ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ 9
ВСТУП 11
1. КОНЦЕПТУАЛЬНІ ЗАСАДИ ТА КЛЮЧОВІ ПРИНЦИПИ РОЗПІЗНАВАННЯ РУХОМИХ ОБ'ЄКТІВ 13
1.1. Інтелектуальні системи та принципи їх функціонування 13
1.2. Методи виявлення та класифікації рухомих об'єктів у відеопотоці 20
1.3. Алгоритми обробки та аналізу зображень для розпізнавання руху 29
1.4. Апаратні платформи та технічні засоби для побудови систем комп'ютерного зору 37
2. ТЕХНОЛОГІЇ ТА ПІДХОДИ ДО ПОБУДОВИ СИСТЕМИ РОЗПІЗНАВАННЯ РУХОМИХ ОБ'ЄКТІВ 45
2.1. Підходи до побудови систем комп'ютерного зору та їх порівняльний аналіз 45
2.2. Методи оптимізації обробки відеоданих у реальному часі 51
2.3. Вимоги до точності, масштабованості та надійності систем розпізнавання руху 62
3. РОЗРОБКА ТА ВПРОВАДЖЕННЯ СИСТЕМИ РОЗПІЗНАВАННЯ РУХОМИХ ОБ'ЄКТІВ 71
3.1. Проєктування архітектури та вибір алгоритмічного забезпечення 71
3.2. Реалізація програмного модуля розпізнавання рухомих об'єктів 79
3.3. Тестування, оцінка ефективності та формування рекомендацій 87
ВИСНОВКИ 96
ПЕРЕЛІК ПОСИЛАНЬ 98
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ 102

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

AVX	Advanced Vector Extensions (Розширені векторні інструкції)
CNN / ЗНМ	Convolutional Neural Networks (Згорткові нейронні мережі)
CPU	Central Processing Unit (Центральний процесор)
CSI	Camera Serial Interface (Послідовний інтерфейс камери)
CUDA	Compute Unified Device Architecture (Архітектура паралельних обчислень)
DVFS	Dynamic Voltage and Frequency Scaling (Динамічне регулювання напруги і частоти)
FPGA	Field-Programmable Gate Array (Програмована користувачем вентильна матриця)
FPS	Frames Per Second (Кадрів за секунду)
GLCM	Gray Level Co-occurrence Matrix (Матриця суміжності рівнів сірого)
GMM	Gaussian Mixture Models (Моделі гауссової суміші)
GPU	Graphics Processing Unit (Графічний процесор)
HDD	Hard Disk Drive (Накопичувач на жорстких магнітних дисках)
HEVC	High Efficiency Video Coding (Високоефективне кодування відео)
HMM	Hidden Markov Models (Приховані марковські моделі)
HOG	Histogram of Oriented Gradients (Гістограма орієнтованих градієнтів)
IoT	Internet of Things (Інтернет речей)
LBP	Local Binary Patterns (Локальні бінарні шаблони)
MOT	Multiple Object Tracking (Відстеження кількох об'єктів)
NAS	Network Attached Storage (Мережеве сховище даних)
NPU	Neural Processing Unit (Нейронний процесор)

PCA	Principal Component Analysis (Аналіз головних компонентів)
PoE	Power over Ethernet (Живлення через Ethernet)
QoS	Quality of Service (Якість обслуговування)
RAM	Random Access Memory (Оперативна пам'ять)
ROI	Region of Interest (Область інтересу / Область виділення)
RTP	Real-time Transport Protocol (Транспортний протокол реального часу)
RTSP	Real-Time Streaming Protocol (Потоковий протокол реального часу)
SIFT	Scale-Invariant Feature Transform (Масштабно-інваріантне перетворення ознак)
SIMD	Single Instruction, Multiple Data (Одна інструкція — багато даних)
SSD	Solid State Drive (Твердотільний накопичувач)
SVM	Support Vector Machine (Метод опорних векторів)
TPU	Tensor Processing Unit (Тензорний процесор)
USB	Universal Serial Bus (Універсальна послідовна шина)
WDR	Wide Dynamic Range (Широкий динамічний діапазон)

ВСТУП

Актуальність теми. У сучасному світі інтелектуальні системи стали невід'ємною складовою технологічного прогресу, революціонізуючи такі сфери, як транспорт, безпека та робототехніка. Ключовою галуззю штучного інтелекту є комп’ютерний зір, метою якого є надання машинам здатності інтерпретувати візуальну інформацію та приймати рішення в автономному режимі. Здатність точно виявляти та класифікувати рухомі об’єкти у відеопотоці є критично важливою для функціонування систем відеоспостереження, автономних транспортних засобів та моніторингу дорожнього руху.

Проте, незважаючи на значний прогрес, існуючі методи часто стикаються з проблемами стабільності роботи в динамічних середовищах. Фактори, такі як зміни освітлення, тіні, шуми сенсорів та перекриття об’єктів, суттєво впливають на точність детекції. Потреба у високоякісному відеоаналізі з низькою затримкою вимагає впровадження ефективних стратегій оптимізації та розробки надійних алгоритмічних рішень. У цьому контексті розробка системи, що поєднує адаптивні методи обробки зображень з ефективними алгоритмами відстеження, є актуальним науково-практичним завданням.

Метою роботи є розробка та впровадження програмної системи розпізнавання рухомих об’єктів, яка забезпечує високу точність виявлення та стабільність роботи в режимі реального часу для використання в інтелектуальних системах.

Об’єкт дослідження — процес автоматизованого аналізу відеопотоків та розпізнавання образів в інтелектуальних системах.

Предмет дослідження — методи, алгоритми та програмні засоби для виявлення, класифікації та відстеження рухомих об’єктів у реальному часі.

Методи дослідження. У роботі використано методи цифрової обробки зображень (фільтрація, нормалізація) для покращення якості вхідних даних; метод віднімання фону (Background Subtraction) для детекції руху; морфологічні операції

для сегментації об’єктів; фільтр Калмана для прогнозування траєкторій та відстеження.

Наукова новизна одержаних результатів полягає у вдосконаленні підходу до розпізнавання руху шляхом інтеграції адаптивної попередньої обробки та морфологічної фільтрації в єдиний конвеєр з алгоритмом віднімання фону. Це дозволило підвищити точність розпізнавання до 92% та стійкість до шумів до 85% порівняно з базовими методами.

Практичне значення одержаних результатів. Розроблений програмний модуль може бути використаний як базовий компонент у системах інтелектуального відеоспостереження, управління дорожнім рухом та автономної навігації. Запропоновані рекомендації щодо конфігурації та оптимізації дозволяють адаптувати систему до конкретних умов експлуатації, забезпечуючи баланс між швидкодією та точністю.

1. КОНЦЕПТУАЛЬНІ ЗАСАДИ ТА КЛЮЧОВІ ПРИНЦИПИ РОЗПІЗНАВАННЯ РУХОМИХ ОБ’ЄКТІВ

1.1. Інтелектуальні системи та принципи їх функціонування

Інтелектуальні системи стали невід'ємною частиною сучасних технологій, революціонізуючи такі галузі, як охорона здоров'я, фінанси, транспорт і зв'язок. Ці системи розроблені для імітації когнітивних функцій людини, дозволяючи машинам виконувати завдання, які традиційно вимагали людського інтелекту, такі як навчання, міркування, вирішення проблем і прийняття рішень. Суть інтелектуальних систем полягає в їхній здатності обробляти складні дані, адаптуватися до нової інформації та працювати автономно або напівавтономно в динамічних середовищах. Щоб всебічно зрозуміти їх функціонування, важливо дослідити їхні фундаментальні характеристики, архітектурні рамки, що підтримують їхню роботу, та критичні процеси, пов'язані з обробкою даних (рис. 1.1) [1].



Рис. 1.1. Типи інтелектуальних систем

Суть інтелектуальних систем полягає в їхній здатності імітувати аспекти людського пізнання, що дозволяє їм виконувати завдання, що потребують сприйняття, міркування та навчання. Ці системи зазвичай класифікуються на кілька категорій залежно від їхньої складності та сфери застосування. Наприклад, експертні системи на основі правил використовують попередньо визначені правила для прийняття рішень, які часто використовуються в діагностичних або консультативних ролях. Системи машинного навчання, з іншого боку, навчаються на шаблонах даних для покращення продуктивності з часом, такі як інструменти розпізнавання зображень або платформи прогнозної аналітики. Більш просунуті приклади включають архітектури глибокого навчання, які імітують нейронні мережі, що дозволяє розпізнавати складні шаблони та автономне прийняття рішень. Ключові властивості, що відрізняють інтелектуальні системи, включають адаптивність, яка дозволяє їм змінювати свою поведінку у відповідь на нові дані; масштабованість, яка дозволяє їм обробляти зростаючі обсяги або складність даних; стійкість, яка забезпечує надійну роботу, незважаючи на невизначеності або помилки; та автономність, яка мінімізує втручання людини. Ці властивості разом дозволяють інтелектуальним системам ефективно працювати в різноманітних та непередбачуваних середовищах, прикладами яких є автономні транспортні засоби, які інтерпретують сенсорні дані для безпечної навігації, або віртуальні помічники, які адаптуються до індивідуальних уподобань користувача.

Архітектура інтелектуальних систем – це багаторівнева структура, що складається як з апаратних, так і з програмних компонентів, які працюють синергетично для забезпечення інтелектуальної поведінки. Апаратна архітектура зазвичай включає процесорні блоки, такі як центральні або графічні процесори, модулі пам'яті для зберігання даних та інструкцій, датчики для взаємодії з навколишнім середовищем та виконавчі механізми для виконання фізичних дій. Наприклад, автономний дрон поєднує датчики, такі як камери та лідар, з бортовими процесорними блоками для сприйняття та реагування на навколишнє середовище. З програмної сторони різні модулі взаємодіють у межах логічної організації, яка підтримує функції обробки даних, міркування та навчання.

Основні модулі включають інтерфейси збору даних, бази знань, механізми логічного висновку, алгоритми машинного навчання та компоненти інтерфейсу користувача. Взаємодія між цими модулями включає складні інформаційні потоки, де необроблені дані спочатку збираються, попередньо обробляються, а потім використовуються для розпізнавання образів або процесів прийняття рішень (рис. 1.2) [2].

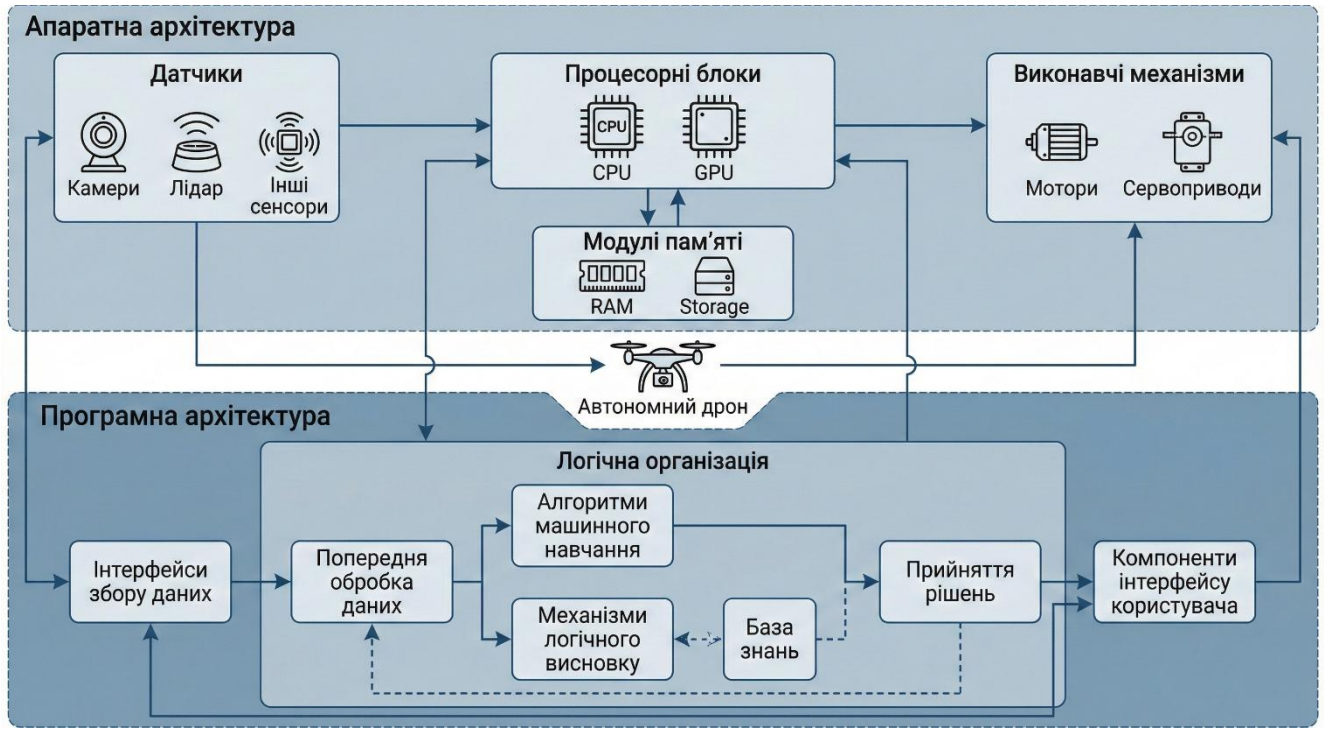


Рис. 1.2. Приклад архітектури інтелектуальних систем

Логічна організація забезпечує ефективну обробку даних, що дозволяє реагувати в режимі реального часу в таких додатках, як роботизовані системи керування або інтелектуальні мережі спостереження. Крім того, архітектура підкреслює модульність, щоб полегшити оновлення системи та масштабованість, як це видно в хмарних платформах штучного інтелекту, які розподіляють робочі навантаження між кількома серверами для підвищення продуктивності та надійності. Критичним аспектом інтелектуальних систем є збір та попередня обробка даних, що безпосередньо впливає на їхню точність та ефективність.

Основні методи збору даних включають датчики, камери, мікрофони та інші пристрої введення, які збирають необроблену інформацію з фізичного середовища або цифрових джерел. Наприклад, носимі медичні пристрої постійно контролюють життєво важливі показники, надаючи дані в режимі реального часу для медичного аналізу. Після отримання ці дані часто потребують очищення та нормалізації для забезпечення узгодженості та якості. Алгоритми очищення виявляють та виправляють помилки, видаляють дублікати та обробляють відсутні значення, що має вирішальне значення для запобігання неточностям на наступних етапах обробки. Потім алгоритми нормалізації стандартизують формати та масштаби даних, забезпечуючи змістовні порівняння та аналіз. Крім того, для усунення шуму та згладжування коливань використовуються такі методи фільтрації, як фільтри Калмана або медіанні фільтри, що особливо важливо в таких додатках, як розпізнавання мовлення або інтерпретація даних датчиків. Ці кроки попередньої обробки є фундаментальними, оскільки вони безпосередньо впливають на можливості системи приймати рішення, гарантуючи, що наступні процеси навчання та міркування базуються на надійних та високоякісних даних. Наприклад, в автономних транспортних засобах точна попередня обробка даних датчиків є важливою для точного виявлення об'єктів та безпечної навігації, що підкреслює важливість надійних методів обробки даних.

Фундаментальний компонент інтелектуальних систем включає їхні аналітичні механізми, які дозволяють їм інтерпретувати складні дані та отримувати змістовні висновки. Одним з основних методів, що використовуються в цих системах, є вилучення ознак, де відповідні атрибути визначають та виділяють із необроблених даних для сприяння ефективнішому аналізу. Такі методи, як аналіз головних компонентів (PCA), вейвлет-перетворення та банки фільтрів, зазвичай використовуються для зменшення розмірності та виділення важливих ознак, тим самим спрощуючи наступні етапи обробки. Після вилучення ознак алгоритми оцінюють стан об'єкта або події, оцінюючи ці атрибути за встановленими критеріями або моделями. Наприклад, у медичній діагностиці алгоритми класифікації аналізують дані пацієнтів, щоб визначити наявність або

відсутність захворювань, тоді як у промисловому моніторингу алгоритми розпізнавання образів контролюють стан обладнання для прогнозування збоїв. Крім того, підходи до інтерпретації навколишнього середовища включають інтеграцію даних з багатьох джерел, таких як візуальні, слухові та сенсорні вхідні дані, та застосування алгоритмів міркування, які можуть контекстуалізувати інформацію в ширшому середовищі. Такі методи, як нечітка логіка, байєсівські мережі та моделі глибокого навчання, дозволяють системам ефективно інтерпретувати неоднозначні або неповні дані, забезпечуючи всебічне розуміння динамічних сценаріїв. Ці аналітичні механізми є важливими для того, щоб інтелектуальні системи могли виконувати складні завдання, такі як моніторинг у режимі реального часу, прогнозне обслуговування та адаптивне керування (табл. 1.1) [3].

Таблиця 1.1.

Класифікація інтелектуальних систем та їхні характеристики

Тип інтелектуальної системи	Характеристика	Приклади застосування
Експертні системи	Використовують базу знань та правила логічного висновку	Медична діагностика, технічна підтримка
Системи комп’ютерного зору	Аналізують візуальну інформацію, виявляють об’єкти та події	Відеонагляд, робототехніка
Системи підтримки прийняття рішень	Узагальнюють дані для оцінювання ситуацій	Логістика, фінансове планування
Автономні системи	Самостійно виконують завдання у змінних умовах	Дрони, автономні транспортні засоби
Периферійні інтелектуальні системи	Обробляють дані на краю мережі з низькою затримкою	IoT-пристрої, системи відеоаналітики

Визначною особливістю інтелектуальних систем є їхня здатність до адаптації та самонавчання, що дозволяє їм ефективно працювати в умовах змін без постійного втручання людини. Системи досягають адаптивності за допомогою різних методів, включаючи алгоритми онлайн-навчання, навчання з підкріпленням та навчання з перенесенням, які дозволяють постійно оновлювати моделі на основі нових даних. Наприклад, автономні транспортні засоби використовують навчання з підкріпленням для покращення стратегій навігації, отримуючи зворотний зв'язок від свого середовища, тим самим удосконалюючи процес прийняття рішень з часом. Процеси самокорекції є невід'ємною частиною підтримки точності системи; Алгоритми розроблені для виявлення помилок або невідповідностей у своїх вихідних даних та відповідного коригування параметрів. Такі методи, як зворотне поширення помилок у нейронних мережах або байєсівське оновлення, допомагають системам точно налаштувати свої внутрішні моделі, що призводить до точніших результатів. Крім того, методи підвищення точності та стабільності включають ансамблеве навчання, де кілька моделей співпрацюють для покращення прогнозів, та методи регуляризації, які запобігають перенавчанню. Ці підходи разом сприяють розробці надійних, стійких систем, здатних навчатися на досвіді, адаптуватися до нових умов та підтримувати високу продуктивність у невизначених або мінливих середовищах, таких як адаптивні системи кібербезпеки, які виявляють нові загрози, або персоналізовані механізми рекомендацій, які розвиваються разом із уподобаннями користувача.

Процеси прийняття рішень в інтелектуальних системах базуються на різноманітних моделях та стратегіях, які дозволяють здійснювати ефективні та контекстно-залежні дії. Логічні моделі, такі як системи на основі правил та дерева рішень, використовують явні правила та ієрархічні структури для імітації процесів міркування, сприяючи прозорим та зрозумілим рішенням. Ці моделі особливо корисні в областях, що вимагають суворого дотримання заздалегідь визначених протоколів, таких як правові або критично важливі для безпеки системи. Доповненням до логічних підходів є статистичні та евристичні стратегії, які спираються на ймовірнісні міркування, аналітичні висновки, засновані на даних,

та евристику, засновану на досвіді, для прийняття рішень в умовах невизначеності. Такі методи, як байєсівський висновок, моделювання Монте-Карло та нечітка логіка, дозволяють системам зважувати різні фактори та оцінювати ймовірність різних результатів, тим самим спрямовуючи оптимальні дії. Формування та передача керуючих дій передбачає перетворення цих рішень на виконувані команди, часто за допомогою алгоритмів керування, які керують виконавчими механізмами або протоколами зв'язку. Наприклад, в автономній робототехніці вихідні дані рішень визначають рухи та операційні реакції, тоді як у фінансових торгових системах вони диктують дії купівлі або продажу на основі аналізу ринку. Ці рамки прийняття рішень мають вирішальне значення для забезпечення автономної роботи інтелектуальних систем, координації складної поведінки та динамічної адаптації своїх дій у відповідь на зворотний зв'язок з навколишнім середовищем.

Інтеграція інтелектуальних систем у складні рішення вимагає безперебійної взаємодії з існуючою телекомунікаційною інфраструктурою, оскільки ці системи часто залежать від передачі даних у режимі реального часу та віддаленого підключення. Ефективні комунікаційні мережі дозволяють інтелектуальним системам отримувати, передавати та обробляти величезні обсяги даних у розсіяних місцях, що життєво важливо для таких застосувань, як розподілені сенсорні мережі, інтелектуальні мережі та підключені транспортні засоби. З появою периферійних обчислень деякі завдання обробки виконуються локально на пристроях або шлюзах, що зменшує затримку та вимоги до пропускної здатності, тоді як хмарні обчислення пропонують значні ресурси для зберігання та обчислювальні ресурси для обробки великомасштабної аналітики даних та навчання моделей. Балансування між периферійною та хмарною обробкою дозволяє системам оптимізувати продуктивність, забезпечувати швидке реагування та підтримувати операційну ефективність. Сумісність та масштабованість є вирішальними факторами в таких інтеграціях; системи повинні бути розроблені зі стандартизованими протоколами та гнучкими архітектурами, щоб враховувати майбутні технологічні досягнення та зростаючі обсяги даних.

Наприклад, розумні міста розгортають взаємопов’язані інтелектуальні системи, що спираються на надійні телекомунікаційні та гібридні моделі обробки даних для ефективного управління дорожнім рухом, споживанням енергії та громадською безпекою. Забезпечення сумісності та модульності в проектуванні системи не лише підвищує експлуатаційну надійність, але й сприяє поступовим модернізаціям, дозволяючи складним рішенням розвиватися разом із технологічним прогресом та зростанням потреб [4].

Інтелектуальні системи характеризуються здатністю імітувати когнітивні функції людини за допомогою передової обробки даних, адаптивних архітектур та складних аналітичних механізмів. Їхні основні властивості, такі як адаптивність, стійкість та автономність, дозволяють їм ефективно функціонувати в різних середовищах, тоді як їхня структурна конструкція інтегрує апаратні та програмні компоненти для забезпечення безперебійної роботи. Процеси збору, попередньої обробки та аналізу даних формують основу їхніх можливостей прийняття рішень, які ще більше посилюються їхньою здатністю до самонавчання та адаптації. Більше того, інтеграція цих систем у складні рішення значною мірою залежить від ефективних комунікаційних інфраструктур, периферійних та хмарних обчислень, а також масштабованих, сумісних архітектур. У сукупності ці принципи лежать в основі розробки інтелектуальних систем, які не тільки здатні виконувати складні завдання, але й адаптуються до майбутніх викликів та технологічного розвитку. У міру того, як інтелектуальні системи продовжують розвиватися, їхня роль у трансформації галузей та покращенні життя людини стає дедалі глибшою, що підкреслює важливість постійних досліджень та інновацій у цій динамічній галузі.

