швидко тестувати різні архітектури CNN та експериментувати з різними шарами та параметрами в середовищі MATLAB.

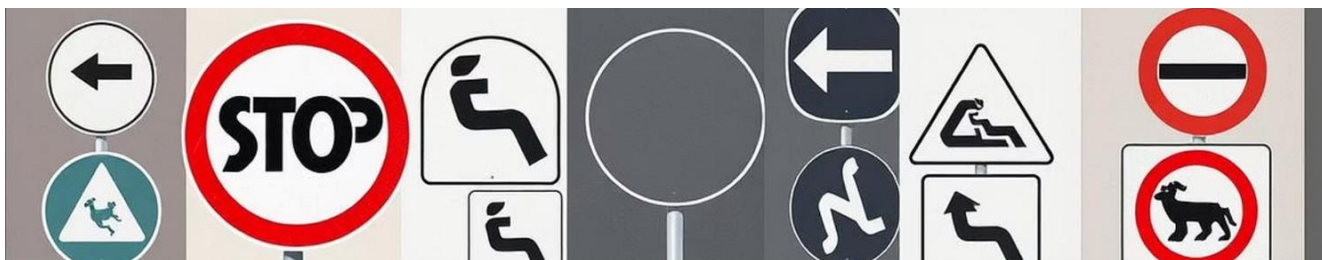
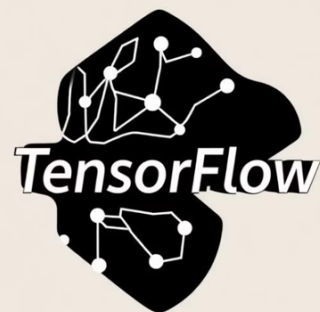
7

Made with GAMMA

Опис використаних методів: TensorFlow

TensorFlow — це провідна відкрита бібліотека для машинного навчання, розроблена Google. Вона підтримує побудову, навчання та розгортання нейронних мереж з високою продуктивністю на CPU, GPU та TPU.

Застосовується в галузях комп'ютерного зору, обробки мови і автоматизації, TensorFlow забезпечує масштабованість для наукових досліджень і комерційних продуктів.



Підбір наборів даних

Для тренування системи було обрано датасет Open Images — один з найбільших у світі наборів із приблизно 9.2 мільйонами анотованих зображень. Він містить різноманітні дорожні знаки в різних умовах.

Використання такого масштабного і детального датасету дозволяє моделі навчитися розпізнавати навіть складні та нетипові випадки в реальному оточенні.

Результати дослідження



Початкове зображення з кількома дорожніми знаками.



Результат роботи системи: точне розпізнавання та класифікація знайдених знаків.

Отримані результати демонструють високу точність і швидкість системи у розпізнаванні дорожніх знаків навіть на складних зображеннях з кількома об'єктами.

Це свідчить про практичну застосовність розробленої системи в реальних умовах руху.

11

Made with GAMMA



Система розпізнавання дорожніх знаків на основі нейронних мереж

Ця презентація присвячена розробці та впровадженню інтелектуальної системи розпізнавання дорожніх знаків на основі нейронних мереж. Ми розглянемо ключові етапи створення системи, від навчання моделі до її практичної реалізації та тестування, демонструючи її високу точність та надійність у динамічних умовах дорожнього руху.

12

Made with GAMMA

Актуальність та завдання дослідження

Автономне водіння



Системи розпізнавання знаків є фундаментом для безпечного функціонування безпілотних автомобілів, забезпечуючи їхній рух згідно з правилами дорожнього руху.

Безпека руху



Автоматична ідентифікація знаків значно знижує ризик ДТП, попереджаючи водіїв про обмеження швидкості, небезпечні ділянки та інші важливі зміни на дорозі.

Подолання недоліків



Існуючі комерційні системи часто вразливі до погодних умов або пошкоджень знаків, що вимагає розробки більш стійких та адаптивних рішень.

У сучасному світі, з розвитком автономного водіння та систем допомоги водію, критично важливим є забезпечення надійної та своєчасної інформації про дорожню обстановку. Автоматичне розпізнавання дорожніх знаків є ключовим аспектом, що дозволяє підвищити безпеку дорожнього руху, зменшити ризик ДТП, а також забезпечити комфортне та ефективне керування транспортним засобом. Існуючі комерційні системи розпізнавання дорожніх знаків мають певні недоліки, такі як чутливість до пошкоджень знаків, погодних умов, нестандартного розташування або тимчасових знаків. Це створює потребу в розробці більш стійких та точних рішень.