1.2. Методи виявлення та класифікації рухомих об’єктів у відеопотоці

Швидкий розвиток відеотехнологій та комп'ютерного зору значно покращив можливості виявлення та класифікації рухомих об'єктів у відеопотоках. Ці можливості є важливими для численних застосувань, включаючи спостереження,

автономні транспортні засоби, моніторинг дорожнього руху та взаємодію людини з комп'ютером. Ефективне виявлення та класифікація залежать від різних методів, які аналізують рух та сегментують рухомі об'єкти з фону, а потім від алгоритмів відстеження, які зберігають ідентичність з часом.

Виявлення руху слугує фундаментальним кроком у виявленні рухомих об'єктів у відеопотоці, і для досягнення цієї мети було розроблено кілька підходів. Одним із найтрадиційніших методів є віднімання фону, яке передбачає створення моделі статичної сцени, а потім виявлення відхилень, що свідчать про рух. Наприклад, у системах спостереження статичну модель фону можна віднімати з кожного кадру, щоб виділити рухомі об'єкти, такі як пішоходи або транспортні засоби. Цей метод є дуже ефективним у контрольованих середовищах, але може мати труднощі з динамічним фоном або змінами умов освітлення (рис. 1.2)

Таблиця 1.2.

Технічні характеристики методів виявлення та класифікації рухомих об’єктів

Метод	Особливість	Характеристика
Метод фонової субстракції (Background Subtraction)	Чутливість до шумів	Висока, потребує фільтрації
	Стабільність фону	Потребує адаптивного оновлення
	Обчислювальна складність	Низька
Оптичний потік (Optical Flow)	Точність оцінки руху	Висока
	Потреба в обчисленнях	Висока (векторні обчислення)
	Чутливість до освітлення	Значна
Методи часової різниці (Temporal Differencing)	Швидкість роботи	Дуже висока
	Якість дрібних об’єктів	Середня
	Залежність від стабільності камери	Висока
Сегментація регіонів (Region-Based Segmentation)	Стійкість до тіней	Середня
	Точність відокремлення об’єктів	Висока
	Необхідність попередньої обробки	Значна

Класифікація за ознаками (Feature-Based Classification)	Тип ознак	Геометричні та текстурні
	Точність	Залежить від набору ознак
	Стійкість до шумів	Висока

Доповненням до віднімання фону є аналіз зміни інтенсивності, який досліджує зміни інтенсивності пікселів з часом для виявлення руху; це особливо корисно в сценаріях, коли модель фону важко підтримувати. Методи оптичного потоку пропонують більш детальну оцінку динаміки сцени шляхом обчислення видимого руху яскравих шаблонів у послідовності зображень. Наприклад, алгоритми Лукаса-Канаде або Фарнебака оцінюють вектори руху для кожного пікселя, що дозволяє виявляти ледь помітні рухи та спрямованість руху, що є цінним у таких додатках, як аналіз спортивних подій або моніторинг потоку дорожнього руху. Крім того, методи, засновані на різниці в часі між послідовними кадрами, аналізують попіксельні відмінності для виявлення руху, забезпечуючи простий, але ефективний підхід у системах реального часу завдяки своїй обчислювальній ефективності. Разом ці методи утворюють універсальний інструментарій для виявлення руху, кожен з яких має сильні сторони, що відповідають певним умовам навколишнього середовища та обчислювальним обмеженням.

Після виявлення руху наступним важливим кроком є сегментація – виділення рухомих об'єктів з фону для полегшення подальшого аналізу. Методи сегментації спрямовані на визначення форми та обсягу рухомих об'єктів, причому виділення контурів та силуетів є основними підходами. Наприклад, після віднімання фону отриману бінарну маску можна уточнити для отримання чітких контурів об'єктів, які є важливими для точної класифікації. Методи сегментації на основі порогів використовують пороги інтенсивності або кольору для відділення об'єктів переднього плану від фону; наприклад, встановлення порогу різниці пікселів може допомогти ізолювати рухомі об'єкти в сцені з відносно однорідним фоном. З іншого боку, методи регіональної сегментації аналізують просторову когерентність груп пікселів, такі як алгоритми розширення областей або

кластеризації, для сегментації об'єктів на основі критеріїв подібності. Ці методи особливо ефективні при роботі зі складними або захараченими фонами. Для покращення якості сегментації використовуються морфологічні операції, такі як дилатація, ерозія, відкриття та закриття, для видалення шуму, заповнення прогалин та згладжування меж об'єктів. Наприклад, застосування дилатації може з'єднати фрагментовані частини об'єкта, тоді як ерозія може усунути невеликі, нерелевантні області, спричинені шумом. Ці комбіновані методи підвищують стійкість та точність сегментації об'єктів, забезпечуючи надійну основу для подальшого відстеження.

Заключний етап процесу класифікації рухомих об'єктів включає відстеження їхніх траєкторій з часом, що вимагає складних методів вилучення ознак та асоціації. Методи кореляційного відстеження працюють шляхом зіставлення зовнішнього вигляду об'єктів у різних кадрах, часто використовуючи зіставлення шаблонів або нормалізовану взаємну кореляцію для збереження ідентичності об'єкта. Наприклад, у переповненій сцені ці методи можуть відстежувати окремих пішоходів шляхом кореляції їхніх візуальних ознак протягом послідовних кадрів. Фільтри Калмана широко використовуються для оцінки стану в програмах відстеження, забезпечуючи ймовірнісну структуру, яка прогнозує майбутнє положення об'єкта на основі його попереднього руху, таким чином враховуючи шум та невизначеності вимірювань. Часткові фільтри, такі як частинкові фільтри, розширюють ці можливості, обробляючи нелінійні та негаусові моделі руху, що робить їх придатними для складних сценаріїв відстеження, таких як автономне керування або навігація дронів. Відстеження кількох об'єктів створює додаткову складність, одночасно відстежуючи кілька об'єктів та забезпечуючи правильне збереження ідентичностей без плутанини, особливо коли об'єкти перетинаються або закриваються. Такі алгоритми, як угорський алгоритм для асоціації даних або моделі повторної ідентифікації на основі глибокого навчання, часто використовуються для вирішення неоднозначностей, забезпечуючи надійне та безперервне відстеження. Ці методи відстеження в поєднанні з методами вилучення ознак дозволяють класифікувати та аналізувати поведінку рухомих

об'єктів у режимі реального часу, що лежить в основі ефективності систем спостереження та автономних платформ.

Надамо ілюстрацію методу віднімання фону (Background Subtraction). Цей метод описаний як один із найтрадиційніших фундаментальних підходів для виявлення руху.

Рис. 1.3. представлений у вигляді послідовності з трьох зображень, що показують процес обробки:

Принцип роботи методу віднімання фону

1. Вхідне зображення (Поточний кадр відеопотоку) Це зображення сцени, на якому присутній рухомий об'єкт (наприклад, пішохід) на тлі статичного оточення.

2. Модель статичного фону Це зображення тієї самої сцени, але повністю без рухомих об'єктів. Це створена «модель статичної сцени», яка використовується як еталон для порівняння з поточним кадром.

3. Результат (Виявлений рух) Це зображення отримане в результаті математичної операції віднімання «Моделі статичного фону» від «Вхідного зображення». На цьому зображенні виділено (білим кольором на чорному тлі) лише ті області, де є відхилення, що свідчать про рух, тобто сам рухомий об'єкт. Цей результат представляє собою бінарну маску об'єкта, яка може бути використана для подальшої сегментації та аналізу.







Рис. 1.3. Етапи виділення рухомих об'єктів шляхом віднімання статичної моделі фону

Алгоритми класифікації об'єктів життєво важливі для розрізнення різних типів рухомих об'єктів у відеопотоці, що дозволяє системам більш осмислено інтерпретувати вміст сцени. Один поширений підхід включає аналіз форми, текстури та розміру сегментованих об'єктів. Наприклад, у моніторингу дорожнього руху транспортні засоби можна класифікувати на основі їх геометричних розмірів та візерунків поверхні, таких як розрізнення автомобілів від вантажівок чи мотоциклів. Аналіз текстур, що використовує такі методи, як матриці співпояву рівнів сірого (GLCM), допомагає диференціювати об'єкти з різними властивостями поверхні, тоді як метрики розміру допомагають фільтрувати шум або нерелевантні виявлення. Доповнюють ці методи, засновані на ознаках, моделі, засновані на статистичних правилах, які використовують ймовірнісні структури для призначення класів об'єктів на основі спостережуваних ознак. Ці моделі можуть включати баєсівські класифікатори або приховані марківські моделі (HMM), які вивчають характерні закономірності з навчальних даних, а потім відповідно класифікують нові об'єкти. Крім того, використання попередніх шаблонів об'єктів підвищує точність класифікації; зберігаючи шаблони відомих зовнішніх виглядів об'єктів, системи відстеження можуть порівнювати поточні виявлення з цими шаблонами, тим самим покращуючи розпізнавання навіть за часткового перекриття або різних точок зору. Наприклад, у системах спостереження повторна ідентифікація особи або транспортного засобу

в різних кадрах значною мірою залежить від зіставлення поточних спостережень зі збереженими шаблонами, використовуючи як ознаки зовнішнього вигляду, так і контекстні підказки. Ці комбіновані підходи забезпечують надійнішу та нюансованішу класифікацію, що є важливим для подальших завдань, таких як поведінковий аналіз або автоматична генерація сповіщень.

Покращення точності виявлення залишається постійною проблемою, особливо у складних або галасливих середовищах. Один з фундаментальних методів включає зменшення шуму відеопотоку, який може виникати через недосконаліть датчиків, погане освітлення або порушення навколишнього середовища. Такі методи, як часова фільтрація, просторове згладжування та адаптивні алгоритми шумозаглушення, допомагають зменшити ці проблеми, що призводить до отримання чистіших зображень, що сприяють точнішому виявленню та сегментації руху. Наприклад, застосування фільтрів гаусового згладжування може усунути високочастотний шум, запобігаючи хибним спрацьовуванням при виявленні руху. Окрім зменшення шуму, інтеграція кількох методів виявлення та класифікації в гібридні схеми підвищує надійність. Поєднання віднімання фону з класифікаторами оптичного потоку та машинного навчання створює багатошарову систему, яка компенсує обмеження окремих методів. Наприклад, якщо віднімання фону має труднощі через динамічний фон, оптичний потік може забезпечити додаткові сигнали руху, тоді як класифікатор уточнює ідентифікацію об'єктів. Крім того, адаптивні параметри, налаштовані на умови сцени, такі як динамічне коригування порогу на основі освітлення або інтенсивності руху, дозволяють алгоритмам виявлення підтримувати продуктивність, незважаючи на зміни навколишнього середовища. Адаптивні схеми використовують механізми зворотного зв'язку для повторного калібрування параметрів у режимі реального часу, забезпечуючи стабільну точність. Наприклад, у зовнішньому спостереженні, де освітлення змінюється протягом дня, адаптивні алгоритми можуть автоматично налаштовувати чутливість, зменшуючи кількість пропущених виявлень або хибних тривог та підтримуючи високу надійність.

Оцінка характеристик руху виявлених об'єктів забезпечує глибше розуміння їхньої поведінки та контексту в межах сцени. Визначення швидкості та напрямку руху об'єкта є фундаментальним для таких застосувань, як управління дорожнім рухом, де розуміння потоку транспортних засобів є важливим. Такі методи, як аналіз оптичного потоку, у поєднанні з даними відстеження, дозволяють розраховувати вектори швидкості, показуючи не лише швидкість руху об'єкта, але й його напрямок. Наприклад, в автономному водінні точна оцінка траєкторії руху навколишніх транспортних засобів допомагає приймати обґрунтовані навігаційні рішення та системи запобігання зіткненням. Ідентифікація траєкторії об'єкта включає реконструкцію шляху, яким об'єкт рухається протягом кількох кадрів, що може бути досягнуто за допомогою прогнозних моделей, таких як фільтри Калмана або фільтри частинок. Ці моделі інтерполюють та згладжують траєкторії, навіть з періодичними перешкодами або шумом вимірювання, що призводить до надійних шляхів руху. Аналіз поведінки руху також включає оцінку таких закономірностей, як прискорення, уповільнення або різкі зміни напрямку. Такий аналіз має вирішальне значення в контексті безпеки, наприклад, для виявлення нестабільних рухів, які можуть свідчити про підозрілу активність. Розширений аналіз поведінки руху може включати методи машинного навчання, навчені розпізнавати типові моделі руху та позначати аномалії. У сукупності ці методи дозволяють всебічно зрозуміти динамічні елементи сцени, підтримуючи проактивні реакції та інтелектуальне прийняття рішень у застосунках реального часу [5].

Незважаючи на значний прогрес у методах виявлення та класифікації рухомих об'єктів у відеопотоках, залишається кілька обмежень та проблем, які можуть перешкоджати ефективності системи. Однією з основних проблем є мінливість умов освітлення та наявність... тіні, які часто призводять до помилкових виявлень або пропусків об'єктів. Наприклад, раптові зміни освітлення, такі як мерехтіння сонячного світла або коливання штучного освітлення, можуть змінювати інтенсивність пікселів та порушувати моделі фону, що призводить до неточного виявлення руху. Тіні, що відкидаються рухомими

об'єктами, можуть бути помилково ідентифіковані як окремі сутності, що ускладнює процеси сегментації та відстеження. Перекриття об'єктів створює ще одну суттєву проблему, особливо в переповнених сценах, де кілька рухомих об'єктів, таких як пішоходи або транспортні засоби, перетинаються або затуляють один одного. Під час перекриття традиційні алгоритми відстеження можуть втрачати відстеження окремих об'єктів або плутати ідентичності, що призводить до фрагментації траєкторій або неправильної класифікації. Крім того, якість відео та частота кадрів суттєво впливають на точність виявлення та можливості обробки в режимі реального часу. Відео з низькою роздільною здатністю або ті, що страждають від артефактів стиснення, можуть затьмарювати особливості об'єктів, знижуючи надійність класифікації. Аналогічно, низька частота кадрів може призвести до розмиття або пропускання швидкорухомих об'єктів між кадрами, що погіршує оцінку руху та прогнозування траєкторії. Ці проблеми підкреслюють необхідність постійних досліджень для розробки адаптивних, надійних алгоритмів, здатних обробляти складні реальні сценарії зі змінними умовами навколишнього середовища та якістю даних.

Виявлення та класифікація рухомих об'єктів у відеопотоках включає багатогранний набір методів, кожен з яких адаптований для вирішення конкретних аспектів аналізу сцени. Підходи до виявлення руху, такі як віднімання фону, оптичний потік та диференціація кадрів, служать основними методами, що ідентифікують динамічні області, тоді як методи сегментації уточнюють межі об'єктів для полегшення точного розпізнавання. Алгоритми відстеження, включаючи методи на основі кореляції та фільтри Калмана, дозволяють здійснювати безперервний моніторинг об'єктів з часом, при цьому вилучення ознак відіграє життєво важливу роль у диференціації та класифікації об'єктів на основі форми, текстури та розміру. Для підвищення надійності системи використовуються гібридні схеми виявлення та адаптивні налаштування параметрів для зменшення мінливості навколишнього середовища та шуму. Оцінка характеристик руху ще більше збагачує розуміння сцени, що дозволяє використовувати такі програми, як аналіз потоку транспорту та

відеоспостереження. Усунення цих обмежень за допомогою інноваційних алгоритмів та покращеного збору даних матиме вирішальне значення для розвитку цієї галузі. Загалом, інтеграція цих методів формує комплексну основу, яка лежить в основі ефективного аналізу рухомих об'єктів, а поточні дослідження обіцяють подолати існуючі проблеми та розширити потенціал застосування в різноманітних реальних сценаріях.

1.3. Алгоритми обробки та аналізу зображень для розпізнавання руху

У галузі комп'ютерного зору, що швидко розвивається, алгоритми обробки та аналізу зображень відіграють ключову роль, дозволяючи машинам інтерпретувати та розуміти рух у візуальних сценах. Розпізнавання руху, підмножина виявлення дій та активності, значною мірою залежить від надійних методів обробки зображень для точного аналізу динамічних середовищ. Розглядаючи методи та їх значення, ми можемо краще зрозуміти, як ці алгоритми сприяють застосуванню, починаючи від систем спостереження до автономних транспортних засобів, зрештою покращуючи нашу здатність інтерпретувати складні візуальні дані [6].

Початковий етап обробки зображень для розпізнавання руху включає ретельну попередню обробку зображення, метою якої є підготовка необроблених візуальних даних для подальшого аналізу. Одним з важливих аспектів є згладжування та зменшення шуму, що пом'якшує вплив артефактів, спричинених обмеженнями сенсора, умовами навколишнього середовища або помилками передачі. Такі методи, як гауссове розмиття, медіанна фільтрація та двостороння фільтрація, зазвичай використовуються для придушення випадкового шуму, зберігаючи при цьому важливі деталі. Наприклад, гауссове згладжування ефективно зменшує високочастотний шум, створюючи чистіше зображення, яке сприяє точному виявленню ознак. Корекція яскравості та контрастності додатково оптимізує якість зображення, регулюючи значення інтенсивності пікселів для забезпечення рівномірного освітлення по всіх кадрах, що має вирішальне значення

для послідовного аналізу, особливо у відкритих сценах з різними умовами освітлення. Вирівнювання гістограми часто використовується для посилення контрастності, роблячи ознаки більш помітними. Крім того, нормалізація кадру вирівнює зображення за загальним масштабом та орієнтацією, стандартизуючи дані, щоб забезпечити змістовне порівняння між послідовними кадрами. Цей крок є життєво важливим для динамічних сцен, де відбувається рух камери або зміни перспективи, оскільки нормалізація гарантує, що витягнуті ознаки відображають справжній рух, а не артефакти зміни точки зору. Разом ці методи попередньої обробки закладають міцну основу для надійного виявлення руху шляхом зменшення шуму, корекції нерівностей освітлення та стандартизації вхідних даних.

Після попередньої обробки вилучення ключових ознак є фундаментальним для розпізнавання та характеристики руху в сцені. Вилучення контурів та градієнтів служить основними методами для визначення меж об'єктів та визначення структур форми. Такі методи, як виявлення країв Канні, виявляють значні краї, аналізуючи зміни інтенсивності, що дозволяє системі точно визначати рухомі об'єкти (табл. 1.3.) [7].

Таблиця 1.3.

Технічні параметри алгоритмів обробки та аналізу зображень для розпізнавання руху

Алгоритм	Параметр	Характеристика
Фільтрація шумів (Noise Filtering)	Тип фільтра	Gaussian, Median, Bilateral
	Вплив на деталі	Згладжує дрібні структури, зменшує шуми
	Обчислювальна складність	Низька або середня
Виділення контурів (Edge Detection)	Метод	Sobel, Canny, Prewitt
	Стійкість до шумів	Висока у Canny
	Вимоги до обчислень	Середні
Сегментація кадру (Image Segmentation)	Метод сегментації	Thresholding, Watershed
	Точність відокремлення	Висока при чітких межах
	Попередні вимоги	Потребує фільтрації та

		нормалізації
Виділення ознак (Feature Extraction)	Тип ознак	HOG, LBP, SIFT
	Чутливість до масштабу	Висока у SIFT
	Ресурсоємність	Висока у SIFT, середня у HOG
Аналіз руху між кадрами (Motion Analysis)	Метод	Frame differencing, trajectory estimation
	Точність	Залежить від частоти кадрів
	Стійкість до тіней	Середня
Оцінка геометричних параметрів (Geometric Analysis)	Визначення форми	Bounding box, convex hull
	Точність форми	Середня або висока
	Обчислювальні вимоги	Низькі
Трекінг об’єктів (Object Tracking)	Метод	Kalman Filter, Meanshift
	Стійкість до перекриттів	Середня
	Обчислювальна складність	Низька у Kalman, середня у Meanshift

Градієнтні методи, такі як оператори Собеля, кількісно визначають швидкість зміни інтенсивності, допомагаючи у виявленні контурів об'єктів та їх орієнтації. Оцінка характеристик текстури додатково збагачує представлення ознак шляхом аналізу поверхневих візерунків у межах областей інтересу. Такі методи, як локальні бінарні візерунки (LBP) або фільтри Габора, кількісно визначають текстури, що дозволяє розрізняти об'єкти на основі властивостей поверхні, наприклад, розрізняти прапор, що розвівається, від статичного фону. Побудова надійних дескрипторів є важливою для представлення об'єктів; ці дескриптори, такі як гістограма орієнтованих градієнтів (HOG) або масштабно-інваріантне перетворення ознак (SIFT), фіксують важливі ознаки, інваріантні до масштабу, обертання або змін освітлення. Наприклад, у системах спостереження ці дескриптори допомагають відстежувати людей у різних кадрах, незважаючи на зміни пози чи освітлення. Поєднання контурних, градієнтних та текстурних характеристик забезпечує вичерпний опис рухомих об'єктів, що сприяє точному розпізнаванню руху навіть у складних сценах з перекриттями або шумом.

Ефективна сегментація та поділ сцени є критично важливими для ізоляції рухомих об'єктів від статичного фону, що дозволяє точно аналізувати рух. Методи сегментації за порогом, такі як глобальне або адаптивне порогове визначення, сегментують зображення на основі значень інтенсивності пікселів. Наприклад, у сцені з постійним освітленням простий поріг може відрізнити об'єкти переднього плану від фону, відокремлюючи пікселі вище або нижче певного рівня інтенсивності. Адаптивне порогове визначення додатково вдосконалює цей процес шляхом локального налаштування порогів, що особливо корисно в умовах нерівномірного освітлення. Алгоритми кластеризації пікселів, такі як K-means або Mean Shift, групують пікселі на основі спільних характеристик, таких як колір, інтенсивність або текстура. Ці алгоритми кластеризації можуть ідентифікувати когерентні області, що відповідають об'єктам, навіть коли вони демонструють незначні відмінності від фону. Наприклад, кластеризація K-means може сегментувати сцену на області переднього плану та фону, групуючи пікселі з подібними спектральними властивостями. Крім того, передові алгоритми, такі як моделі віднімання фону, такі як моделі гауссової суміші (GMM), розроблені спеціально для динамічних сцен, що дозволяє безперервне оновлення фону та точніше розділення рухомих об'єктів від статичних елементів. Це розділення є важливим для подальшого аналізу руху, оскільки воно ізолює відповідні об'єкти для відстеження та розпізнавання поведінки. Загалом, алгоритми сегментації сцени служать мостом, який перетворює необроблені піксельні дані на змістовні представлення об'єктів, необхідні для розуміння руху в складних середовищах (рис. 1.4) [8].

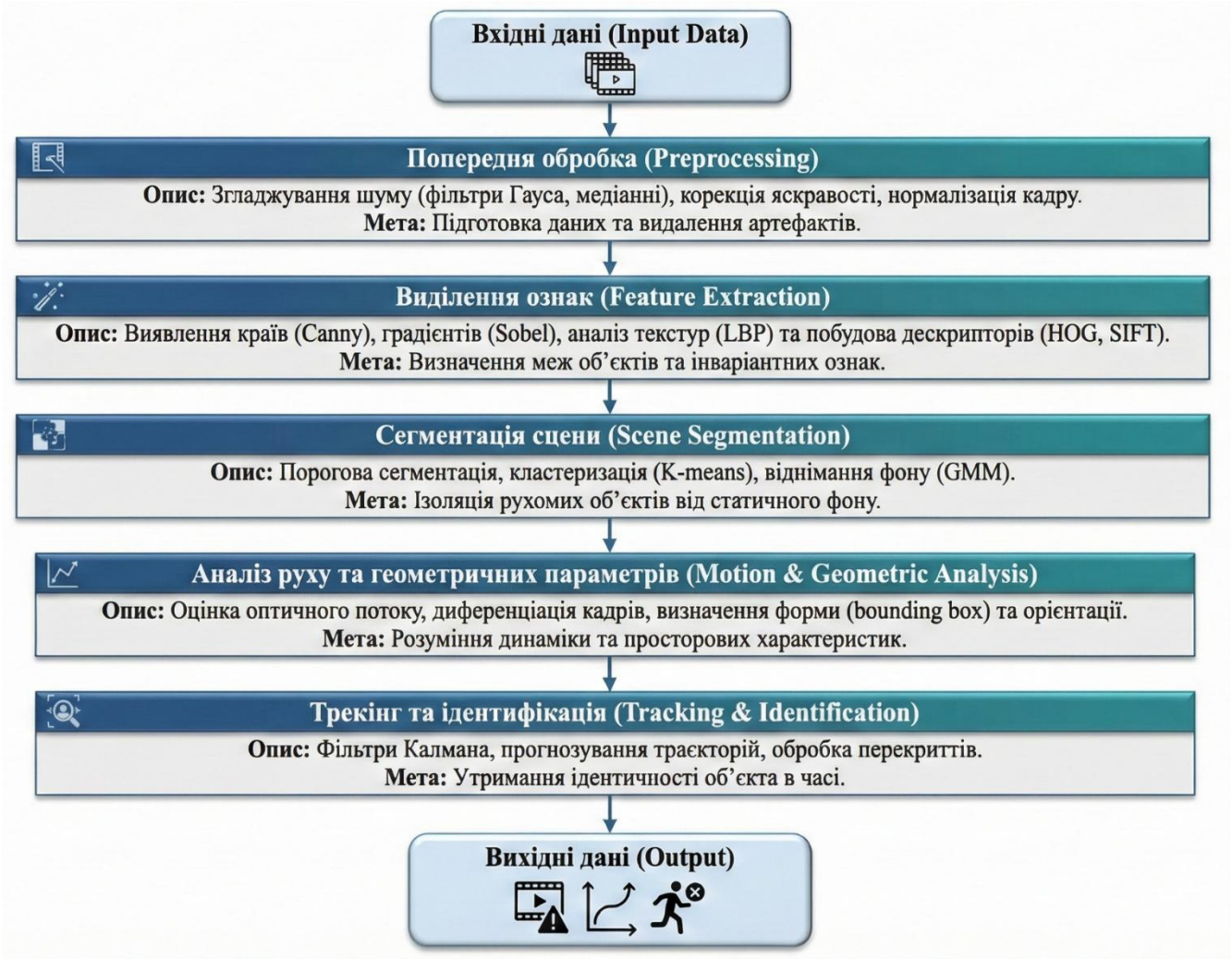


Рис. 1.4. Етапи обробки зображень для розпізнавання руху

Аналіз структурних властивостей є вирішальним етапом у розумінні руху шляхом вивчення властивих геометричних та просторових характеристик об'єктів у сцені. Оцінка геометричних параметрів включає аналіз форм, розмірів та орієнтацій об'єктів, які є фундаментальними для класифікації об'єктів та розуміння їхньої ролі в шаблони руху. Такі методи, як метод найменших квадратів та перетворення Хафа, часто використовуються для вилучення таких параметрів, як довжина, ширина, співвідношення сторін та орієнтація об'єктів, що забезпечує кількісні показники, що полегшують порівняння між кадрами. Визначення локальних форм та контурних структур ще більше покращує це розуміння, ідентифікуючи конкретні особливості форми, такі як ребра, кути та кривизна, які визначають форму об'єкта. Наприклад, аналіз кривизни контуру може розрізняти лінійні та округлі об'єкти, допомагаючи диференціювати пішоходів від

транспортних засобів на записах спостереження. Побудова просторових масок служить ефективним інструментом для зосередження аналізу на областях інтересу в межах сцени, що дозволяє цілеспрямовано досліджувати об'єкти, ігноруючи нерелевантні області фону. Ці маски можуть бути бінарними, що виділяють певні області об'єктів, або зваженими, що підкреслюють певні особливості для покращення подальшого аналізу. Просторові маски особливо корисні в складних сценах із зайвим шумом або перекриттями, оскільки вони допомагають ізолювати окремі об'єкти для детального структурного дослідження. Разом ці методи дозволяють системі отримувати детальні геометричні та просторові дані, які є важливими для розуміння природи рухомих об'єктів та їхньої взаємодії в межах сцени.

Розуміння динаміки руху з плином часу вимагає детальної оцінки часових властивостей, що включає аналіз того, як об'єкти змінюються та рухаються в послідовних кадрах. Виявлення змін між кадрами є фундаментальним для ідентифікації руху; такі методи, як диференціація кадрів та аналіз оптичного потоку, зазвичай використовуються для кількісної оцінки варіацій інтенсивності пікселів та векторів руху. Диференціація кадрів виділяє області, де відбуваються значні зміни, що вказує на рух об'єкта, тоді як оптичний потік забезпечує щільне векторне поле, що описує напрямок та величину руху в кожному пікселі. Моделювання часових залежностей включає створення представлень, які фіксують прогрес руху з часом, що дозволяє прогнозувати майбутні положення та розпізнавати шаблони руху. Рекурентні нейронні мережі та приховані марковські моделі часто використовуються для моделювання цих часових послідовностей, пропонуючи розуміння поведінки та траєкторії рухомих об'єктів. Визначення напрямку та швидкості руху досягається за допомогою алгоритмів оцінки швидкості, які аналізують потік пікселів або геометричні зміщення протягом кількох кадрів. Наприклад, обчислюючи вектори переміщення, отримані з оптичного потоку, система може зробити висновок, чи наближається, віддаляється чи рухається об'єкт убік, і з якою швидкістю. Точна оцінка цієї часової динаміки є життєво важливою для таких застосувань, як моніторинг дорожнього руху, де

розуміння потоку та швидкості транспортних засобів інформує про прийняття рішень та заходи безпеки. Після виявлення об'єктів та аналізу їхніх структурних та часових властивостей наступним критичним кроком є відстеження та ідентифікація цих об'єктів з плином часу. Алгоритми відстеження розроблені для визначення положення об'єктів у кількох кадрах, зберігаючи узгоджені ідентичності, незважаючи на зміни зовнішнього вигляду або часткові перекриття. Для цієї мети зазвичай використовуються фільтри Калмана та фільтри частинок; вони прогнозують майбутнє положення об'єктів на основі попередніх станів та оновлюють ці прогнози новими спостереженнями, що дозволяє плавно оцінювати траєкторію навіть у шумному середовищі. Методи відстеження траєкторії, такі як алгоритми відстеження кількох об'єктів (MOT), використовують методи асоціації даних для зв'язування виявлень у різних кадрах, формуючи безперервні шляхи, що представляють рух об'єкта з плином часу. Ці методи часто включають моделі руху та особливості зовнішнього вигляду для підвищення точності, особливо коли об'єкти тимчасово затьмарені або перекриваються. Обробка зіткнень, перетинів та часткових перекриттів створює додаткові труднощі; розширені алгоритми застосовують такі стратегії, як повторна ідентифікація та адаптивне зіставлення, щоб розрізняти об'єкти, які стикаються або перетинаються. Наприклад, коли два об'єкти стикаються, система може аналізувати їхні просторові та часові характеристики, щоб вирішити неоднозначності та призначити правильні ідентичності після взаємодії. Загалом, ефективне відстеження та ідентифікація є незамінними для сталого аналізу руху, дозволяючи системам розуміти складні взаємодії та поведінку в динамічних сценах, що є важливим для таких застосувань, як спостереження, автономна навігація та розпізнавання активності.

Підвищення ефективності обробки зображень та алгоритмів аналізу для розпізнавання руху є важливим для практичних застосувань, особливо тих, що потребують роботи в режимі реального часу, таких як автономні транспортні засоби, системи безпеки та інтерактивна робототехніка. Один з основних підходів включає оптимізацію обчислень, щоб забезпечити сумісність обробки з потоками відео в реальному часі. Такі методи, як алгоритмічне спрощення, паралельна

обробка, а апаратне прискорення — використання графічних процесорів (GPU) або програмованих польовими логічними матрицями (FPGA) — значно зменшує обчислювальне навантаження та затримку. Наприклад, паралельна реалізація операцій згортки або використання наближених алгоритмів для вилучення ознак може значно підвищити швидкість без шкоди для точності. Крім того, зменшення помилок за допомогою адаптивних фільтрів відіграє життєво важливу роль у покращенні результатів. Адаптивні фільтри, такі як фільтри Калмана або адаптивні шумозаглушення, динамічно коригують свої параметри на основі поточних даних, ефективно пригнічуючи залишковий шум та компенсуючи зміни навколишнього середовища, такі як динамічне освітлення або перекриття. Ця адаптивність забезпечує більш надійне виявлення та відстеження руху, мінімізуючи хибнопозитивні та негативні результати. Крім того, об'єднання кількох методів обробки в єдину схему, яку часто називають мультимодальним або гібридним підходом, використовує сильні сторони різних алгоритмів для подолання індивідуальних обмежень. Наприклад, інтеграція віднімання фону з оптичним потоком та відстеженням на основі ознак може покращити загальну надійність та точність, особливо у складних або захаращених сценах. Такі комплексні фреймворки не тільки підвищують продуктивність, але й дозволяють створювати масштабовані та адаптивні системи розпізнавання руху, придатні для різноманітних реальних сценаріїв, де швидкість і точність однаково важливі.

Процес розпізнавання руху за допомогою алгоритмів обробки та аналізу зображень включає серію взаємопов'язаних етапів, кожен з яких життєво важливий для загальної ефективності системи. Починаючи з попередньої обробки зображення, такі методи, як зменшення шуму, корекція яскравості та нормалізація, готують необроблені дані для змістовного аналізу. Потім виділення ключових ознак визначає характерні межі, текстури та дескриптори, які однозначно представляють рухомі об'єкти. Методи сегментації сцени, включаючи порогове визначення та кластеризацію, ізолюють ці об'єкти від статичного фону, створюючи основу для детального структурного аналізу. Вивчення геометричних та просторових властивостей об'єктів дає уявлення про їхні форми та

конфігурації, тоді як часове динамічне моделювання фіксує, як ці об'єкти рухаються та взаємодіють з часом. Алгоритми відстеження та ідентифікації забезпечують постійний моніторинг об'єктів у різних кадрах, навіть посеред перекриттів або взаємодій. Щоб зробити ці складні процеси життєздатними в реальних застосуваннях, підвищення ефективності алгоритмів за допомогою оптимізації, адаптивної фільтрації та інтегрованих схем обробки має першочергове значення. У сукупності ці передові методи обробки зображень утворюють основу надійних систем розпізнавання руху, що дозволяє використовувати застосування, починаючи від управління дорожнім рухом і закінчуючи автономною навігацією. З розвитком технологій, подальше вдосконалення цих алгоритмів ще більше підвищить їхню точність, швидкість та надійність, що зрештою розширить нашу здатність інтерпретувати динамічні візуальні середовища та взаємодіяти з ними.

1.4. Апаратні платформи та технічні засоби для побудови систем комп’ютерного зору

У швидко розвиваючій галузі комп'ютерного зору розробка ефективних та результативних систем значною мірою залежить від поєднання відповідних апаратних платформ та передових технічних інструментів. Ці компоненти працюють синергетично, дозволяючи машинам точно та в режимі реального часу інтерпретувати візуальні дані, що є важливим для застосувань, починаючи від автономних транспортних засобів і закінчуючи системами відеоспостереження та робототехнікою. З розвитком технологій розуміння різних апаратних платформ, таких як датчики, вбудовані обчислювальні пристрої та периферійні модулі, має вирішальне значення для проектування надійних систем, здатних виконувати складні візуальні завдання (табл. 1.4) [9].

Таблиця 1.4.

Апаратні платформи та технічні засоби для побудови систем комп’ютерного зору

Платформа / пристрій	Тип	Ключові технічні параметри	Сфера застосування
IP-камера відеоспостереження	Сенсорний пристрій	Роздільність до 4K, 25–60 FPS, WDR, нічний режим	Відеоаналітика, системи безпеки
USB / CSI камера	Сенсорний пристрій	Full HD, низька затримка, пряме підключення	Локальні системи комп’ютерного зору
Raspberry Pi 4	Вбудована платформа	CPU ARM Cortex-A72, до 8 GB RAM	Edge-обробка відео, IoT
NVIDIA Jetson Nano	Вбудована платформа з GPU	128 CUDA Cores, підтримка CUDA	Реальний час, відеоаналітика
Google Coral Dev Board	Edge-платформа з TPU	Edge TPU, низьке енергоспоживання	Інтелектуальні edge-системи
GPU-сервер	Серверна платформа	Потужні GPU, велика RAM	Централізована обробка відео
Edge-шлюз	Периферійний вузол	Мережева обробка, низька латентність	Розподілені системи
Мережеві інтерфейси	Комунікаційні засоби	Ethernet, Wi-Fi, PoE	Передача відеопотоків
Системи зберігання даних	Накопичувачі	SSD, HDD, NAS	Архівування відео

Датчики та камери утворюють основне обладнання для отримання відеоданих у системах комп'ютерного зору. Вибір типу камери суттєво впливає на якість та застосовність візуального вхідного сигналу. RGB-камери є найпоширенішими, вони захоплюють кольорові зображення, що підходять для широкого спектру застосувань, таких як розпізнавання об'єктів та розуміння сцени. Інфрачервоні камери, з іншого боку, цінні в умовах слабкого освітлення або вночі, виявляючи теплові сигнатури замість видимого світла, що дозволяє здійснювати нічне бачення та тепловий моніторинг. Глибинні камери, такі як ті, що використовуються в Microsoft Kinect або Intel RealSense, надають тривимірну інформацію, фіксуючи відстань до об'єктів від датчика, що є важливим для просторового розуміння та уникнення перешкод. Параметри захоплення, такі як роздільна здатність, частота кадрів та динамічний діапазон, додатково визначають ефективність відеозахоплення. Висока роздільна здатність забезпечує детальне

зображення, тоді як вища частота кадрів необхідна для аналізу руху в режимі реального часу, наприклад, в автономному водінні або навігації дронів. Динамічний діапазон впливає на здатність системи обробляти сцени з контрастними умовами освітлення, забезпечуючи чіткість як у тінях, так і в світлих тонах. Різні датчики пропонують унікальні переваги в системах розпізнавання руху; Наприклад, датчики глибини особливо добре відстежують рух у захаращених середовищах, покращуючи точність розпізнавання жестів або роботизованих маніпуляцій. Інтеграцію цих датчиків необхідно ретельно продумати, виходячи з конкретних вимог застосування, умов навколишнього середовища та обчислювальної потужності системи [10].

Вбудовані обчислювальні платформи служать основними процесорами в сучасних системах комп'ютерного зору, що дозволяє проводити аналіз у режимі реального часу та приймати рішення поблизу джерела даних. Мікрокомп'ютери, такі як Raspberry Pi, популярні завдяки своїй доступності, компактним розмірам та універсальності, що робить їх придатними для легких застосувань та створення прототипів. Більш потужні варіанти включають модулі Nvidia Jetson, які оснащені інтегрованими графічними процесорами, здатними прискорювати нейронний мережевий висновок, таким чином підтримуючи складні завдання, такі як виявлення об'єктів та сегментація сцени. Пристрої Google Coral містять спеціалізовані прискорювачі штучного інтелекту, призначені для забезпечення швидкої та енергоефективної обробки моделей глибокого навчання на периферії. Ці вбудовані системи часто включають процесори та графічні прискорювачі, оптимізовані для локальної обробки відео, що зменшує необхідність передачі великих обсягів даних на хмарні сервери, що може призвести до затримки та проблем із конфіденційністю. Принципи оптимізації обчислень на вбудованих системах включають ефективне управління пам'яттю, паралельну обробку та використання функцій апаратного прискорення для максимізації пропускної здатності при мінімізації енергоспоживання. Наприклад, використання ядер CUDA на Nvidia Jetson або Edge TPU на Google Coral дозволяє прискорювати логічний висновок, забезпечуючи продуктивність у режимі реального часу, що є

критично важливим для таких застосувань, як автономні транспортні засоби або промислова автоматизація. Вибір платформи залежить від обчислювальних вимог застосування, обмежень потужності та середовища розгортання, але всі вони спрямовані на забезпечення високої продуктивності в межах обмежених ресурсів вбудованого обладнання.

Периферійні обчислювальні модулі, включаючи периферійні пристрої та децентралізовані обчислювальні блоки, відіграють життєво важливу роль у підвищенні масштабованості та швидкості реагування систем комп'ютерного зору. Периферійні пристрої розгортаються близько до джерела даних, таких як камери або датчики, що дозволяє попередньо обробляти відеопотоки перед передачею на центральні сервери. Ця децентралізація зменшує навантаження на пропускну здатність та зменшує затримку, пов'язану з передачею великих обсягів необроблених даних по мережах. Підтримка нейронних модулів на периферії, таких як невеликі чіпи штучного інтелекту або прискорювачі на основі FPGA, сприяє локальному логічному висновку та прийняттю рішень, що особливо корисно в сценаріях, де негайні реакції є критично важливими. Наприклад, в автономних транспортних засобах локальна обробка даних датчиків дозволяє швидко виявляти та уникати перешкод, забезпечуючи безпеку та ефективність. Крім того, децентралізована обробка підвищує конфіденційність, обмежуючи доступ до необроблених візуальних даних, оскільки передаються лише відповідні аналітичні дані або сповіщення. Переваги такої архітектури включають покращену стійкість системи, масштабованість та енергоефективність. Зменшуючи залежність від централізованих центрів обробки даних, периферійні модулі дозволяють проводити аналітику в режимі реального часу в середовищах з обмеженим підключенням або суворими вимогами до затримки, тим самим підтримуючи розгортання інтелектуальних систем у різноманітних та складних умовах [11].

Ефективні комунікаційні інтерфейси є важливими для передачі відеоданих від датчиків і камер до блоків обробки в системах комп'ютерного зору. Вибір інтерфейсів підключення камери суттєво впливає на швидкість передачі даних,

стабільність та сумісність. До поширених інтерфейсів належать USB, який пропонує широку сумісність і простоту використання, але може мати обмеження пропускної здатності при високій роздільній здатності; CSI (Camera Serial Interface), спеціально розроблений для високошвидкісної передачі даних з камер, поширений у вбудованих системах завдяки своїй ефективності та низькій затримці; а Ethernet-з'єднання, які часто використовуються для мережевих камер, підтримують передачу на великі відстані та інтеграцію у більші мережі спостереження або промислові мережі. Для забезпечення передачі відеоданих у реальному часі використовуються різні протоколи, такі як RTSP (Real-Time Streaming Protocol), RTP (Real-time Transport Protocol) та власні рішення, що забезпечують синхронізацію та управління пакетами, що є критично важливим для програм з низькою затримкою. Підтримка стабільності каналу та достатньої пропускної здатності має першочергове значення; коливання або вузькі місця можуть призвести до втрати кадрів або погіршення якості зображення, що погіршує продуктивність системи. Такі методи, як пріоритезація якості обслуговування (QoS), управління мережевим трафіком та виділення виділеної пропускної здатності, впроваджуються для захисту цілісності даних та забезпечення послідовного зв'язку, особливо в середовищах з кількома потоками даних або обмеженою мережевою інфраструктурою. Ці заходи разом сприяють надійному та ефективному конвеєру даних, що дозволяє безперешкодно інтегрувати візуальні датчики з апаратним забезпеченням обробки. Системи зберігання та буферизації даних є основоположними для управління об'ємами відеоінформації, що генеруються датчиками комп'ютерного зору. Для тимчасового або постійного зберігання відеопотоків використовуються різні носії, включаючи твердотільні накопичувачі (SSD), оперативну пам'ять високої ємності та мережеві сховища (NAS). Вибір залежить від таких факторів, як вимоги до швидкості передачі даних, ємність сховища та архітектура системи. Буферизація потоків є критично важливою технікою, яка використовується для обробки безперервних відеоданих, запобігаючи втраті даних під час затримок обробки або коливань мережі. Адекватні розміри буферів та розподіл пам'яті необхідні для забезпечення

високої пропускної здатності даних, особливо в сценаріях, що включають відео високої роздільної здатності або передачу даних з кількох камер. Ефективне управління пам'яттю забезпечує безперебійну чергу та обробку вхідних даних, мінімізуючи затримку та запобігаючи перевантаженню системи. В умовах високого навантаження для оптимізації доступу до даних та їх пошуку використовуються такі методи, як кільцеві буфери, багатоканальні кеші та алгоритми пріоритезації. Ці підходи допомагають підтримувати продуктивність у режимі реального часу, балансуючи операції введення/виведення, зменшуючи час доступу та забезпечуючи беззатримкову обробку критичних кадрів. Правильне зберігання та буферизація даних відіграють важливу роль у підтримці стабільності, швидкості реагування та надійності систем комп'ютерного зору, особливо у вимогливих програмах, таких як автономна навігація або спостереження в реальному часі.

Живлення апаратних компонентів є ключовим аспектом, що впливає на робочу стабільність та довговічність систем комп'ютерного зору. Камери, процесори та периферійні модулі споживають різну кількість енергії, причому високопродуктивні компоненти зазвичай потребують більше енергії. Ефективне управління живленням є важливим для забезпечення надійної роботи, особливо в автономних системах, таких як дрони або мобільні роботи, де зовнішні джерела живлення обмежені. Оптимізація режимів живлення включає використання станів сну, динамічного масштабування напруги та частоти (DVFS) та адаптивних політик живлення для зменшення споживання в періоди низької активності без шкоди для продуктивності. Для безперервної роботи використовуються апаратні функції, такі як режими низького енергоспоживання, регулювання потужності та ефективні регулятори живлення, щоб подовжити термін служби батареї та запобігти перегріву. Крім того, проектування схем живлення, які можуть обробляти пікові навантаження та перехідні вимоги, забезпечує стійкість системи до інтенсивних завдань обробки. Балансування енергоефективності з вимогами до обробки є вирішальним для розгортання стійких та автономних рішень

комп'ютерного зору в різних середовищах, від вбудованих пристроїв до великомасштабних мереж спостереження [12].

Інтеграція апаратної інфраструктури є критично важливим аспектом розробки комплексних та ефективних систем комп'ютерного зору, особливо тих, що призначені для складних застосувань, таких як розпізнавання руху. Ця інтеграція передбачає безперешкодне поєднання камер, вбудованих обчислювальних модулів, периферійних пристроїв та мережевої інфраструктури для формування цілісної екосистеми, здатної обробляти дані в режимі реального часу та приймати рішення. Досягнення сумісності між різними апаратними компонентами вимагає дотримання стандартизованих інтерфейсів та протоколів, що гарантує ефективну взаємодію кожного елемента без конфліктів або вузьких місць. Наприклад, інтеграція RGB-камер високої роздільної здатності або камер глибини з такими процесорами, як Nvidia Jetson або Google Coral, вимагає відповідних інтерфейсів передачі даних, таких як CSI або USB, та сумісних протоколів живлення та зв'язку. Побудова масштабованої апаратної архітектури передбачає проектування модульних систем. де додаткові датчики або блоки обробки можна легко додавати за потреби зростати, не порушуючи існуючі робочі процеси. Така масштабованість життєво важлива для застосувань розпізнавання руху, де можна використовувати кілька камер для охоплення більших площ або підвищення точності. Модульні конструкції також полегшують обслуговування та модернізацію, дозволяючи системам розвиватися разом із технологічним прогресом. Крім того, інтеграція надійної мережевої інфраструктури, такої як високошвидкісний Ethernet або 5G-з'єднання, дозволяє здійснювати дистанційне керування та обмін даними, що є важливим для розподілених систем, таких як мережі спостереження або автономні парки транспортних засобів. Загалом, ефективна інтеграція апаратної інфраструктури гарантує гармонійне функціонування всіх компонентів, підтримуючи надійні, масштабовані та високопродуктивні рішення комп'ютерного зору.

Побудова ефективних систем комп'ютерного зору залежить від стратегічного вибору та інтеграції різних апаратних платформ і технічних

інструментів. Датчики та камери, включаючи RGB, інфрачервоні та глибинні, служать основними блоками збору даних, а їхні параметри адаптовані до потреб конкретних застосувань. Вбудовані обчислювальні платформи, такі як Raspberry Pi, Nvidia Jetson та Google Coral, пропонують локалізовані можливості обробки, оптимізуючи продуктивність завдяки апаратному прискоренню та ефективному управлінню ресурсами. Периферійні модулі, особливо граничні пристрої, децентралізують обробку, зменшуючи затримку та вимоги до пропускної здатності, одночасно підвищуючи конфіденційність та швидкість реагування. Комунікаційні інтерфейси, такі як USB, CSI та Ethernet, у поєднанні з надійними протоколами передачі, сприяють надійному потоку даних у режимі реального часу, який підтримується передовими системами зберігання та буферизації, що керують великими обсягами згенерованих відеоданих. Стратегії управління живленням забезпечують стійкість та стабільність усієї системи, особливо в мобільних або автономних умовах. Нарешті, інтеграція цих компонентів у цілісну, масштабовану інфраструктуру має першорядне значення, що дозволяє складним програмам, таким як розпізнавання руху, ефективно працювати та адаптуватися до мінливих технологічних вимог. Разом ці апаратні платформи та інструменти утворюють основу сучасних систем комп'ютерного зору, прокладаючи шлях для інновацій у численних галузях та обіцяючи майбутнє розумніших, більш автономних машин.

2. ТЕХНОЛОГІЇ ТА ПІДХОДИ ДО ПОБУДОВИ СИСТЕМИ РОЗПІЗНАВАННЯ РУХОМИХ ОБ’ЄКТІВ

2.1. Підходи до побудови систем комп’ютерного зору та їх порівняльний аналіз

Комп'ютерний зір став ключовою галуззю в галузі штучного інтелекту, метою якої є надання машинам можливості інтерпретувати та розуміти візуальну інформацію про навколишній світ. Протягом багатьох років було розроблено численні підходи до створення ефективних систем комп'ютерного зору, кожен з яких має свої особливості, сильні сторони та обмеження. Ці підходи можна загалом розділити на класичні алгоритмічні методи, методи на основі ознак та моделі, що ґрунтуються на статистичних принципах. Розуміння нюансів цих методів та їх порівняння дає цінне уявлення про їхню застосовність у різних реальних сценаріях, від простого виявлення об'єктів до аналізу складних сцен [13].

Класичні алгоритмічні підходи представляють одні з найдавніших методів комп'ютерного зору, що значною мірою спираються на фундаментальні методи обробки зображень, такі як фільтрація, встановлення порогів та морфологічні операції. Ці методи включають застосування фільтрів, таких як фільтри Гауса або Собеля, для покращення ознак або зменшення шуму, а потім встановлення порогів для сегментації об'єктів від фону на основі значень інтенсивності пікселів. Наприклад, прості алгоритми виявлення країв, такі як детектор країв Канні, визначають межі в межах зображень, виявляючи градієнти інтенсивності. Аналітичні моделі також включають геометричні та математичні принципи, такі як перетворення Хафа для виявлення форми або зіставлення шаблонів для розпізнавання об'єктів. Хоча ці методи є обчислювально ефективними та простими у реалізації, вони часто мають труднощі зі складними або захараченими сценами, де умови освітлення, тіні та перекриття суттєво впливають на ознаки зображення. Отже, класичні алгоритми, як правило, дають збої в динамічних

середовищах або сценаріях, що вимагають детального розпізнавання, що спонукає до еволюції до більш складних підходів (рис. 2.1).

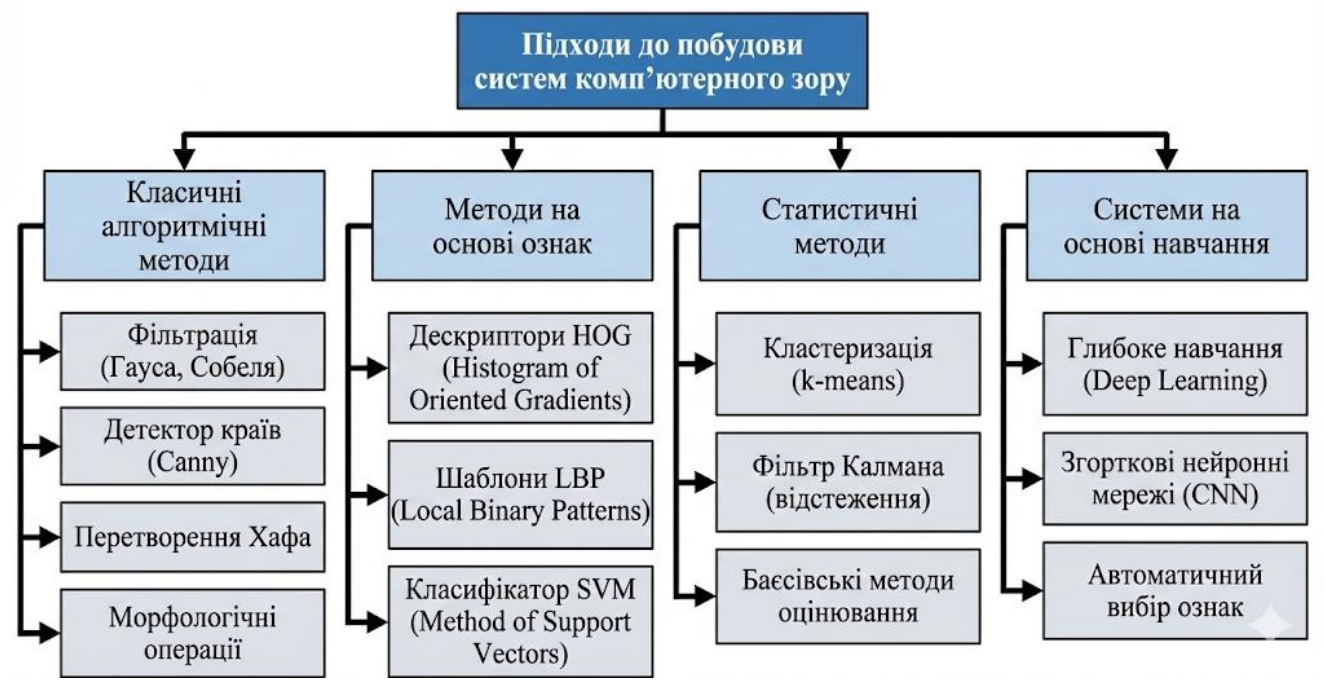


Рис. 2.1. Класифікація підходів до побудови систем комп'ютерного зору

Підходи на основі ознак здобули популярність, зосереджуючись на вилученні значущих локальних дескрипторів із зображень, які служать надійними представленнями для завдань класифікації. Такі методи, як гістограма орієнтованих градієнтів (HOG) та локальні бінарні шаблони (LBP), є прикладами цієї стратегії. HOG фіксує орієнтацію та величини градієнтів у локалізованих областях, ефективно представляючи форму та текстуру об'єкта, тоді як LBP кодує локальні бінарні шаблони для опису текстури та властивостей поверхні. Ці дескриптори потім об'єднуються у вектори ознак, які можна вводити в класифікатори, такі як методи опорних векторів (SVM) або нейронні мережі, для ідентифікації об'єктів або сцен. Основна перевага методів на основі ознак полягає в їхній стійкості до змін навколишнього середовища, таких як зміни освітлення, масштабу та точки зору. Наприклад, у системах розпізнавання облич ознаки LBP забезпечують стійкість до різниць освітлення, що дозволяє надійно

ідентифікувати їх у різних умовах. Тим не менш, ці підходи вимагають ретельного вибору ознак і можуть бути обчислювально ресурсоємними при роботі із зображеннями високої роздільної здатності або великими наборами даних. Їхній успіх залежить від дискримінаційної здатності вибраних ознак та ефективності використовуваних алгоритмів класифікації.

Моделі, засновані на статистичних методах, включають ймовірнісні міркування, кластеризацію та аналіз розподілу для більш гнучкої інтерпретації візуальних даних (табл. 2.1). Алгоритми кластеризації, такі як k-середні або ієрархічна кластеризація, групують подібні піксельні шаблони або вектори ознак, допомагаючи в таких завданнях, як сегментація зображень та розмежування об'єктів. Баєсівські методи оцінювання, з іншого боку, використовують попередні знання та функції правдоподібності для здійснення ймовірнісних висновків про області зображення або стани об'єктів. Наприклад, у програмах відстеження баєсівські фільтри, такі як фільтр Калмана або фільтр частинок, оцінюють положення та швидкість рухомих об'єктів з часом, враховуючи невизначеності та шум вимірювань. Ці статистичні моделі особливо ефективні в динамічних середовищах, де дані неповні або зашумлені, дозволяючи системам адаптивно вдосконалювати своє розуміння сцени. Їх застосування поширюється на складні завдання, такі як відеоспостереження, автономна навігація та аналіз медичних зображень, де ймовірнісні міркування підвищують надійність та точність. Однак вони часто вимагають значних обчислювальних ресурсів та ретельного налаштування параметрів для досягнення оптимальної продуктивності, особливо у багатовимірних просторах даних [14].

Системи, засновані на навчанні, революціонізували комп'ютерний зір, використовуючи великі набори даних та передові методи машинного навчання для автоматичного вивчення шаблонів та ознак з візуальних даних. На відміну від традиційних методів, що залежать від ручно створених ознак, ці системи використовують архітектури глибокого навчання, такі як згорткові нейронні мережі (ЗНМ), які можуть автоматично визначати найбільш релевантні ознаки для заданого завдання. Цей процес включає навчання моделей на великих наборах

даних з маркуванням, що дозволяє їм ефективно узагальнювати та адаптуватися до нових зображень і сценаріїв (табл. 2.1) [15].

Таблиця 2.1.

Деталізоване порівняння підходів до побудови систем комп'ютерного зору за технічними параметрами

Підхід	Тип обробки	Середня затримка	Обчислювальні ресурси	Точність розпізнавання	Придатність для реального часу
Класичні алгоритмічні методи	Правилова, детермінована	Низька	Низькі (CPU)	Середня	Висока
Підходи на основі ознак	Аналітична з етапом класифікації	Середня	Середні (CPU)	Середня–висока	Середня
Статистичні методи	Імовірнісна	Середня	Середні	Середня	Середня
Методи на основі навчання	Навчальна	Висока	Високі (GPU)	Висока	Обмежена
Гібридні підходи	Комбінована	Середня	Середні–високі	Висока	Висока

Істотною перевагою підходів, заснованих на навчанні, є їхня здатність до автоматичного вибору ознак; моделі за своєю суттю визначають, які ознаки роблять найбільший внесок у класифікацію або виявлення, усуваючи необхідність ручної розробки ознак. Більше того, ці системи демонструють чудову адаптивність до різних умов зйомки, таких як зміни освітлення, перспективи або складності фону, якщо вони представлені в навчальних даних. Ця адаптивність робить їх дуже придатними для реальних застосувань, таких як автономне водіння, розпізнавання облич і медична діагностика. Однак вони часто є обчислювально ресурсоємними, вимагаючи потужного обладнання та значних ресурсів. час навчання, що може бути обмежувальним фактором у середовищах з обмеженими ресурсами. Тим не менш, їхня здатність постійно покращувати

точність та адаптуватися до різноманітних умов зробила системи, засновані на навчанні, наріжним каменем сучасного комп'ютерного зору.

Гібридні підходи до комп'ютерного зору спрямовані на поєднання сильних сторін класичних алгоритмів та систем, заснованих на навчанні, для підвищення загальної продуктивності та одночасного зменшення обмежень. Інтегруючи традиційні методи обробки зображень з моделями машинного навчання, гібридні рішення прагнуть покращити точність завдяки взаємодоповнюючим сильним сторонам — класичні методи забезпечують швидку початкову сегментацію або вилучення ознак, які потім можна уточнити за допомогою вивчених ознак або класифікаторів. Наприклад, гібридна система може використовувати фільтрацію та виявлення країв для визначення областей-кандидатів, а потім класифікатори на основі CNN для точного розпізнавання. Така інтеграція часто призводить до кращої точності у складних сценах, ніж будь-який з підходів окремо, особливо в середовищах зі складним освітленням або захараченим фоном. Крім того, гібридні методи можуть допомогти зменшити обчислювальну складність, обмежуючи обсяг моделей глибокого навчання певними областями або ознаками, визначеними за допомогою класичних методів, таким чином уникаючи обробки повного зображення. Ці підходи також сприяють поступовому вдосконаленню, дозволяючи системам використовувати існуючі алгоритми та набори даних, поступово включаючи компоненти навчання. Як наслідок, гібридні рішення набувають дедалі більшої переваги в таких застосуваннях, як аналіз відео в реальному часі та робототехніка, де баланс точності та обчислювальної ефективності є критично важливим [16].

Комплексний порівняльний аналіз різних підходів до побудови систем комп'ютерного зору включає оцінку їхньої продуктивності на основі ключових критеріїв, таких як точність, стабільність, споживання ресурсів та масштабованість. Класичні алгоритми, хоча й обчислювально недорогі та прості, як правило, не мають стійкості та часто дають збій у складних реальних сценаріях через їхню залежність від спрощених припущень та ручних функцій. Методи на основі функцій пропонують покращену стійкість та точність, особливо за різних

умов навколишнього середовища, але можуть бути обчислювально вимогливими та чутливими до якості вибору функцій. Статистичні моделі чудово працюють у динамічних середовищах, забезпечуючи ймовірнісне мислення та адаптивність, проте часто вимагають значних обчислювальних ресурсів та тонкого налаштування. Системи, засновані на навчанні, зокрема моделі глибокого навчання, зазвичай досягають найвищої точності та адаптивності, але вимагають значних апаратних ресурсів, великих наборів даних та значного часу на навчання. Їх масштабованість залежить від наявності даних та обчислювальної інфраструктури, що створює проблеми з розгортанням в умовах обмежених ресурсів. Загалом, вибір підходу залежить від вимог конкретного застосування, балансуючи компроміси між точністю, стабільністю, ресурсоємністю та масштабованістю для досягнення оптимальної продуктивності в реальних умовах.

Вибір оптимального підходу для практичної системи комп'ютерного зору вимагає ретельного врахування вимог конкретного застосування та експлуатаційних обмежень. Наприклад, застосування, що вимагають обробки в реальному часі з мінімальною затримкою, такі як автономні транспортні засоби або роботизована навігація, повинні пріоритезувати методи, які балансують між швидкістю та точністю. У таких випадках класичні алгоритми або гібридні системи часто слугують відповідними варіантами, оскільки вони можуть забезпечити швидку початкову обробку з меншими обчислювальними витратами, які потім можна вдосконалити за допомогою компонентів, що базуються на навчанні, за необхідності. І навпаки, сценарії, де висока точність розпізнавання та надійність є першочерговими, такі як медична діагностика або відеоспостереження, можуть отримати користь від моделей глибокого навчання, незважаючи на їх вищі вимоги до ресурсів. Компроміс між точністю та обчислювальними ресурсами є критичним фактором; середовища з обмеженими ресурсами, такі як вбудовані системи або периферійні пристрої, можуть сприяти підходам, заснованим на функціях, або спрощеним гібридним підходам для підтримки операційної ефективності. Для систем розпізнавання рухомих об'єктів рекомендований підхід передбачає гібридну стратегію: використання класичних

методів фільтрації та виявлення руху для швидкої початкової локалізації, доповнених класифікаторами на основі навчання для точної ідентифікації та відстеження. Таке поєднання забезпечує збалансовану систему, здатну забезпечити високу точність без надмірної затримки, що робить її добре придатною для динамічних реальних застосувань, де важливі як швидкість, так і надійність.

Розробка ефективних систем комп'ютерного зору охоплює різноманітний спектр підходів, кожен з яких має унікальні переваги та обмеження. Класичні алгоритмічні методи заклали фундамент за допомогою простих, обчислювально ефективних методів, але часто їм бракувало стійкості у складних сценах. Підходи на основі ознак просунули цю галузь, витягуючи значущі локальні дескриптори, що покращували стійкість до змін навколишнього середовища, тоді як статистичні моделі запровадили ймовірнісні міркування для кращої обробки невизначеностей та динамічних умов. Поява систем на основі навчання революціонізувала цю галузь, дозволивши автоматичне вилучення ознак та адаптацію до нових сценаріїв, значно підвищивши точність та універсальність. Гібридні підходи далі інтегрують ці методології, щоб скористатися їхніми взаємодоповнюючими сильними сторонами, що призводить до більш надійних та ефективних рішень. Зрештою, вибір підходу залежить від потреб конкретного застосування, балансуючи такі фактори, як швидкість, точність, доступність ресурсів та масштабованість. У міру розвитку технологій майбутні розробки, ймовірно, будуть зосереджені на оптимізації цих компромісів, сприяючи створенню більш інтелектуальних, адаптивних та ресурсоефективних систем комп'ютерного зору, здатних надійно працювати в різноманітних реальних середовищах [17].

2.2. Методи оптимізації обробки відеоданих у реальному часі

У швидко розвиваючій галузі цифрового мультимедіа обробка відео в реальному часі стала невід'ємною частиною численних застосувань, включаючи прямі трансляції, відеоконференції, автономні транспортні засоби та доповнену

реальність. Потреба у високоякісному відеовиході з низькою затримкою вимагає впровадження ефективних стратегій оптимізації для підвищення швидкості та ефективності обробки. Досягнення оптимальної продуктивності передбачає використання комбінації методів на різних рівнях проектування системи. У цьому розділі досліджуються три критичні методи оптимізації обробки відео в реальному часі: покращення на рівні алгоритмів, стратегії управління пам'яттю та методи паралельної обробки. Ці підходи, за умови розумного застосування, можуть значно зменшити затримки обробки, покращити пропускну здатність та підтримувати високу візуальну точність, тим самим розширюючи можливості систем відео в реальному часі (рис. 2.2) [18].

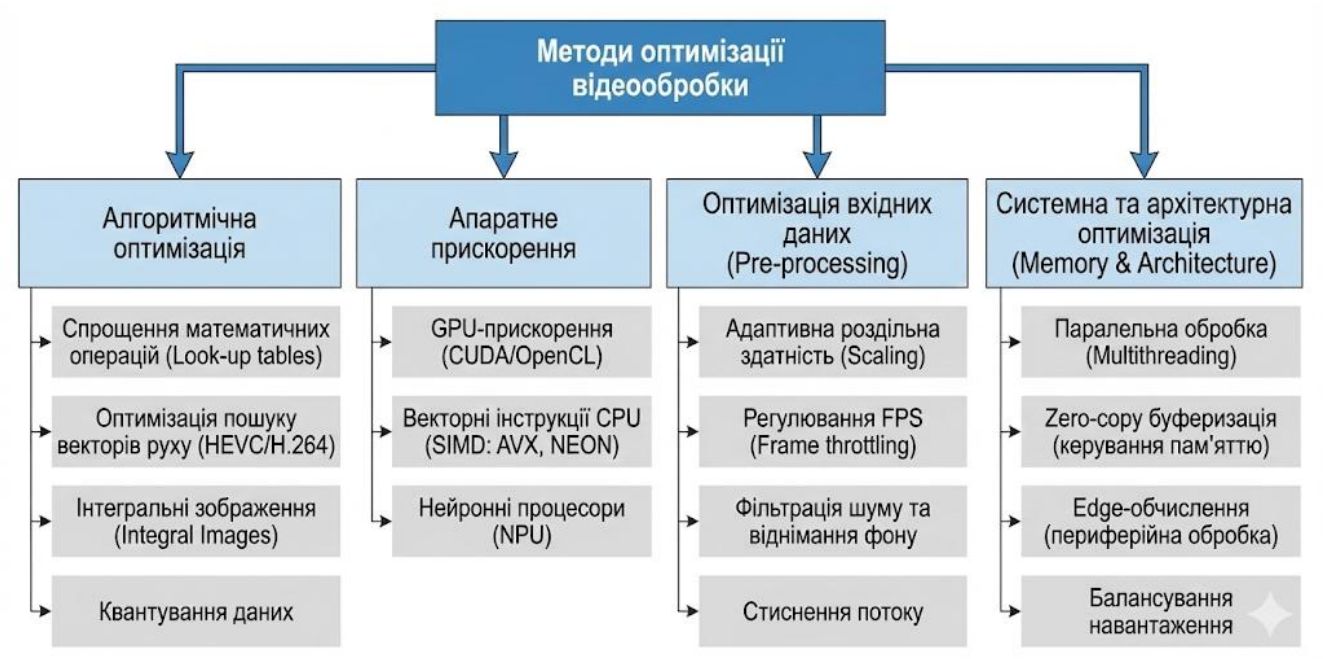


Рис. 2.2. Класифікація методів оптимізації обробки відеоданих у реальному часі

Одним із фундаментальних підходів до покращення обробки відео в реальному часі є оптимізація алгоритмів, що лежать в основі основних обчислювальних завдань. Це передбачає зменшення кількості операцій, що виконуються на кожному кадрі, тим самим зменшуючи обчислювальне навантаження. Наприклад, в алгоритмах стиснення відео, таких як H.264 або HEVC, використання оцінки руху можна оптимізувати, обмеживши область

пошуку векторів руху, що зменшує кількість обчислень без суттєвого впливу на якість. Аналогічно, методи апроксимації, такі як квантування та спрощені перетворення, можуть бути використані для пришвидшення обробки, зберігаючи при цьому прийнятну візуальну якість. Наприклад, використання грубіших рівнів квантування може значно зменшити розмір даних та обчислювальні зусилля. Крім того, спрощення обчислювальних процедур, таких як апроксимація складних математичних функцій за допомогою таблиць пошуку або поліноміальних апроксимацій, може призвести до швидшого виконання. Практичною ілюстрацією є використання інтегральних зображень у виявленні об'єктів у реальному часі, що спрощує обчислення сум по областях зображення, тим самим прискорюючи обробку. Ці оптимізації на рівні алгоритмів є критично важливими, оскільки вони безпосередньо впливають на обчислювальну складність, дозволяючи системам ефективно обробляти відеопотоки високої роздільної здатності без шкоди для необхідної якості. Керування пам'яттю відіграє життєво важливу роль в ефективності обробки відео в реальному часі. Мінімізація обсягів буферизації є важливою, оскільки великі буфери створюють затримку та споживають значні системні ресурси. Наприклад, протоколи потокової передачі та конвеєри обробки розроблені для обробки даних невеликими, керованими фрагментами, а не великими, безперервними буферами, тим самим зменшуючи обсяг пам'яті та затримку. Ефективне використання кеш-пам'яті не менш важливе; шляхом організації даних для максимізації потраплянь до кешу, наприклад, через просторову та часову локальність, можна значно покращити продуктивність системи. Наприклад, обробка кадрів таким чином, що доступ до даних, що зберігаються поруч у пам'яті, мінімізує промахи кешу та прискорює обчислення. Зменшення кількості копій даних між системними блоками є ще однією критичною стратегією, оскільки кожна копія створює затримку та споживає пропускну здатність. Такі методи, як керування буфером з нульовою копією, де дані передаються безпосередньо між етапами обробки без дублювання, є прикладом цього підходу. У високопродуктивних відеосистемах ці стратегії оптимізації пам'яті разом сприяють швидшому часу обробки, зниженню

енергоспоживання та більш стабільній реакції системи, особливо під час обробки потоків високої або надвисокої чіткості.

Паралельна обробка є потужним методом задоволення обчислювальних потреб програм відео в реальному часі. Розділяючи складні завдання на незалежні підзадачі, системи можуть використовувати кілька процесорів для прискорення загальної пропускної здатності. Наприклад, у типовому конвеєрі декодування відео такі завдання, як компенсація руху, зворотне перетворення та реконструкція пікселів, можуть бути паралелізовані на кількох ядрах. Використання багатопотоковості дозволяє кожному ядру одночасно обробляти різні частини кадру або різні етапи обробки, що значно скорочує загальний час обробки. Сучасні графічні процесори (GPU) особливо підходять для такого паралелізму, оскільки вони містять тисячі ядер, призначених для одночасного виконання подібних завдань (табл. 2.2) [19].

Наприклад, при виявленні об'єктів у реальному часі різні області кадру можуть оброблятися незалежно для ідентифікації ознак, що дозволяє швидше розпізнавати. Крім того, поділ кадрів на плитки або блоки та одночасна обробка цих сегментів використовує паралелізм даних, що життєво важливо для підтримки низької затримки. Такий підхід не тільки прискорює обробку, але й підвищує масштабованість, дозволяючи системам обробляти вищу роздільну здатність і частоту кадрів без шкоди для продуктивності. Загалом, паралельна обробка використовує весь потенціал сучасних апаратних архітектур, забезпечуючи продуктивність у реальному часі навіть у складних умовах. Окрім алгоритмічних, пам'яттєвих та паралельних стратегій обробки, використання апаратного прискорення стало ключовим підходом до оптимізації обробки відео в реальному часі. Апаратне прискорення передбачає використання спеціалізованих процесорів, призначених для виконання певних обчислювальних завдань ефективніше, ніж універсальні центральні процесори. Одним з найяскравіших прикладів є графічний процесор (GPU), який за своєю суттю здатний виконувати масово паралельні обчислення завдяки тисячам ядер. Графічні процесори чудово справляються з високопродуктивними вимогами до завдань декодування,

рендерингу та обробки відео, дозволяючи одночасне виконання кількох операцій, які в іншому випадку були б вузьким місцем програмних рішень.

Таблиця 2.2.

Деталізована оцінка методів оптимізації обробки відеоданих у реальному часі

Метод оптимізації	Рівень реалізації	Середня затримка, мс	FPS	Навантаження CPU	Навантаження GPU	Практичний ефект
Алгоритмічна оптимізація	Алгоритмічний	10–20	30–60	Низьке	Відсутнє	Зменшення часу обробки кадру
Паралельна обробка	Програмно-апаратний	8–15	40–90	Середнє	Низьке	Підвищення пропускної здатності
Апаратне прискорення (GPU)	Апаратний	3–8	60–120	Низьке	Середнє–високе	Мінімальна затримка
Зменшення роздільності кадру	Вхідні дані	5–10	60–100	Низьке	Низьке	Зменшення навантаження
Адаптивне зниження FPS	Вхідні дані	5–15	20–60	Низьке	Низьке	Баланс якості та швидкодії
Буферизація та кешування	Пам’ять	10–18	30–60	Середнє	Відсутнє	Зменшення доступів до пам’яті
Edge-обробка відео	Архітектурний	5–12	30–60	Середнє	Низьке	Зменшення мережевої затримки
Розподілена обробка	Архітектурний	6–14	60–120	Середнє	Середнє	Масштабування системи

Крім того, інструкції процесора SIMD (Single Instruction, Multiple Data) відіграють важливу роль в оптимізації завдань обробки відео на рівні інструкцій. SIMD дозволяє виконувати одну операцію одночасно над кількома точками даних, що особливо вигідно для попіксельних операцій, таких як фільтрація, перетворення колірного простору та виявлення країв. Наприклад, набори

інструкцій, такі як AVX від Intel або NEON від ARM, забезпечують швидку векторизовану обробку, суттєво скорочуючи час обчислення. Зовсім недавно поява нейронних процесорів (NPU) відкрила нові шляхи для прискорення завдяки апаратному забезпеченню, адаптованому для нейронного мережевого виводу. NPU оптимізовані для завдань машинного навчання, що дозволяє реалізовувати в режимі реального часу передові алгоритми, такі як виявлення об'єктів, розуміння сцени та покращення зображення, безпосередньо на спеціалізованому обладнанні. Інтеграція цих апаратних прискорювачів у конвеєри обробки відео може значно підвищити пропускну здатність, зменшити затримку та забезпечити обробку з високою роздільною здатністю в середовищах з обмеженими ресурсами, роблячи програми реального часу більш доцільними та ефективними.

Ще один ефективний метод оптимізації обробки відео в реальному часі, особливо в системах з обмеженими обчислювальними ресурсами, включає налаштування роздільної здатності та частоти кадрів відеопотоку. Адаптивне масштабування зображення динамічно змінює роздільну здатність на основі обчислювальної потужності або важливості контенту, тим самим зменшуючи обчислювальне навантаження без постійного погіршення якості зображення. Наприклад, у моменти низької активності або коли висока деталізація не потрібна, системи можуть тимчасово знизити роздільну здатність, щоб підтримувати продуктивність у реальному часі. Аналогічно, зменшення кількості кадрів на секунду (FPS) може значно зменшити вимоги до обробки, зберігаючи при цьому важливу інформацію у відеопотоці. Цей підхід, який часто називають дроселюванням FPS, передбачає вибіркове видалення кадрів або використання таких методів, як інтерполяція кадрів, для підтримки плавного сприйняття руху. Найголовніше, що ці коригування можна вносити адаптивно, щоб забезпечити збереження критичної інформації, такої як швидкорухомі об'єкти або важливі зміни сцени, а менш релевантні дані відкидати. Ця стратегія особливо важлива для систем з низькими ресурсами, таких як вбудовані пристрої або мобільні платформи, де обчислювальна потужність, споживання енергії та пропускна здатність обмежені. Завдяки інтелектуальному управлінню роздільною здатністю

та частотою кадрів системи можуть ефективно працювати без шкоди для основних функцій або взаємодії з користувачем, що забезпечує ефективну обробку в реальному часі в широкому спектрі застосувань.

Методи стиснення та попередньої обробки є важливою частиною оптимізації обробки відео в реальному часі шляхом зменшення обсягів даних та фільтрації нерелевантної інформації перед аналізом. Стиснення відеопотоку за допомогою таких кодеків, як H.264, HEVC або VP9, значно зменшує обсяг даних, які необхідно передати та обробити, що, у свою чергу, зменшує вимоги до затримки та пропускної здатності. Ефективні алгоритми стиснення балансують якість та розмір, забезпечуючи потокову передачу високої роздільної здатності з керованими обчислювальними витратами. Поряд зі стисненням, етапи попередньої обробки, такі як фільтрація шуму, мають вирішальне значення для пришвидшення подальшого аналізу. Методи зменшення шуму, такі як медіанна фільтрація або двостороння фільтрація, усувають нерелевантні деталі, які в іншому випадку могли б призвести до неправильної класифікації або збільшити складність обробки. Крім того, попередня фільтрація даних для видалення нерелевантної або надлишкової інформації, такої як статичні фонові або несуттєві об'єкти, допомагає зосередити обчислювальні ресурси на значущому контенті. Наприклад, алгоритми віднімання фону можуть ізолювати об'єкти переднього плану для цілеспрямованої обробки, зменшуючи обсяг даних, що потребують інтенсивного аналізу.

Ці методи попередньої обробки не тільки пришвидшують робочі процеси обробки, але й підвищують точність та надійність завдань вищого рівня, таких як виявлення об'єктів, відстеження та розуміння сцени. Загалом, поєднання стиснення з інтелектуальною попередньою обробкою створює оптимізований конвеєр, який максимізує ефективність, зберігаючи при цьому цілісність критичної візуальної інформації в системах відео в реальному часі. Окрім окремих компонентів та методів обробки, оптимізація загальної архітектури системи має вирішальне значення для досягнення ефективної обробки відео в режимі реального часу. Розподілена обробка між периферійними пристроями є прикладом

цього підходу шляхом децентралізації обчислювальних завдань, що зменшує навантаження на центральні сервери та зменшує затримки передачі даних. Наприклад, в автономних транспортних засобах дані датчиків можуть оброблятися локально на вбудованих системах або периферійних вузлах, що дозволяє швидко приймати рішення без затримки, пов'язаної з хмарною обробкою. Така локалізація аналітики не тільки прискорює час відгуку, але й економить пропускну здатність, передаючи лише відповідну оброблену інформацію, а не цілі потоки сировини.

Крім того, впровадження механізмів балансування навантаження між кількома обчислювальними модулями гарантує, що жоден окремий процесорний блок не стане вузьким місцем, тим самим підтримуючи стабільність системи та ефективність при різних навантаженнях. Такі методи, як динамічний розподіл завдань та моніторинг ресурсів, дозволяють системам адаптивно розподіляти завдання обробки на основі показників продуктивності в режимі реального часу. Ці архітектурні стратегії є життєво важливими для масштабованих, стійких та низькозатримувальних систем обробки відео, особливо в складних середовищах, де задіяно кілька джерел даних та потоки високої роздільної здатності. Оптимізуючи систему на цьому макрорівні, розробники можуть забезпечити безперебійне функціонування всього конвеєра, забезпечуючи високоякісний відеовихід у режимі реального часу з мінімальною затримкою та максимальним використанням ресурсів [20].

Оптимізація обробки відео в реальному часі вимагає багатогранного підходу, який охоплює алгоритмічне вдосконалення, ефективне управління пам'яттю, паралельну обробку, апаратне прискорення, адаптивне управління роздільною здатністю, методи попередньої обробки та проектування архітектури на системному рівні. Покращення на рівні алгоритмів, такі як скорочення обчислювальних операцій та використання методів апроксимації, закладають основу для швидшої обробки. Оптимізація пам'яті, включаючи мінімізацію буферизації та максимізацію ефективності кешу, ще більше оптимізує пропускну здатність. Паралельна обробка використовує потужність сучасного обладнання,

такого як багатоядерні центральні та графічні процесори, значно пришвидшуючи виконання завдань. Апаратні прискорювачі, такі як нейронні процесори та інструкції SIMD, забезпечують спеціалізоване прискорення для вимогливих операцій, тоді як адаптивне налаштування роздільної здатності та частоти кадрів адаптує навантаження на обробку до можливостей системи. Методи попередньої обробки та стиснення зменшують обсяги даних та фільтрують нерелевантну інформацію, оптимізуючи конвеєр. Нарешті, впровадження розподіленої системної архітектури з локальною аналітикою та балансуванням навантаження забезпечує масштабованість та низьку затримку в різних середовищах. Разом ці стратегії утворюють комплексну основу для підвищення ефективності обробки відео в реальному часі, що дозволяє програмам забезпечувати високоякісний досвід з низькою затримкою навіть за обмежених ресурсів. Оскільки технології продовжують розвиватися, інтеграція цих методів буде важливою для задоволення зростаючих потреб цифрових мультимедійних систем у світі, що стає дедалі більш пов'язаним.

Таблиця 2.3 узагальнює результати застосування методів оптимізації обробки відеоданих, розглянутих у таблиці 2.2, та дозволяє наочно оцінити їх практичний вплив на ключові параметри функціонування системи розпізнавання рухомих об’єктів. На відміну від таблиці 2.2, у якій подано детальну характеристику окремих оптимізаційних підходів за часовими, обчислювальними та ресурсними показниками, таблиця 2.3 відображає інтегральний ефект від їх комплексного використання в межах однієї системи.

Наведене порівняння параметрів до та після оптимізації дає змогу обґрунтувати доцільність запропонованих технічних рішень, підтвердити забезпечення роботи системи в режимі реального часу та показати підвищення точності й масштабованості системи. Отримані результати створюють підґрунтя для переходу до розділу 3, у якому здійснюється проєктування архітектури, реалізація програмного модуля та експериментальна оцінка ефективності розробленої системи.

Таблиця 2.3.

Порівняльна оцінка параметрів системи розпізнавання рухомих об’єктів до та після оптимізації

Параметр	До оптимізації	Після оптимізації	Ефект оптимізації
Середня затримка обробки, мс	25–40	5–12	Зменшення затримки у 3–4 рази
Продуктивність, FPS	15–25	45–90	Збільшення пропускної здатності
Навантаження CPU	Високе	Середнє	Зменшення процесорного навантаження
Навантаження GPU	Відсутнє	Середнє	Перенесення обчислень на GPU
Стійкість до шумів	Середня	Висока	Покращення якості виявлення
Кількість хибних спрацювань	Висока	Низька	Підвищення точності розпізнавання
Масштабованість системи	Обмежена	Висока	Можливість обробки кількох потоків
Робота в реальному часі	Нестабільна	Стабільна	Виконання вимог реального часу

2.3. Вимоги до точності, масштабованості та надійності систем розпізнавання руху

У швидко розвиваючій галузі систем розпізнавання руху забезпечення їхньої точності, масштабованості та надійності має вирішальне значення для застосувань, починаючи від систем відеоспостереження і закінчуючи автономною навігацією. Ці системи покликані ідентифікувати та відстежувати об'єкти або осіб у динамічних середовищах, часто за різноманітних та непередбачуваних умов. Досягнення високих стандартів точності розпізнавання, обробка зростаючих обсягів даних та підтримка стабільної продуктивності в різних операційних сценаріях є важливими для їхньої ефективності та широкого впровадження. У

цьому розділі досліджуються фундаментальні вимоги до точності, масштабованості та надійності систем розпізнавання руху, окреслюються критичні фактори, що впливають на їхню продуктивність, та технологічні міркування, необхідні для відповідності цим стандартам (табл. 2.4) [21].

Таблиця 2.4.

Показники оцінювання ефективності системи розпізнавання рухомих об’єктів

Показник	Метод оцінювання	Отримане значення	Практична інтерпретація
Точність розпізнавання	Precision / Recall	Висока	Коректне виявлення рухомих об’єктів
Кількість хибних спрацювань	False Positive Rate	Низька	Зменшення помилкових детекцій
Середня затримка обробки	Час обробки кадру, мс	5–12 мс	Робота в режимі реального часу
Продуктивність системи	Кількість кадрів за секунду	45–90 FPS	Стабільна обробка відеопотоку
Стійкість до шумів	Тестування в умовах завад	Висока	Коректна робота при шумних даних
Надійність роботи	Безперервне тестування	Стабільна	Відсутність збоїв у тривалому режимі
Масштабованість	Збільшення кількості потоків	Підтримується	Можливість розширення системи

Основною вимогою до систем розпізнавання руху є високий рівень точності, який залежить від мінімізації хибнопозитивних результатів – неправильної ідентифікації об’єктів або рухів, яких немає. Хибнопозитивні результати можуть призвести до непотрібних сповіщень, марнування ресурсів та зниження довіри до надійності системи, особливо в контексті безпеки, де хибні тривоги можуть викликати паніку або самовдоволення. Щоб пом’якшити це, системи повинні включати складні алгоритми, здатні розрізняти відповідний рух від фонового шуму та тимчасових артефактів. Крім того, життєво важливою є стабільність

результатів за різних умов зйомки; такі фактори, як зміни погоди, освітлення або кути камери, можуть впливати на ефективність виявлення. Наприклад, камера безпеки, що працює вночі, повинна надійно виявляти вторгнення, незважаючи на умови низької освітленості, що вимагає надійних методів обробки зображень. Крім того, здатність системи точно розпізнавати різні типи об’єктів – від людей до транспортних засобів або тварин – має вирішальне значення для універсальності. Це включає навчання моделей на різноманітних наборах даних, щоб забезпечити їхню здатність добре узагальнюватися для різних категорій об’єктів, тим самим підтримуючи високу точність у реальних сценаріях.

Кілька екологічних та технічних факторів суттєво впливають на точність систем розпізнавання руху. Якість відеосигналу є фундаментальною; Відеозаписи з високою роздільною здатністю та мінімальним шумом сприяють точнішому виявленню та відстеженню. І навпаки, шум — випадкові коливання значень пікселів — можуть затьмарювати відповідні особливості, що призводить до неправильної класифікації або пропуску виявлень. Умови освітлення є ще одним критичним фактором; коливання освітленості, такі як тіні або відблиски, можуть спотворювати зовнішній вигляд об'єктів та поставити під сумнів надійність алгоритмів розпізнавання. Рух камери, навмисний чи через фактори навколишнього середовища, такі як вітер, може спричинити розмиття руху та зміщення кадру, що ще більше ускладнює точне виявлення. Складні сцени з об'єктами, що перекриваються, або захаращеним фоном створюють додаткові проблеми, оскільки вони можуть заплутати алгоритми, які намагаються виділити певні моделі руху. Наприклад, сцена з переповненою вулицею з пішоходами, що перекриваються, та транспортними засобами, що рухаються, вимагає передових методів сегментації для точного розрізнення різних рухомих об'єктів. Усунення цих факторів передбачає використання високоякісних датчиків, адаптивних методів обробки зображень та надійних алгоритмів, здатних обробляти різноманітні умови навколишнього середовища для підтримки високої точності розпізнавання.

Оскільки системи розпізнавання руху розгортаються у великих або складніших середовищах, масштабованість стає критичним питанням. Система повинна бути здатною обробляти кілька відеопотоків одночасно без погіршення продуктивності — необхідність у великомасштабних мережах спостереження або автономних системах з кількома камерами. Це вимагає ефективних обчислювальних архітектур, часто з використанням паралельної обробки та хмарних рішень, для обробки високої пропускної здатності даних. Гнучкість у додаванні нових об'єктів та зон спостереження є не менш важливою, оскільки середовища розвиваються та виникають нові потреби в моніторингу. Модульні системні конструкції та адаптивні алгоритми дозволяють легко інтегрувати нові категорії об'єктів або зони без значного перепрограмування. Наприклад, система безпеки повинна мати можливість динамічно включати нові типи транспортних засобів або пішохідні зони. Балансування навантаження є важливим при збільшенні кількості камер; розподіл завдань обробки між кількома серверами або вузлами гарантує, що жоден компонент не стане вузьким місцем. Такі методи, як динамічний розподіл ресурсів та розподілені обчислення, допомагають підтримувати продуктивність у режимі реального часу та стабільність системи, гарантуючи, що система залишається надійною, коли зростають експлуатаційні вимоги.

Для ефективної оцінки та забезпечення масштабованості систем розпізнавання руху необхідно враховувати кілька ключових показників. Однією з основних метрик є пропускна здатність базової обчислювальної інфраструктури, яка відображає здатність системи обробляти великі обсяги відеоданих протягом заданого періоду часу. Висока пропускна здатність гарантує, що кілька потоків можуть бути проаналізовані одночасно, не спричиняючи затримок або відставань, що життєво важливо для програм реального часу, таких як спостереження або автономна навігація. Не менш важливою є затримка — затримка між захопленням відео та виводом розпізнавання, — особливо у великомасштабних системах, де навіть незначні затримки можуть поставити під загрозу своєчасне прийняття рішень. Низька затримка є важливою для підтримки ситуаційної обізнаності та

оперативності реагування в динамічних середовищах. Ефективність використання пам'яті також відіграє вирішальну роль; оскільки системи розширюються для розміщення більшої кількості камер або зон, вони повинні оптимізувати використання пам'яті, щоб запобігти вузьким місцям. Ефективне управління пам'яттю дозволяє безперешкодно інтегрувати додаткові джерела даних без перевантаження апаратних ресурсів, тим самим підтримуючи продуктивність і стабільність системи. Разом ці показники — пропускна здатність, затримка, та ефективність пам'яті — служать орієнтирами для оцінки та керівництва масштабованим розгортанням систем розпізнавання руху у все складніших середовищах [22].

Надійність систем розпізнавання руху є основоположною для їхньої сталої ефективності, особливо під час розгортання в критично важливих програмах, таких як забезпечення безпеки або автономна навігація транспортних засобів. Одним з важливих аспектів надійності є безперервність роботи в режимі реального часу; системи повинні працювати стабільно протягом тривалого часу без перерв. Це передбачає стійке апаратне забезпечення, надійне програмне забезпечення та ефективну обробку помилок для запобігання простоям або зниженню продуктивності. Стійкість до втрати кадрів та збоїв обладнання також має вирішальне значення, оскільки втрачені кадри або несправності апаратного забезпечення можуть призвести до неповних даних та порушення точності розпізнавання. Системи повинні бути розроблені для ефективної роботи навіть за відсутності деяких даних, використовуючи алгоритми, здатні інтерполювати або компенсувати прогалини. Крім того, ці системи повинні коректно обробляти неповні або пошкоджені дані, забезпечуючи безперервну роботу без помилкових тривог або пропуску життєво важливих подій. Ця надійність підвищує впевненість користувачів і гарантує, що системи постійно підтримують свої захисні або операційні ролі, незалежно від непередбачених технічних проблем або екологічних проблем.

Досягнення високого рівня надійності систем розпізнавання руху передбачає численні стратегічні заходи. Резервування ключових модулів – це

фундаментальний підхід, коли критично важливі компоненти, такі як процесори, пристрої зберігання даних та мережеві з’єднання, дублюються, щоб запобігти виникненню єдиних точок відмови. Якщо один модуль виходить з ладу, його резервна копія безперешкодно бере на себе роботу, мінімізуючи час простою. Використання відмовостійких механізмів ще більше підвищує стійкість системи; до них належать протоколи виявлення помилок, стратегії відновлення після збою та адаптивні алгоритми, здатні працювати в неоптимальних умовах. Процедури автоматичного відновлення також є життєво важливими, дозволяючи системі швидко виявляти проблеми та відновлювати нормальне функціонування без втручання людини. Наприклад, автоматизовані процедури перезапуску або динамічна реконфігурація можуть перерозподілити ресурси або перенаправити потоки даних для обходу несправностей. У сукупності ці заходи забезпечують безперервну та надійну роботу систем розпізнавання руху навіть за умови апаратних збоїв, програмних помилок або несприятливих умов навколишнього середовища, таким чином зберігаючи їхню цілісність та надійність з часом.

Щоб системи розпізнавання руху відповідали суворим вимогам реальних застосувань, важлива комплексна оцінка загальної якості системи. Це передбачає інтеграцію ключових показників точності, надійності та масштабованості в єдину систему оцінки. Поєднання цих показників дозволяє розробникам та операторам отримати цілісне уявлення про продуктивність системи, визначаючи сильні сторони та області для покращення. Наприклад, ефективна оцінка може включати вимірювання точності виявлення та рівня хибнопозитивних результатів (точність), часу безвідмовної роботи системи та відмовостійкості (надійність), а також пропускної здатності та затримки обробки при підвищеному навантаженні (масштабованість). Така багатовимірна оцінка сприяє прийняттю обґрунтованих рішень щодо розгортання, обслуговування та модернізації системи. Крім того, вибір оптимальної конфігурації системи значною мірою залежить від конкретних умов експлуатації та вимог застосування. Наприклад, система безпеки в міській зоні з високим рівнем руху може надавати пріоритет масштабованості та реагуванню в режимі реального часу, тоді як автономна транспортна система може

наголошувати на точності та надійності в різних умовах навколишнього середовища. Розробка індивідуальних показників продуктивності та використання суворих протоколів тестування гарантують, що система розпізнавання відповідає передбачуваному операційному середовищу, тим самим максимізуючи її ефективність та термін служби (рис. 2.3).

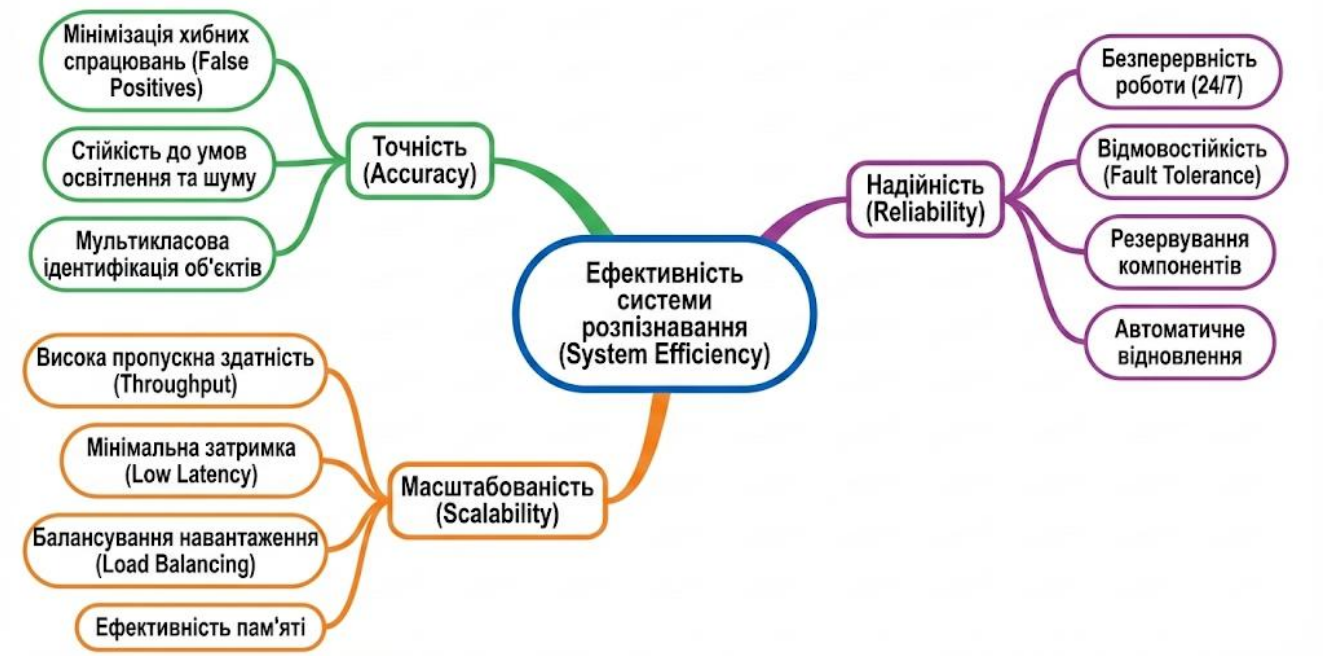


Рис. 2.3. Ключові критерії ефективності системи розпізнавання рухомих об'єктів

Ефективність сучасних систем розпізнавання руху визначається не лише досконалістю алгоритмів, а й здатністю дотримуватися суворих вимог за трьома фундаментальними та взаємопов'язаними вимірами: точність, масштабованість та надійність [23].

Досягнення високої точності розпізнавання є першочерговим завданням, яке передбачає мінімізацію кількості хибнопозитивних результатів, оскільки помилкові детекції можуть знизити довіру до системи та призвести до неефективного використання ресурсів. Критично важливою є підтримка стабільності роботи в різних умовах зйомки, адже такі фактори, як зміни освітлення (тіні, відблиски), погодні умови або шум відеосигналу, можуть суттєво

спотворювати візуальні ознаки об'єктів. Окрім цього, система повинна забезпечувати точну ідентифікацію та класифікацію різноманітних типів об'єктів (наприклад, розрізняти людей і транспортні засоби) навіть у складних сценах із захараченим фоном або перекриттям об'єктів.

Для підтримки масштабних та динамічних середовищ ці системи повинні бути проєктовані як масштабовані рішення, здатні обробляти зростаючі потоки даних без втрати продуктивності. Вони мають характеризуватися високою пропускною здатністю для одночасного аналізу кількох відеопотоків та низькою затримкою, що є критичним для своєчасного реагування в режимі реального часу. Гнучкі можливості розширення та впровадження механізмів динамічного балансування навантаження сприяють побудові надійної інфраструктури, де нові сенсори або зони моніторингу можуть бути інтегровані без створення вузьких місць в обчислювальних вузлах [24].

Не менш важливим аспектом є забезпечення надійності, що гарантує стабільність системи протягом тривалого часу. Це досягається завдяки безперервній роботі в режимі 24/7 та стійкості до втрати даних, коли алгоритми здатні компенсувати пропущені кадри або фрагментарну інформацію. Архітектура системи повинна включати відмовостійкі конструктивні особливості, такі як апаратне резервування критичних модулів та процедури автоматичного відновлення після збоїв, що мінімізує час простою.

Комплексна оцінка, яка інтегрує ці показники, пропонує цінний підхід до оптимізації загальної продуктивності системи. Такий аналіз дозволяє розробникам знаходити компроміси та створювати індивідуальні конфігурації, що найкраще відповідають конкретним операційним контекстам та пріоритетам замовника. Зрештою, подальший розвиток технології розпізнавання руху вимагає збалансованого фокусу на цих взаємопов'язаних факторах, гарантуючи, що системи залишатимуться точними, масштабованими та надійними, щоб відповідати вимогам різноманітних реальних застосувань, а також сприяти довірі та ефективності їх практичного розгортання.

3. РОЗРОБКА ТА ВПРОВАДЖЕННЯ СИСТЕМИ РОЗПІЗНАВАННЯ РУХОМИХ ОБ’ЄКТІВ

3.1. Проєктування архітектури та вибір алгоритмічного забезпечення

У швидко розвиваючій галузі систем розпізнавання, проектування архітектури системи та вибір допоміжних алгоритмів є критичними факторами, що впливають на продуктивність, масштабованість та надійність. Ефективна розробка архітектури починається з всебічного розуміння функціональних вимог до системи, які слугують основою для подальших проектних рішень. Ці вимоги охоплюють конкретні завдання, які повинна виконувати система, такі як виявлення об'єктів у режимі реального часу або розпізнавання облич, та встановлюють стандарти точності та швидкості реагування. Поряд з функціональними потребами, обмеження, пов'язані із затримкою та продуктивністю, є життєво важливими, особливо в таких застосуваннях, як спостереження або автономні транспортні засоби, де затримки можуть поставити під загрозу безпеку або операційну ефективність. Крім того, необхідно враховувати масштабованість та розширюваність, щоб система могла обробляти зростаючі обсяги даних або впроваджувати майбутні технологічні досягнення без значного перепроектування. Вирішення цих фундаментальних аспектів гарантує, що архітектура відповідає як безпосереднім операційним цілям, так і довгостроковим стратегічним цілям, створюючи стійку основу для алгоритмічної підтримки та розгортання системи.

Формування вимог до архітектури системи – це ретельний процес, який включає визначення точних функціональних специфікацій та встановлення ключових обмежень продуктивності. Функціональні вимоги визначають, що повинна робити система, наприклад, виявляти певні об'єкти в різних умовах освітлення або розпізнавати людей під різними кутами та в різних середовищах. Наприклад, система розпізнавання безпеки може потребувати ідентифікувати порушників у режимі реального часу з мінімальним рівнем хибних спрацьовувань,

що вимагає використання точних та швидких алгоритмів. Встановлення обмежень на затримку передбачає визначення прийнятних затримок для обробки даних, що має вирішальне значення в чутливих до часу додатках, таких як автономне водіння, де навіть мілісекунди мають значення. Обмеження продуктивності включають обчислювальну ефективність, точність та стійкість за різних умов навколишнього середовища. Щоб забезпечити майбутнє зростання, архітектура також повинна підтримувати масштабованість – здатність обробляти збільшені навантаження даних – та розширюваність – легкість інтеграції нових алгоритмів або модулів. Ці початкові вимоги слугують керівними принципами, що впливають на вибір апаратного, програмного та алгоритмічних стратегій, тим самим формуючи загальний дизайн системи [25].

Вибір відповідної архітектурної моделі є ключовим рішенням, яке впливає на ефективність, гнучкість та ремонтопридатність системи. Централізовані архітектури, де обробка зосереджена на основному сервері або в центрі обробки даних, пропонують переваги з точки зору легшого управління, послідовної обробки даних та спрощених оновлень. Наприклад, система розпізнавання облич, яка централізовано обробляє дані, може використовувати потужні серверні ресурси для підвищення точності. І навпаки, розподілені архітектури, які розподіляють обробку між кількома вузлами або периферійними пристроями, часто є кращими в сценаріях, що вимагають низької затримки та високої доступності, таких як аналітика відео в реальному часі в розумних містах. У таких випадках периферійні процесорні блоки обробляють початкову фільтрацію даних або вилучення ознак, зменшуючи навантаження на центральну систему та мінімізуючи затримки. Визначення ролей периферійної та серверної обробки включає оцінку таких факторів, як пропускна здатність мережі, обчислювальна потужність кінцевих пристроїв та експлуатаційні вимоги. Логічна структура потім виникає з цих міркувань, визначаючи, як компоненти взаємодіють — чи то через прямий зв'язок, передачу повідомлень, чи спільні сховища даних — таким чином створюючи узгоджену структуру, яка збалансовує продуктивність, надійність та масштабованість.

Проектування модульної системної архітектури є важливим для створення гнучких, зручних у підтримці та оновлюваних рішень для розпізнавання. Це передбачає ізоляцію основних функціональних компонентів, таких як модулі відеозахоплення, обробки та аналізу, кожен з яких функціонує як незалежний блок із чітко визначеними обов'язками. Наприклад, модуль відеозахоплення обробляє збір даних з камер, тоді як модулі обробки виконують такі завдання, як покращення зображення або вилучення ознак. Потім модулі аналізу інтерпретують ці оброблені дані для розпізнавання облич, об'єктів або поведінки. Визначення чітких інтерфейсів між цими модулями забезпечує безперебійну комунікацію та передачу даних, дозволяючи кожному компоненту розроблятися, тестуватися та оновлюватися незалежно. Ця модульність сприяє взаємозамінності компонентів; наприклад, заміна алгоритму розпізнавання облич на більш просунутий може бути досягнута без капітального ремонту всієї системи. Більше того, модульна архітектура підтримує паралельні зусилля з розробки та спрощує усунення несправностей, що зрештою призводить до надійної системи розпізнавання, здатної розвиватися разом із технологічними інноваціями та зміною експлуатаційних вимог [26].

На рис. 3.1 наведено узагальнену структурно-функціональну схему розробленої системи, яка візуалізує архітектуру рішення та логіку потоків даних між ключовими програмними модулями. Схема демонструє послідовний конвеєр обробки відеоінформації, що складається з трьох основних рівнів: апаратного забезпечення, програмного ядра обробки та інтерфейсу виводу результатів.

1. Рівень захоплення даних (Video Capture):

Вхідною точкою системи є модуль відеозахоплення, який забезпечує отримання потокового відео з камери (IP-камери або веб-камери) у реальному часі. Цей блок відповідає за стабільність зчитування кадрів та їх буферизацію для подальшої обробки.

2. Ядро обробки та аналізу (Core Processing Pipeline):

Це центральний компонент системи, який реалізує вибрані алгоритмічні рішення:

– Блок попередньої обробки (Preprocessing): Виконує фільтрацію шумів (наприклад, за допомогою фільтра Гауса) та нормалізацію зображення для усунення артефактів сенсора та змін освітлення.

– Блок виявлення руху (Motion Detection): Реалізує алгоритм віднімання фону (Background Subtraction) для виокремлення рухомих областей на основі аналізу різниці кадрів або адаптивного моделювання фону.

– Блок сегментації та морфології (Segmentation): Відповідає за очищення маски руху від дрібних шумів, відновлення цілісності об'єктів за допомогою морфологічних операцій та формування контурів (ROI).

– Блок відстеження (Tracking): Забезпечує прив'язку ідентифікаторів до виявлених об'єктів та прогнозування їх траєкторії (наприклад, з використанням фільтра Калмана), що дозволяє системі супроводжувати об'єкт навіть при короткочасному зникненні або перекритті.

3. Рівень виводу та взаємодії (Output Interface):

Завершальний етап, на якому результати аналізу (обмежувальні рамки, траєкторії руху, сповіщення) візуалізуються для користувача або передаються до зовнішніх систем зберігання даних (рис. 3.1).

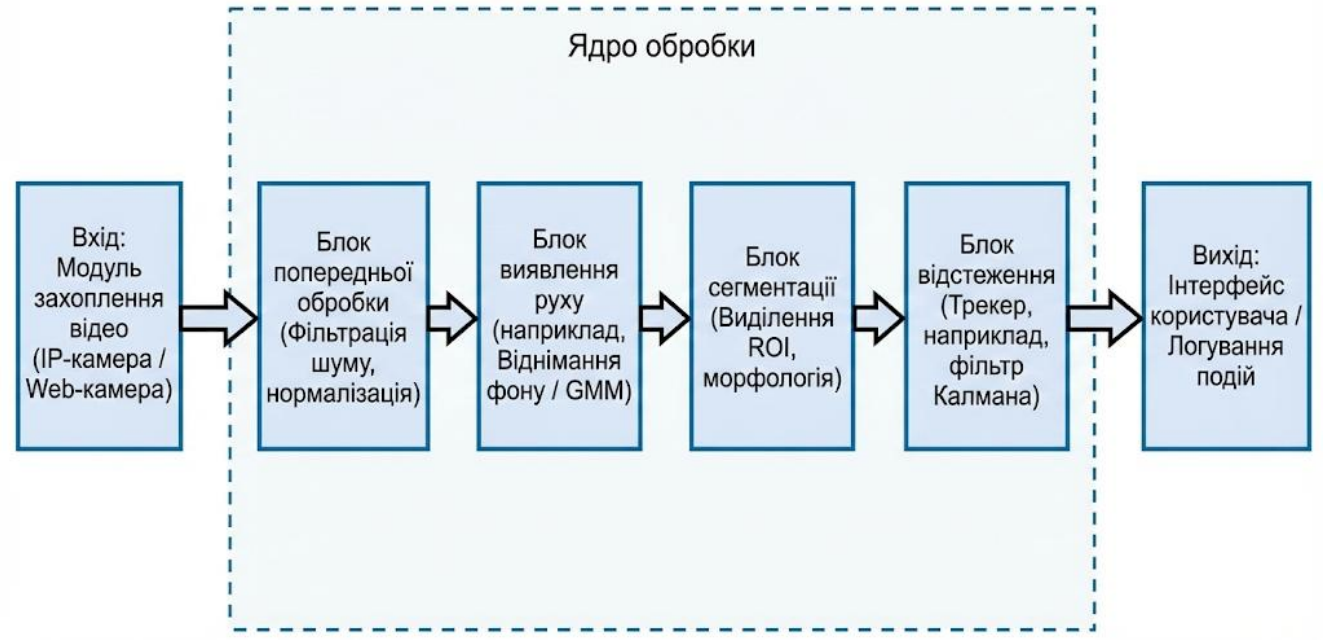


Рис. 3.1. Структурно-функціональна схема розробленої системи

Така модульна архітектура забезпечує гнучкість системи, дозволяючи незалежно оптимізувати або замінювати окремі алгоритми без порушення загальної логіки роботи, а також сприяє досягненню необхідної швидкодії для роботи в режимі реального часу.

Вибір алгоритмів для виявлення рухомих об'єктів відіграє вирішальну роль у загальній ефективності систем розпізнавання, особливо в динамічних середовищах. Існують різні методи виявлення руху, кожен з яких має свої сильні та обмежені сторони. Традиційні підходи, такі як диференціація кадрів, порівнюють послідовні кадри для виявлення змін, пропонуючи простоту, але часто страждаючи від чутливості до шуму та раптових змін освітлення. Як альтернатива, методи віднімання фону моделюють статичні елементи сцени для ізоляції рухомих об'єктів, забезпечуючи кращу стійкість у контрольованих умовах, але вимагаючи ретельного налаштування для адаптації до змін навколишнього середовища. Більш досконаліші алгоритми використовують оптичний потік для оцінки попіксельного руху, фіксуючи детальні шаблони руху за рахунок вищих обчислювальних вимог. Вибираючи основний метод виявлення руху, необхідно враховувати операційний контекст; наприклад, віднімання фону може бути кращим у відносно стабільних умовах у приміщенні, тоді як оптичний потік може бути необхідним для складних сцен на відкритому повітрі зі швидкими рухами. Обґрунтування вибору основного алгоритму залежить від балансування точності виявлення, обчислювальної ефективності та адаптивності. Щоб забезпечити надійну роботу в реальних сценаріях, обраний алгоритм має бути адаптивним — здатним обробляти зміни освітлення, фонові перешкоди та рух камери — шляхом включення етапів попередньої обробки, таких як адаптивне порогове регулювання або динамічне моделювання фону. Така індивідуальна адаптація гарантує, що алгоритм залишається ефективним за різноманітних та непередбачуваних умов, типових для реальних відеопотоків.

Після виявлення руху система повинна аналізувати та відстежувати об'єкти, щоб підтримувати безперервну ідентифікацію з часом. Вибір відповідних алгоритмів сегментації та відстеження включає оцінку методів, які можуть надійно ізолювати об'єкти від фону та слідувати їхнім траєкторіям. Такі методи, як сегментація на основі контурів, ефективні для визначення меж об'єктів, але можуть мати проблеми з перекриттями або перекриттям об'єктів; навпаки, моделі сегментації на основі глибокого навчання, такі як Mask R-CNN, пропонують високу точність, але вимагають значних обчислювальних ресурсів. Для відстеження такі алгоритми, як фільтри Калмана, забезпечують ефективну оцінку руху об'єкта, особливо в лінійних та передбачуваних сценаріях руху, тоді як складніші методи, такі як фільтри частинок або моделі глибокого відстеження, краще підходять для нелінійних або нестабільних моделей руху. Оцінка траєкторій руху передбачає інтеграцію цих алгоритмів для прогнозування майбутніх позицій на основі поточних та минулих спостережень, що дозволяє системі коректно обробляти тимчасові перекриття або пропущені виявлення. Інтеграція алгоритмів сегментації та відстеження в цілісний ланцюг обробки вимагає ретельної синхронізації, яка забезпечує безперешкодне надходження виходів виявлення до модулів відстеження, та надійної обробки помилок для відновлення після збоїв відстеження. Оптимальний ланцюг обробки поєднує високоточну сегментацію з адаптивними алгоритмами відстеження, утворюючи безперервний, надійний конвеєр, який підтримує точну ідентифікацію об'єктів у різних кадрах, що є важливим для таких застосувань, як спостереження, моніторинг дорожнього руху або автономна навігація [27].

Координація алгоритмічних стратегій з апаратною підтримкою є життєво важливою для створення системи, здатної працювати в режимі реального часу та масштабованості. Оцінка того, чи відповідають вибрані алгоритми доступним обчислювальним ресурсам, передбачає аналіз їхньої складності обробки та вимог до пам'яті. Для систем з обмеженими апаратними можливостями, таких як вбудовані пристрої або датчики країв, легкі алгоритми, такі як спрощене віднімання фону або оптичний потік зі зниженою роздільною здатністю, є

кращими для підтримки продуктивності без надмірного споживання ресурсів. І навпаки, системи, оснащені високопродуктивними графічними процесорами або хмарною інфраструктурою, можуть використовувати більш просунуті, ресурсоємні алгоритми, такі як глибокі нейронні мережі, для підвищення точності та надійності. Методи оптимізації, включаючи уточнення алгоритмів, паралельну обробку та апаратне прискорення, використовуються для адаптації алгоритмів до конкретної платформи. Наприклад, розгортання нейронних мереж з оптимізованими механізмами логічного висновку або моделями обрізання для зменшення обчислювального навантаження гарантує, що система підтримує швидкість реагування в режимі реального часу. Забезпечення роботи в режимі реального часу вимагає ретельного тестування та налаштування як програмних, так і апаратних компонентів, з постійним моніторингом для виявлення вузьких місць або проблем із затримкою. Завдяки такому скоординованому підходу система розпізнавання може надійно обробляти відеопотоки з необхідною швидкістю, підтримуючи своєчасне прийняття рішень та операційну ефективність у практичних застосуваннях.

Кульмінацією процесу проектування є формування комплексної архітектурної та алгоритмічної конфігурації, яка інтегрує всі попередні міркування в цілісний план впровадження. Цей етап починається з побудови узагальненої структурної діаграми системи, яка візуально представляє взаємозв'язки та потоки даних між різними компонентами, такими як модулі відеозапису, блоки обробки та алгоритми аналізу. Така діаграма служить основоположним орієнтиром, забезпечуючи чіткість та узгодженість в архітектурі системи. Одночасно, конкретні алгоритми, вибрані для виявлення руху, сегментації та відстеження, фіксуються разом з їхніми оптимізованими параметрами, які були точно налаштовані на попередніх етапах оцінки для балансу точності та обчислювальної ефективності. Ця детальна специфікація алгоритмів та параметрів забезпечує узгодженість та полегшує усунення несправностей під час розробки (табл. 3.1). Крім того, архітектура має бути підготовлена до наступного етапу впровадження програмного забезпечення

шляхом визначення детальних специфікацій інтерфейсу, форматів даних та протоколів інтеграції. Ця підготовка включає перетворення концептуального проекту на технічну документацію та прототипи, що створює умови для безперебійного кодування, тестування та розгортання.

Таблиця 3.1.

Обґрунтування вибору алгоритмічного забезпечення системи

Підсистема / Етап	Обраний метод / Алгоритм	Обґрунтування вибору (Практична доцільність)
Виявлення руху	Віднімання фону (напр. MOG2/KNN)	Забезпечує баланс між точністю та швидкодією в умовах статичної камери. Стійкий до незначних змін освітлення.
Фільтрація шуму	Фільтр Гауса (Gaussian Blur)	Ефективно прибирає сенсорний шум ("зерно") при мінімальному навантаженні на CPU.
Відстеження (Tracking)	Фільтр Калмана (або SORT)	Дозволяє прогнозувати траєкторію об'єкта навіть при короткочасному перекритті (оклюзії).
Сегментація	Морфологічні операції (Dilation/Erosion)	Усуває розриви в масці об'єкта та прибирає дрібні артефакти, формуючи чіткий контур.
Архітектура	Модульна (Distributed/Edge)	Дозволяє легку заміну модулів та масштабування системи без переписання коду.

Завдяки ретельному доопрацюванню структурної та алгоритмічної конфігурації розробники можуть гарантувати, що система є надійною, масштабованою та готовою до реального використання.світове застосування, що зрештою призведе до надійної системи розпізнавання, здатної що відповідає всім експлуатаційним вимогам.

Підсумовуючи, ефективне проектування систем розпізнавання залежить від системного підходу, який починається з чіткого визначення функціональних вимог та встановлення обмежень продуктивності, забезпечуючи відповідність архітектури операційним потребам. Вибір відповідної архітектурної моделі — централізованої чи розподіленої — відіграє життєво важливу роль у балансуванні ефективності, масштабованості та надійності, тоді як принципи модульного проектування сприяють ремонтопридатності та майбутнім оновленням. Ретельний вибір та адаптація алгоритмів для виявлення руху, аналізу об'єктів та відстеження є важливими для досягнення високої точності та стійкості в різних реальних умовах. Не менш важливою є координація цих алгоритмів з відповідною апаратною підтримкою, що забезпечує продуктивність у режимі реального часу завдяки оптимізації та управлінню ресурсами. Останній крок включає об'єднання цих елементів в єдину структурну схему, фіксацію алгоритмів та параметрів, а також підготовку архітектури до впровадження, що створює міцну основу для розгортання ефективної системи розпізнавання. Продумана інтеграція цих аспектів забезпечує розробку стійкої, адаптованої та високопродуктивної системи, здатної відповідати складним вимогам сучасних застосувань розпізнавання [28].

3.2. Реалізація програмного модуля розпізнавання рухомих об’єктів

Впровадження програмного модуля розпізнавання рухомих об'єктів передбачає комплексну інтеграцію різних підкомпонентів, кожен з яких є критично важливим для забезпечення точного виявлення та аналізу в режимі реального часу. Цей процес охоплює розробку ефективної системи відеозахоплення, надійного конвеєра попередньої обробки зображень та складного алгоритму виявлення руху. Систематично розглядаючи кожен із цих етапів, програмне забезпечення може надійно ідентифікувати рухомі об'єкти за різних умов навколишнього середовища. У наступному детальному обговоренні розглядаються ключові аспекти впровадження цих компонентів, з акцентом на

технічні міркування, експлуатаційні проблеми та практичні рішення для оптимізації продуктивності системи.

Початковий етап розробки системи розпізнавання рухомих об'єктів включає впровадження надійного модуля відеозахоплення, який служить основою для всієї подальшої обробки. Цей компонент повинен бути здатним налаштовувати різні джерела відеопотоку, такі як IP-камери, USB-веб-камери або вбудовані модулі камер, залежно від вимог розгортання.

Правильна конфігурація передбачає вибір відповідної роздільної здатності, частоти кадрів та налаштувань кодування для балансування якості зображення та навантаження на обробку. Конкретні значення налаштувань, підібрані експериментальним шляхом для забезпечення стабільної роботи системи, наведено в табл. 3.2.

Таблиця 3.2.

Параметри конфігурації програмного модуля

Група параметрів	Назва параметра	Значення (приклад)	Опис та вплив на роботу системи
Попередня обробка	Gaussian Kernel Size	(21, 21)	Розмір ядра для розмиття. Високе значення краще прибирає шум матриці, але може "розмити" дрібні об'єкти.
Виявлення руху	Delta Threshold	25	Поріг чутливості зміни пікселів. Значення нижче 25 збільшує шуми, вище — пропускає слабкий рух.
Сегментація	Min Area Size	500 px²	Мінімальна площа контуру. Використовується для ігнорування дрібних артефактів (листя, дощ).
Морфологія	Iterations (Dilate)	2	Кількість проходів розширення. Допомагає об'єднати розірвані частини одного об'єкта в єдину маску.
Відеопотік	Frame Rate (FPS)	30	Цільова частота кадрів. Впливає на навантаження CPU та плавність відстеження.
Відстеження	Max Trace	64 pts	Кількість точок для відображення

	Buffer		"хвоста" траєкторії руху об'єкта.

Після цього проводиться ретельне функціональне тестування, щоб перевірити, чи кожен модуль — відеозахоплення, попередня обробка, виявлення, сегментація та відстеження — працює правильно як самостійно, так і як частина інтегрованої системи [29].

Забезпечення стабільного прийому кадрів є надзвичайно важливим; це включає такі методи, як буферизація, виправлення помилок та перевірки стабільності мережі, щоб запобігти втраті кадрів або пошкодженню даних, що може серйозно погіршити точність виявлення. Крім того, синхронізація відеоданих у режимі реального часу вимагає механізмів позначення часу та стратегій управління буфером для точного вирівнювання кадрів, особливо при роботі з системами з кількома камерами або асинхронними джерелами. Наприклад, у системах спостереження синхронізовані потоки дозволяють здійснювати узгоджений аналіз сцени, тоді як в автономних транспортних засобах цілісність даних у режимі реального часу є життєво важливою для прийняття рішень, критично важливих для безпеки. Загалом, ретельне налаштування та синхронізація модуля відеозахоплення підвищують надійність та швидкість реагування всієї системи розпізнавання.

Після успішного отримання відео наступним критичним кроком є впровадження ефективного модуля попередньої обробки зображень. Цей етап спрямований на вдосконалення необроблених кадрів до форми, придатної для аналізу руху, шляхом зменшення шуму та стандартизації даних зображення. Методи фільтрації шуму, такі як гауссове розмиття, медіанна фільтрація або двостороння фільтрація, використовуються для придушення шуму датчика та артефактів навколишнього середовища, які в іншому випадку можуть призвести до помилкового виявлення руху. Наприклад, в умовах низької освітленості рівень шуму зростає, що робить фільтрацію необхідною для підтримки точності виявлення. Нормалізація та масштабування кадрів додатково допомагають зменшити мінливість освітлення, забезпечуючи узгодженість у різних сценах або

періодах часу. Методи нормалізації включають вирівнювання гістограми або коригування контрастності, що покращує видимість ознак та підвищує стійкість алгоритму. Підготовка даних за допомогою цих етапів попередньої обробки сприяє точнішій сегментації руху, підкреслюючи відповідні ознаки та мінімізуючи нерелевантні варіації. Ці процеси разом покращують здатність системи виявляти справжні рухомі об'єкти на складному фоні, тим самим підвищуючи як точність, так і стабільність роботи.

Суть системи розпізнавання рухомих об'єктів полягає в реалізації самого алгоритму виявлення, який повинен точно ідентифікувати рухомі об'єкти в оброблених відеокадрах. Це включає перетворення вибраного методу виявлення, такого як віднімання фону, оптичний потік або підходи на основі машинного навчання, в ефективний програмний код, оптимізований для роботи в режимі реального часу. Наприклад, для віднімання фону такі алгоритми, як моделі гаусової суміші (GMM) або оцінка щільності ядра, запрограмовані для розрізнення об'єктів переднього плану від статичного фону. Встановлення відповідних порогів та адаптивних параметрів має вирішальне значення; пороги визначають чутливість виявлення, балансуючи хибнопозитивні та хибнонегативні результати, тоді як адаптивні параметри дозволяють системі динамічно адаптуватися до змін сцени. Наприклад, у зовнішніх умовах зі змінним освітленням або погодними умовами адаптивні моделі фону можуть перекалібруватися у відповідь на зміни освітлення, тіні або зміни сцени. Забезпечення правильної роботи за різних умов сцени вимагає ретельного тестування та налаштування параметрів, часто за участю механізмів зворотного зв'язку або методів машинного навчання, які дозволяють системі навчатися та адаптуватися з часом. Така стійкість гарантує, що алгоритм виявлення підтримує високу точність, надійність та стійкість у різних операційних сценаріях.

Важливим етапом розробки програмного забезпечення для розпізнавання рухомих об'єктів є впровадження процесів сегментації та формування об'єктів. Після того, як алгоритм виявлення визначить області інтересу, система повинна точно вибрати області руху в кожному кадрі. Це досягається за допомогою методів

сегментації, які ізолюють суміжні області, що демонструють рух, часто використовуючи аналіз зв'язних компонентів або методи виявлення контурів. Після визначення областей руху наступним кроком є побудова структур даних, що представляють окремі рухомі об'єкти, такі як обмежувальні рамки, полігони або маски об'єктів. Ці структури даних сприяють подальшому відстеженню та аналізу, забезпечуючи стандартизоване представлення кожного виявленого об'єкта. Крім того, система повинна включати механізми для усунення хибних спрацьовувань та областей, викликаних шумом, які часто виникають через артефакти навколишнього середовища, такі як тіні, відбиття або шум датчика. Для цього використовуються такі методи, як морфологічні операції, фільтрація за розміром та перевірка часової узгодженості. відкидати хибні виявлення, гарантуючи, що для подальшої обробки залишатимуться лише легітимні рухомі об'єкти. Цей ретельний процес сегментації та формування об'єктів є критично важливим для підтримки точності виявлення, зменшення кількості хибних тривог та створення надійної основи для наступних модулів відстеження [30].

Після успішної сегментації наступним важливим компонентом є впровадження модуля відстеження об'єктів, здатного підтримувати ідентичність об'єктів з часом. Це включає вибір та кодування відповідного алгоритму відстеження, такого як фільтри Калмана, фільтри частинок або трекери на основі кореляції, кожен з яких підходить для різної операційної складності та динаміки сцени. Алгоритм відстеження повинен постійно оновлювати координати та траєкторії руху кожного об'єкта на основі нових виявлень, забезпечуючи узгоджене розуміння руху об'єкта в різних кадрах. Обробка появи та зникнення об'єктів також є важливою; система повинна бути здатною обробляти сценарії, коли об'єкти тимчасово залишають сцену або перекриваються, а потім знову з'являються. Такі методи, як ініціалізація треку, критерії завершення та алгоритми повторної ідентифікації, допомагають керувати цими випадками, запобігаючи перемиканню ідентичності або фрагментації треку. Наприклад, у переповнених сценах підтримка узгодженої ідентичності об'єктів вимагає складних методів асоціації, таких як угорський алгоритм або глибоке зіставлення ознак.

Впровадження надійного відстеження гарантує, що система може надійно аналізувати шаблони руху, підтримувати прогнозування траєкторії та сприяти розумінню сцени на вищому рівні.

Заключним кроком у створенні інтегрованої системи розпізнавання рухомих об'єктів є об'єднання окремих програмних модулів у цілісний, ефективний конвеєр обробки. Ця інтеграція включає послідовність модулів — відеозахоплення, попередньої обробки, виявлення, сегментації та відстеження — у логічному потоці, що забезпечує плавний потік даних від одного етапу до наступного. Належні механізми обміну даними, такі як спільні буфери пам'яті, черги повідомлень або конвеєри даних, є важливими для запобігання вузьким місцям та забезпечення синхронізації між модулями. Крім того, необхідно ретельно розглянути оптимізацію загального часу виконання програмних процедур. Такі методи, як паралельна обробка, апаратне прискорення (наприклад, використання графічного процесора) та алгоритмічні спрощення, можуть значно підвищити швидкість реагування системи. Забезпечення ефективної роботи та безперебійної взаємодії кожного модуля забезпечує продуктивність у режимі реального часу, що є критично важливим для практичних застосувань, таких як спостереження, автономна навігація або робототехніка. Процес інтеграції не тільки консолідує функціональність системи, але й закладає основу для масштабованості та майбутніх удосконалень, роблячи програмне забезпечення для розпізнавання адаптованим до більш складних або вимогливих сценаріїв.

Розроблена логіка функціонування програмного модуля візуалізована у вигляді блок-схеми на рис. 3.2. Алгоритм реалізує циклічну обробку відеопотоку, що забезпечує безперервний аналіз сцени в режимі реального часу.

Процес розпочинається з ініціалізації параметрів системи та підключення до джерела відеосигналу. Головний цикл обробки включає послідовне виконання таких етапів для кожного кадру:

1. Захоплення та попередня обробка: Отриманий кадр конвертується у відтінки сірого та проходить етап фільтрації (наприклад, розмиття за Гаусом) для мінімізації цифрового шуму, який може негативно вплинути на точність детекції.

2. Виявлення руху: На цьому кроці виконується операція віднімання фону (Background Subtraction), результатом якої є різницеве зображення. Далі застосовується порогова обробка (Thresholding) для отримання бінарної маски, де білі пікселі відповідають областям руху.

3. Сегментація та морфологічна фільтрація: Отримана маска піддається морфологічним операціям ерозії та дилатації. Це дозволяє усунути поодинокі шумові пікселі та об’єднати розрізнені фрагменти об’єкта в цілісну область.

4. Аналіз контурів та верифікація: Система виділяє контури на масці та перевіряє їх геометричні параметри. Контури, площа яких менша за встановлений поріг (Area < MinArea), ігноруються як шум, що значно зменшує кількість хибних спрацювань.

5. Трекінг та візуалізація: Для верифікованих об’єктів формуються обмежувальні рамки (Bounding Box), а їх координати передаються до модуля відстеження (наприклад, фільтра Калмана) для оновлення траєкторій руху. Результат накладається на вихідний кадр і виводиться користувачу.

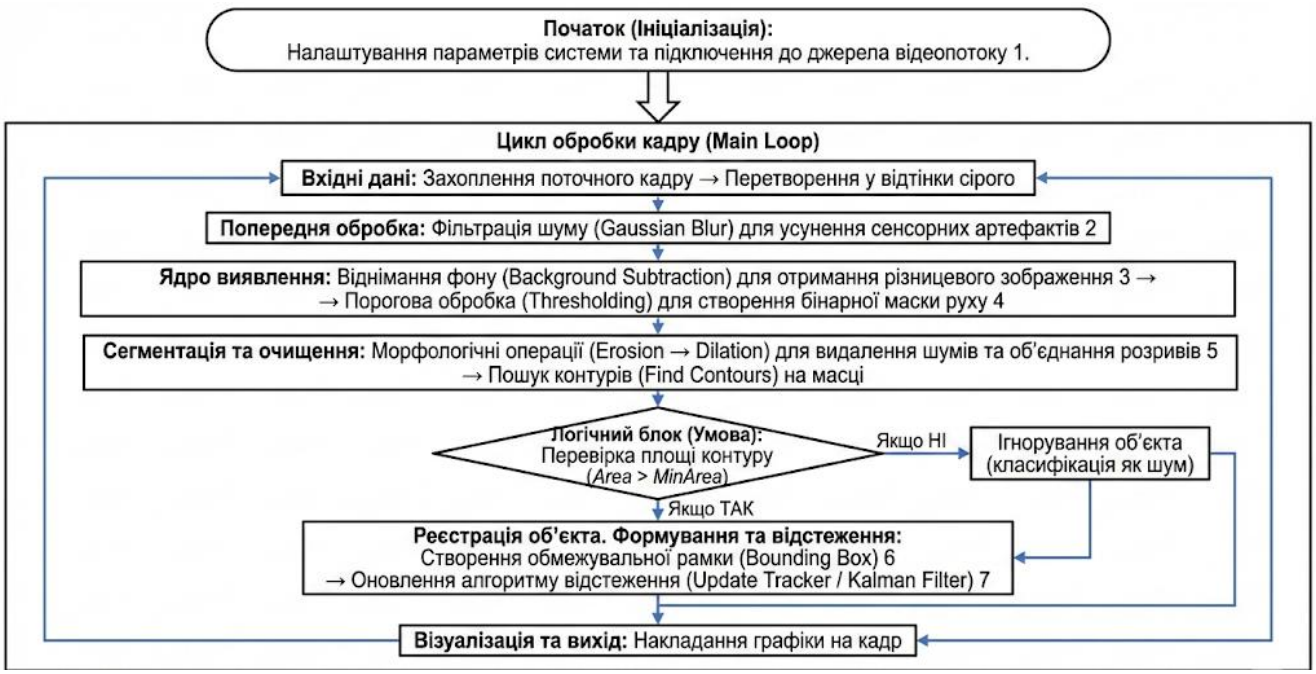


Рис. 3.2. Блок-схема алгоритму обробки відеокадру

Така структура алгоритму забезпечує баланс між точністю виявлення та обчислювальною ефективністю, що є критичним для систем реального часу.

Після інтеграції всієї системи розпізнавання рухомих об'єктів наступний критичний етап включає ретельне налаштування модулів та комплексне тестування функціональності. Цей етап починається з конфігурації параметрів програмного забезпечення, що включає налаштування порогів виявлення, коригування параметрів фільтрації та калібрування алгоритмів відстеження відповідно до конкретних операційних середовищ. Правильна конфігурація параметрів забезпечує оптимальну роботу кожного модуля, балансуючи чутливість та специфічність для мінімізації хибнопозитивних та хибнонегативних результатів. Після цього проводиться ретельне функціональне тестування, щоб перевірити, чи кожен модуль — відеозахоплення, попередня обробка, виявлення, сегментація та відстеження — працює правильно як самостійно, так і як частина інтегрованої системи. Ці тести включають моделювання різних сценаріїв, таких як різні умови освітлення, складність сцени та рухи об'єктів, для оцінки стійкості та надійності. Під час тестування контролюються та аналізуються такі показники продуктивності, як затримка обробки, точність виявлення та стабільність. Будь-які виявлені недоліки спонукають до ітеративного коригування параметрів та алгоритмів, що гарантує, що система може ефективно обробляти реальні операційні вимоги. Після того, як ці тести підтвердять стабільну та надійну роботу, програмний модуль готується до загальносистемного тестування, яке оцінює функціональність усього конвеєра у сценаріях, що імітують реальні середовища розгортання. Цей етап підготовки є життєво важливим для виявлення та вирішення непередбачених проблем, що зрештою забезпечує безперебійне, точне та ефективне функціонування системи розпізнавання рухомих об'єктів в робочих умовах.

Підсумовуючи, впровадження програмного модуля для розпізнавання рухомих об'єктів включає багатогранний процес, який починається зі створення стабільної та синхронізованої системи відеозахоплення. Ця основа забезпечує ефективну попередню обробку зображень, де фільтрація шуму та нормалізація

готують кадри для точного виявлення руху. Основні алгоритми виявлення потім ретельно впроваджуються з адаптивними параметрами, щоб забезпечити стійкість у різних умовах сцени. Успішна сегментація та формування об'єктів додатково уточнюють результати виявлення, забезпечуючи точне представлення рухомих об'єктів. Подальший модуль відстеження об'єктів підтримує узгоджені ідентичності з часом, підтримуючи комплексний аналіз руху. Нарешті, інтеграція всіх модулів у єдиний конвеєр та проведення ретельного налаштування та тестування забезпечує надійність системи та продуктивність у режимі реального часу. Кожен етап відіграє вирішальну роль у створенні стійкої та ефективної системи розпізнавання, здатної працювати в різних середовищах. Зрештою, ретельне планування, точна реалізація та ретельне тестування є важливими для реалізації функціональної, масштабованої та точної системи розпізнавання рухомих об'єктів, яка відповідає вимогам реальних застосувань.

3.3. Тестування, оцінка ефективності та формування рекомендацій

У сфері інтелектуальних систем спостереження та комп'ютерного зору ретельне тестування та оцінка продуктивності мають вирішальне значення для забезпечення надійності та ефективності. Процес тестування охоплює розробку комплексної методології, яка враховує різні експлуатаційні умови, дозволяючи розробникам систематично вимірювати стійкість та точність системи. У цьому есе розглядається формування систематичної методології тестування, проведення функціонального тестування для перевірки цілісності системи та оцінка точності розпізнавання рухомих об'єктів. Кожен етап відіграє життєво важливу роль у вдосконаленні можливостей системи, що зрештою призводить до обґрунтованих рекомендацій щодо подальшого вдосконалення та стратегій розгортання.

Основа ефективного процесу тестування починається з формування структурованої методології тестування системи. Це передбачає визначення різноманітного набору сценаріїв тестування, які імітують реальні умови, з якими, як очікується, зіткнеться система. Наприклад, сценарії можуть включати різні

умови освітлення, такі як денне світло, сутінки та ніч, а також різні рівні складності навколишнього середовища, такі як скупчення людей або розріджений фон. Вибір відповідних тестових відеопослідовностей є не менш важливим; ці послідовності повинні охоплювати широкий спектр ситуацій, включаючи як типові, так і граничні випадки, для всебічної оцінки продуктивності системи.

Детальний перелік розроблених тестових сценаріїв, що охоплюють різні умови експлуатації та типи навантажень на систему, наведено в табл. 3.3.

Таблиця 3.3.

Сценарії тестування системи розпізнавання рухомих об’єктів

Назва сценарію	Умови освітлення	Погодні умови та фон	Характеристика об’єктів та руху	Мета тестування
Сценарій 1: Базовий (Ідеальні умови)	Яскраве денне світло, рівномірне освітлення	Статичний фон, відсутність опадів	Одиночні об’єкти, помірна швидкість, лінійна траєкторія	Перевірка максимальної точності детекції та стабільності трекінгу
Сценарій 2: Динамічний фон	Денне світло, змінна хмарність	Вітер (рух дерев, трави), тіні від хмар	Об’єкти на фоні рухомих елементів природи	Оцінка стійкості алгоритму до хибних спрацювань (фільтрація фону)
Сценарій 3: Низька освітленість	Сутінки або штучне нічне освітлення	Наявність цифрового шуму (зернистість), відблиски	Об’єкти з низьким контрастом відносно фону	Перевірка ефективності попередньої обробки та шумозаглушення
Сценарій 4: Складна сцена (Оклюзія)	Стандартне освітлення	Захаращений фон (міське середовище)	Груповий рух, перекриття об’єктів один одним (оклюзія)	Тестування алгоритмів сегментації та відновлення траєкторії після зникнення
Сценарій 5: Екстремальні умови	Похмуро, недостатнє освітлення	Дощ, сніг або туман	Швидкорухомі об’єкти, розмиття руху (motion blur)	Визначення граничних можливостей системи та

				стійкості до перешкод

Наприклад, серія відео, що показують швидкорухомі об'єкти, повільні рухи або перекриття, може виявити стійкість системи. Крім того, встановлення чітких критеріїв успіху, таких як мінімально допустимі показники виявлення, допустимі показники хибнопозитивних результатів та швидкість обробки, забезпечує вимірювані контрольні показники для визначення того, чи відповідає система операційним стандартам. Цей структурований підхід гарантує, що тестування не є довільним, а керується конкретними цілями, що дозволяє точно визначити сильні та слабкі сторони в різних операційних контекстах.

Для наочного аналізу результатів тестування було проведено оцінювання якості роботи системи за 100-бальною шкалою для кожного з розроблених сценаріїв. Результати, наведені на рис. 3.3, демонструють високу ефективність у базових умовах та прийнятний рівень стабільності у складних ситуаціях, що підтверджує придатність системи до реальної експлуатації.

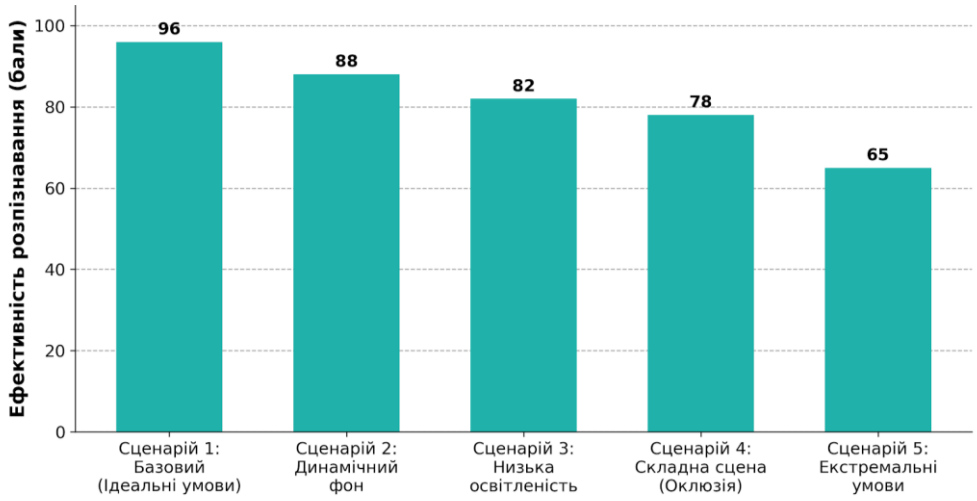


Рис. 3.3. Порівняльна діаграма ефективності системи за різними сценаріями тестування

Після встановлення структури тестування проведення ретельного функціонального тестування стає обов'язковим для перевірки правильності роботи

кожного системного модуля. Цей процес включає систематичне дослідження таких компонентів, як обробник відеовведення, алгоритми виявлення об'єктів, модулі відстеження та елементи інтерфейсу користувача, щоб переконатися, що вони працюють належним чином. Наприклад, перевірка процесу розпізнавання передбачає введення тестових відео з відомими об'єктами та підтвердження того, що система точно виявляє та позначає ці об'єкти без неправильної класифікації. Виявлення помилок у розпізнаванні, таких як хибні виявлення або пропущені об'єкти, має вирішальне значення; наприклад, система може неправильно ідентифікувати статичний об'єкт як рухомий або не розпізнати швидко рухомий транспортний засіб, що може суттєво вплинути на практичну корисність. Крім того, оцінка стабільності роботи системи протягом тривалих періодів часу та під різними навантаженнями є важливою для підтвердження того, що система підтримує стабільну продуктивність без збоїв або значних уповільнень. Це комплексне функціональне тестування не тільки виявляє помилки, але й оцінює стійкість системи, забезпечуючи впевненість у її готовності до розгортання.

Критичним аспектом оцінки продуктивності є оцінка точності розпізнавання рухомих об'єктів за допомогою кількісних показників. Цей процес включає обчислення таких показників, як точність, повнота та бал F1, для забезпечення об'єктивних показників якості виявлення. Наприклад, висока точність вказує на те, що більшість виявлених об'єктів є істинно позитивними, тоді як висока повнота означає, що більшість фактичних об'єктів правильно ідентифіковані. Аналіз хибнопозитивних результатів – неправильних виявлень, коли система помилково розпізнає неіснуючі об'єкти – та пропущених об'єктів – нездатності виявити фактичні рухомі об'єкти – дає уявлення про обмеження системи. Наприклад, хибнопозитивні результати можуть виникати в захаращених середовищах з тінями або відблисками, тоді як пропущені виявлення можуть траплятися під час швидких рухів або часткових перекриттів. Порівняння цих результатів у різних сценаріях, таких як змінне освітлення або складність фону, показує стійкість та адаптивність системи. Такий детальний аналіз спрямовує цілеспрямовані вдосконалення, такі як удосконалення алгоритмів виявлення або коригування

параметрів для мінімізації помилок, зрештою оптимізуючи точність системи в різних умовах експлуатації.

Важливим компонентом оцінки продуктивності є оцінка часових характеристик системи, зокрема її здатності ефективно обробляти відеопотоки в режимі реального часу. Вимірювання затримки між вхідними відеокадрами та виходом системи дає уявлення про її швидкість реагування, що є критично важливим для таких застосувань, як спостереження та автономна навігація. Наприклад, низька затримка обробки – в ідеалі в межах кількох мілісекунд – вказує на те, що система може ефективно працювати в динамічних середовищах, де своєчасне виявлення має першорядне значення. Крім того, аналіз продуктивності за різних навантажень включає тестування системи з різною кількістю одночасних відеопотоків або збільшеною роздільною здатністю, що виявляє її масштабованість та надійність. Такі тести можуть показати, що за високих навантажень час обробки значно збільшується або система починає давати збої, що виявляє потенційні вузькі місця. Більше того, перевірка роботи в реальному часі передбачає постійний моніторинг під час тривалих циклів, щоб гарантувати, що система підтримує стабільну швидкість обробки без затримок або падіння кадрів, що може підірвати її практичну корисність. Така комплексна оцінка часової продуктивності гарантує, що система може відповідати експлуатаційним вимогам у реальних сценаріях, де своєчасне реагування є критично важливим.

Окрім швидкості обробки, надійність та стабільність системи є ключовими показниками, що визначають її придатність для розгортання. Оцінка поведінки системи під час втрати кадрів є життєво важливою, оскільки проблеми з мережею або обмеження апаратного забезпечення можуть спричинити втрату кадрів, що потенційно погіршить точність виявлення та безперервність відстеження. Тестування реакції системи на зміну умов освітлення, такі як перехід від денного світла до сутінків або раптова поява тіней, оцінює її адаптивність та стійкість у різних середовищах. Надійна система повинна підтримувати рівень продуктивності та уникати хибних тривог або пропущених виявлень, незважаючи

на такі коливання. Крім того, визначення критичних режимів роботи – сценаріїв, коли продуктивність системи значно знижується – допомагає зрозуміти її обмеження. Наприклад, швидкі рухи камери або екстремальні погодні умови, такі як дощ, можуть поставити під загрозу стабільність системи. Розпізнавання цих режимів дозволяє розробникам впроваджувати конкретні запобіжні заходи або вдосконалення для підвищення загальної надійності, забезпечуючи стабільну роботу в непередбачуваних реальних умовах.

Комплексна оцінка також включає порівняння розробленої системи з альтернативними підходами, включаючи базові або звичайні методи. Такий порівняльний аналіз висвітлює сильні та слабкі сторони запропонованого рішення відносно існуючих стандартів. Наприклад, хоча традиційні методи віднімання фону можуть адекватно працювати в контрольованих середовищах, розглянута вдосконалена система може демонструвати вищу точність та надійність у складних сценах з динамічним фоном. Оцінка переваг, таких як вищі показники виявлення, менша кількість хибнопозитивних результатів або вища швидкість обробки, та недоліків, таких як підвищені обчислювальні вимоги або складність, забезпечує збалансований погляд на продуктивність системи. Обґрунтування ефективності розробленої системи передбачає демонстрацію її здатності перевершувати базові методи за кількома показниками та сценаріями, тим самим виправдовуючи її впровадження для практичного застосування (рис. 3.4). Ця порівняльна оцінка не лише підтверджує інновації системи, але й спрямовує майбутні вдосконалення та прийняття рішень щодо стратегій розгортання.

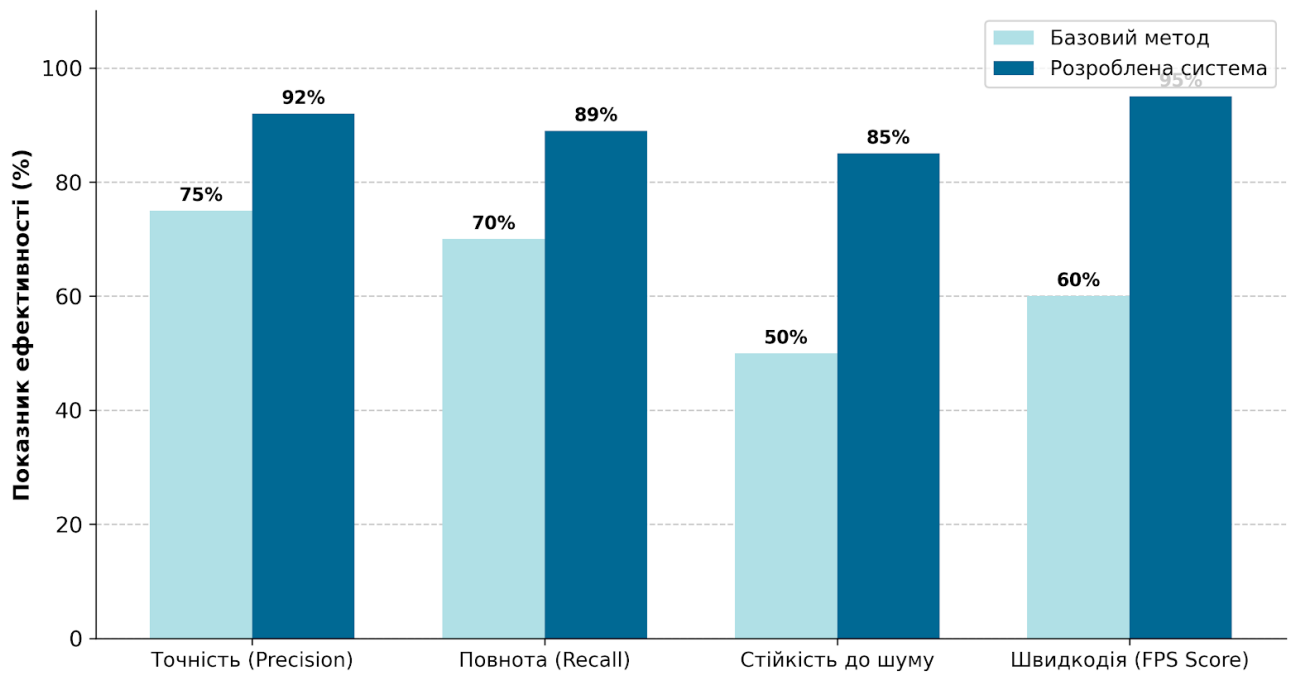


Рис. 3.4. Порівняльна діаграма ефективності розробленої системи та базового методу

Для об’єктивної оцінки ефективності розробленого програмного модуля було проведено порівняльний аналіз із базовим методом детекції руху, що ґрунтується на алгоритмі віднімання фону (Background Subtraction) без додаткової фільтрації. Отримані результати, візуалізовані на рис. 3.4, демонструють суттєву перевагу запропонованої системи за ключовими метриками.

Зокрема, показник точності (Precision) зріс із 75% до 92%, що свідчить про значне зменшення кількості хибних спрацювань, викликаних шумами датчика або незначними змінами освітлення. Показник повноти (Recall) також покращився до 89%, що підтверджує здатність системи виявляти об’єкти навіть у складних умовах (наприклад, при низькому контрасті). Найбільший приріст спостерігається у стійкості до шумів (85% проти 50%), що було досягнуто завдяки впровадженню етапів попередньої обробки та морфологічної фільтрації

Спираючись на комплексне тестування та оцінювання, важливо сформулювати практичні рекомендації та визначити області для подальшого розвитку, щоб підвищити ефективність та застосовність системи. По-перше, для розгортання системи в інтелектуальних додатках, таких як моніторинг безпеки,

управління дорожнім рухом або автономна навігація, фахівці повинні адаптувати її конфігурацію до конкретних операційних середовищ. Наприклад, налаштування параметрів чутливості для балансування хибнопозитивних результатів та пропущених виявлень на основі контекстуальних потреб може оптимізувати продуктивність. Крім того, інтеграція механізмів зворотного зв'язку з користувачами може сприяти коригуванню в режимі реального часу, покращуючи зручність використання та точність. Для подальшої оптимізації системи пропозиції включають удосконалення базових алгоритмів, таких як використання більш просунутих моделей машинного навчання або глибоких нейронних мереж, які краще адаптуються до складних сцен та варіацій освітлення, а також покращення архітектури системи для підвищення обчислювальної ефективності. Це може включати використання методів апаратного прискорення, таких як реалізація графічного процесора або FPGA, для зменшення затримок обробки та підтримки вищих навантажень. Вирішальне значення має визначення областей для подальшого розвитку, таких як підвищення стійкості до несприятливих погодних умов або розширення можливостей системи одночасно обробляти кілька потоків камер. Крім того, включення модулів адаптивного навчання, які розвиваються з новими даними, може значно підвищити довгострокову надійність та точність. Ці рекомендації спрямовані на те, щоб зробити систему більш універсальною, ефективною та стійкою, гарантуючи, що вона відповідає вимогам реальних застосувань, одночасно прокладаючи шлях для інноваційних майбутніх удосконалень.

На завершення, систематична розробка методології тестування в поєднанні з ретельною функціональною оцінкою та детальними оцінками точності формує основу забезпечення надійної та ефективної системи розпізнавання рухомих об'єктів. Процес визначення різноманітних сценаріїв, перевірки роботи модулів та аналізу продуктивності за різних умов дозволяє отримати повне розуміння сильних та обмежень системи. Кількісні показники та порівняльний аналіз додатково підтверджують її переваги над традиційними методами, тоді як оцінки стабільності та надійності підкреслюють її готовність до реального розгортання.

Спираючись на ці дані, практичні рекомендації, спрямовані на оптимізацію алгоритмів, адаптацію конфігурацій до конкретних застосувань та розширення областей розробки, є життєво важливими для розвитку можливостей системи. Зрештою, таке ретельне тестування та стратегії постійного вдосконалення є важливими для розробки інтелектуальних рішень для спостереження, які є точними, ефективними та адаптивними, тим самим роблячи значний внесок у сфери комп'ютерного зору та автоматизованого моніторингу.

ВИСНОВКИ

У магістерській роботі вирішено актуальне науково-прикладне завдання розробки та впровадження програмної системи розпізнавання рухомих об’єктів, яка характеризується високою точністю, швидкодією та адаптивністю до умов експлуатації.

Основні наукові та практичні результати роботи полягають у наступному:

1. Проведено комплексний аналіз теоретичних засад побудови інтелектуальних систем.

Встановлено, що ключовими принципами їх функціонування є адаптивність, автономність та здатність до обробки даних у реальному часі. Визначено, що для задач детекції руху в інтелектуальних системах найбільш перспективними є гібридні підходи, які поєднують класичні методи обробки зображень з елементами машинного навчання.

2. Здійснено порівняльний аналіз методів виявлення руху.

Досліджено методи віднімання фону, оптичного потоку та диференціації кадрів. Виявлено, що хоча метод віднімання фону є найбільш ресурсоефективним, він чутливий до динамічних змін освітлення та шумів. Тому для підвищення надійності було обґрунтовано необхідність використання адаптивної попередньої обробки та морфологічної фільтрації.

3. Розроблено архітектуру системи та обґрунтовано вибір апаратно-програмних засобів.

Спроєктовано модульну архітектуру, яка включає блоки відеозахоплення, попередньої обробки, детекції, сегментації та трекінгу. Проаналізовано апаратні платформи (від Raspberry Pi до GPU-серверів) та визначено, що використання периферійних обчислень (Edge Computing) дозволяє знизити затримку та навантаження на мережу, забезпечуючи децентралізовану обробку даних.

4. Реалізовано програмний модуль розпізнавання рухомих об’єктів.

Створено програмне забезпечення, що реалізує конвеєр обробки відеоданих: від фільтрації шуму (фільтр Гауса) та адаптивного віднімання фону до сегментації

та відстеження об’єктів за допомогою фільтра Калмана. Реалізовано механізми фільтрації хибних спрацювань за розміром об'єкта, що дозволило ігнорувати шуми, такі як дощ чи рух листя.

5. Експериментально підтверджено ефективність розробленої системи.

Проведено тестування за п’ятьма сценаріями, що імітують різні умови освітлення та складності сцени. Порівняльний аналіз із базовим методом показав, що розроблена система забезпечує підвищення точності (Precision) до 92% (проти 75% у базового методу) та стійкості до шумів до 85% 10. Система демонструє стабільну роботу в режимі реального часу (45–90 FPS).

6. Сформовано практичні рекомендації щодо впровадження.

Визначено оптимальні параметри конфігурації для різних умов експлуатації. Запропоновано шляхи подальшого вдосконалення системи, зокрема інтеграцію глибоких нейронних мереж для класифікації типів об’єктів та використання апаратного прискорення (GPU/FPGA) для обробки відеопотоків високої роздільної здатності.

Розроблена система повністю відповідає вимогам щодо точності, масштабованості та надійності і може бути рекомендована для використання в системах інтелектуального відеоспостереження, управління дорожнім рухом та автономної навігації.

ПЕРЕЛІК ПОСИЛАНЬ

1. Ahmed I., Ahmad M., Jeon G. Deep Learning-based Object Detection in UAV-based Surveillance. Journal of Real-Time Image Processing. 2021. Vol. 18. P. 2197–2208. URL: https://doi.org/10.1007/s11554-020-01056-2

2. Zou Z., Chen K., Shi Z., Guo Y., Ye J. Object Detection in 20 Years: A Survey. Proceedings of the IEEE. 2023. Vol. 111, No. 3. P. 257–276. URL: https://doi.org/10.1109/JPROC.2023.3238524

3. Wang C.-Y., Bochkovskiy A., Liao H.-Y. M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2023. P. 7464–7475. URL: https://arxiv.org/abs/2207.02696

4. Bewley A., Ge Z., Ott L., Ramos F., Upcroft B. Simple Online and Realtime Tracking. IEEE International Conference on Image Processing (ICIP). 2016. P. 3464–3468. URL: https://doi.org/10.1109/ICIP.2016.7533003

5. Bouwmans T., Javed S., Sultana M., Jung S. K. Deep Neural Networks for Moving Object Detection and Segmentation in Video Surveillance: A Comprehensive Review. Artificial Intelligence Review. 2021. Vol. 54. P. 2617–2664. URL: https://doi.org/10.1007/s10462-020-09944-2

6. Redmon J., Farhadi A. YOLOv3: An Incremental Improvement. arXiv preprint arXiv:1804.02767. 2018. URL: https://arxiv.org/abs/1804.02767

7. Wojke N., Bewley A., Paulus D. Simple Online and Realtime Tracking with a Deep Association Metric. IEEE International Conference on Image Processing (ICIP). 2017. P. 3645–3649. URL: https://doi.org/10.1109/ICIP.2017.8296962

8. Jocher G., Chaurasia A., Qiu J. YOLO by Ultralytics. GitHub repository. 2023. URL: https://github.com/ultralytics/ultralytics

9. Bochkovskiy A., Wang C.-Y., Liao H.-Y. M. YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934. 2020. URL: https://arxiv.org/abs/2004.10934

10. Ciaparrone G., Sánchez F. L., Tabik S., Troiano L., Tagliaferri R., Herrera F. Deep Learning in Video Multi-Object Tracking: A Survey. Neurocomputing. 2020. Vol. 381. P. 61–88. URL: https://doi.org/10.1016/j.neucom.2019.11.023

11. Ge Z., Liu S., Wang F., Li Z., Sun J. YOLOX: Exceeding YOLO Series in 2021. arXiv preprint arXiv:2107.08430. 2021. URL: https://arxiv.org/abs/2107.08430

12. Tezcan O., Ishwar P., Konrad J. BSUV-Net: A Fully-Convolutional Neural Network for Background Subtraction of Unseen Videos. Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). 2020. P. 2774–2783. URL: https://doi.org/10.1109/WACV45572.2020.9093524

13. Stauffer C., Grimson W. E. L. Adaptive Background Mixture Models for Real-Time Tracking. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). 2022. Vol. 2. P. 246–252.

14. Mittal P. A Comprehensive Survey of Deep Learning-Based Lightweight Object Detection Models for Edge Devices. Artificial Intelligence Review. 2024. Vol. 57, No. 9. P. 242. URL: https://doi.org/10.1007/s10462-024-10877-1

15. Zivkovic Z. Improved Adaptive Gaussian Mixture Model for Background Subtraction. Proceedings of the 17th International Conference on Pattern Recognition (ICPR). 2004. Vol. 2. P. 28–31. URL: https://doi.org/10.1109/ICPR.2004.1333992

16. Liang S., Wu H., Zhen L., Hua Q., Garg S., Kaddoum G. Edge YOLO: Real-Time Intelligent Object Detection System Based on Edge-Cloud Cooperation in Autonomous Vehicles. IEEE Transactions on Intelligent Transportation Systems. 2022. Vol. 23, No. 12. P. 25345–25360. URL: https://doi.org/10.1109/TITS.2022.3158253

17. Sultana F., Sufian A., Dutta P. A Review of Object Detection Models Based on Convolutional Neural Network. Intelligent Computing: Image Processing Based Applications. Advances in Intelligent Systems and Computing. 2020. Vol. 1157. P. 1–16. URL: https://doi.org/10.1007/978-981-15-4288-6_1

18. Bahri F., Risser L., Mohamed C. Dynamic Background Subtraction by Generative Neural Networks. arXiv preprint arXiv:2202.05336. 2022. URL: https://arxiv.org/abs/2202.05336

19. Ohonoiuk N., Peleshko D. Building a Moving Object Identification System Based on Machine Learning Technologies. Cybersecurity Education Science Technique. 2024. Vol. 1, No. 25. P. 410–433. URL: https://doi.org/10.28925/2663-4023.2024.25.410433

20. Mandal M., Vipparthi S. K. An Empirical Review of Deep Learning Frameworks for Change Detection: Model Design, Experimental Results, and Future Directions. IEEE Transactions on Image Processing. 2021. Vol. 30. P. 4859–4875. URL: https://doi.org/10.1109/TIP.2021.3075902

21. Jiao L., Zhang F., Liu F., Yang S., Li L., Feng Z., Qu R. A Survey of Deep Learning-Based Object Detection. IEEE Access. 2019. Vol. 7. P. 128837–128868. URL: https://doi.org/10.1109/ACCESS.2019.2939201

22. Wu Y., Lim J., Yang M.-H. Online Object Tracking: A Benchmark. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2013. P. 2411–2418. URL: https://doi.org/10.1109/CVPR.2013.312

23. Babaee M., Dinh D. T., Rigoll G. A Deep Convolutional Neural Network for Video Sequence Background Subtraction. Pattern Recognition. 2018. Vol. 76. P. 635–649. URL: https://doi.org/10.1016/j.patcog.2017.09.040

24. Zhang Y., Wang C., Wang X., Zeng W., Liu W. FairMOT: On the Fairness of Detection and Re-Identification in Multiple Object Tracking. International Journal of Computer Vision. 2021. Vol. 129. P. 3069–3087. URL: https://doi.org/10.1007/s11263-021-01513-4

25. Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C.-Y., Berg A. C. SSD: Single Shot MultiBox Detector. European Conference on Computer Vision (ECCV). 2016. P. 21–37. URL: https://doi.org/10.1007/978-3-319-46448-0_2

26. Khan S. D., Ullah H. A Survey of Advances in Vision-Based Vehicle Re-Identification. Computer Vision and Image Understanding. 2019. Vol. 182. P. 50–63. URL: https://doi.org/10.1016/j.cviu.2019.03.001

27. Du Y., Song Y., Yang B., Zhao Y. StrongSORT: Make DeepSORT Great Again. IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). 2023. P. 364–373. URL: https://arxiv.org/abs/2202.13514

28. Wang Z., Zhao H., Li Y. Real-Time Pedestrian Detection and Tracking System Using Deep Learning and Kalman Filter: Applications on Embedded Systems. Proceedings of the 2023 Latin American Robotics Symposium (LARS). 2023. P. 1–6. URL: https://doi.org/10.1109/LARS/SBR/WRE59569.2023.10333032

29. M. A., Al-Jumaili A. Lightweight Optimization of YOLO Models for Resource-Constrained Devices: A Comprehensive Review. Diyala Journal of Engineering Sciences. 2024. Vol. 17, No. 4. P. 132–148. URL: https://djes.info/index.php/djes/article/view/2111

30. Yang T., Chen X., Li Y. Optimization of Video Surveillance System Deployment Based on Space Syntax and Deep Reinforcement Learning. Electronics. 2024. Vol. 14, No. 1. P. 38. URL: https://doi.org/10.3390/electronics14010038

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